ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 1

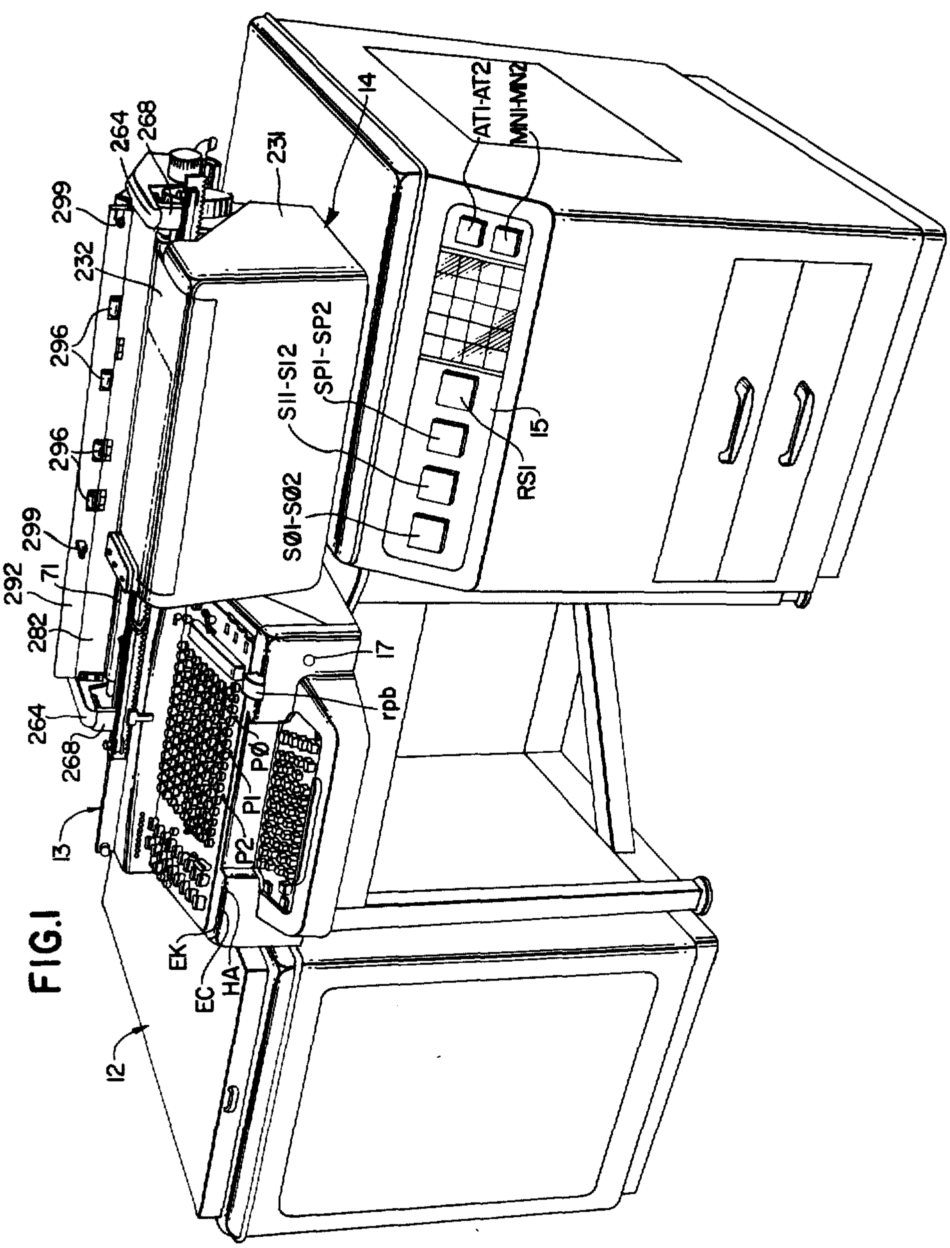


FIG.1

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E.V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

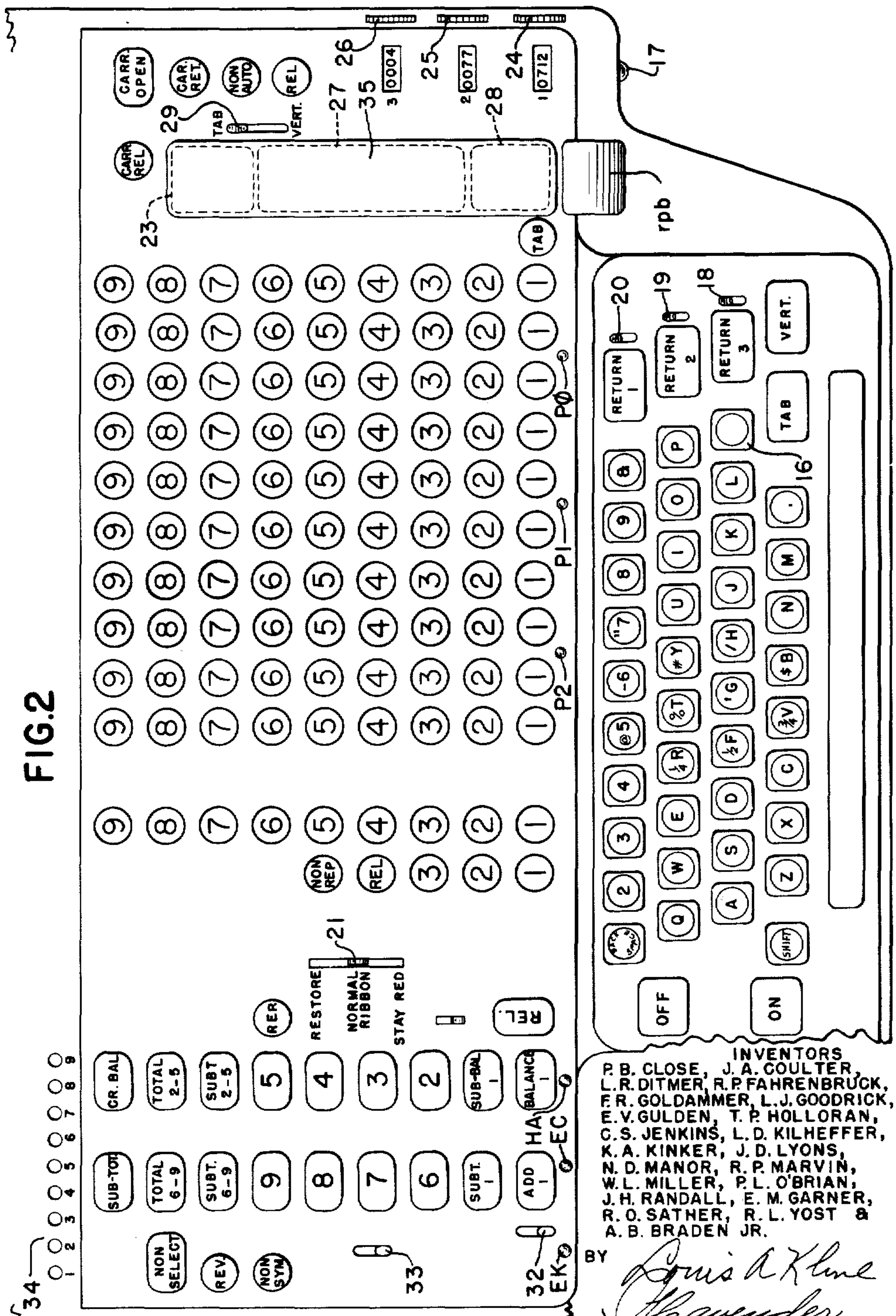


FIG. 2

INVENTORS
P. B. CLOSE, J. A. COULTER,
L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK,
E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER,
K. A. KINKER, J. D. LYONS,
N. D. MANOR, R. P. MARVIN,
W. L. MILLER, P. L. O'BRIAN,
J. H. RANDALL, E. M. GARNER,
R. O. SATHER, R. L. YOST &
A. B. BRADEN JR.

BY Louis A. Kline
H. Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 3

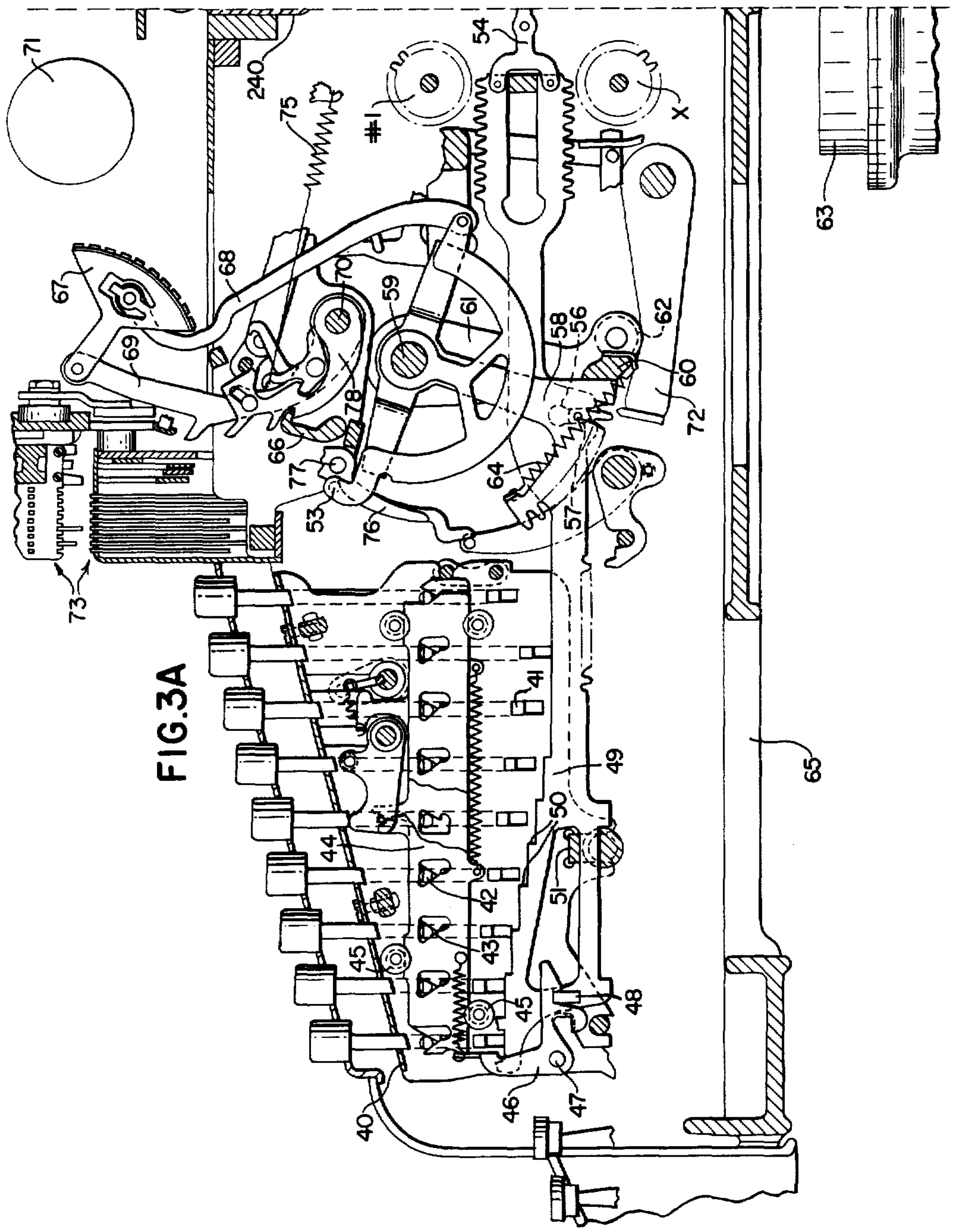


FIG.3A

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 4

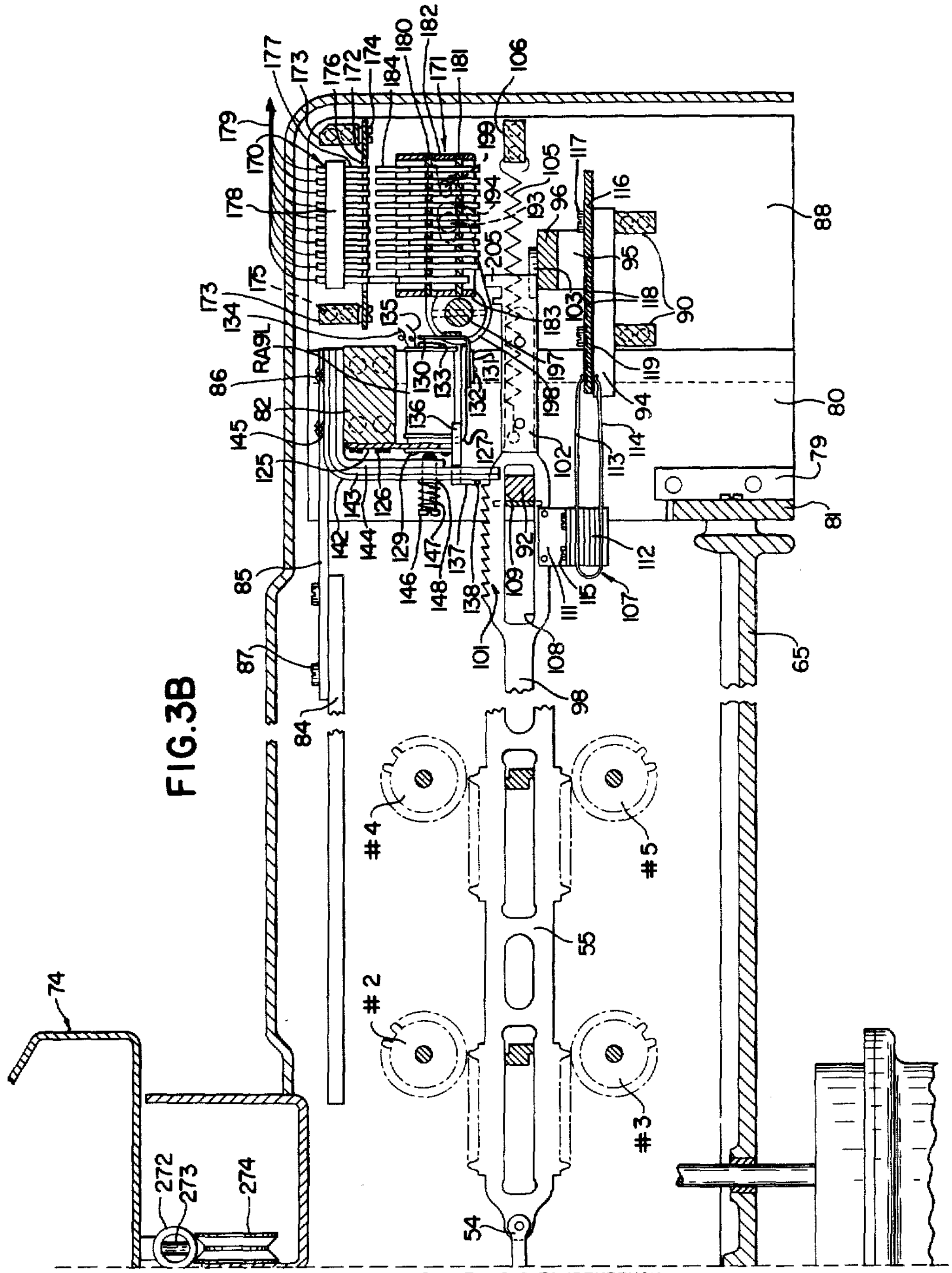


FIG.3B

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Cavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 5

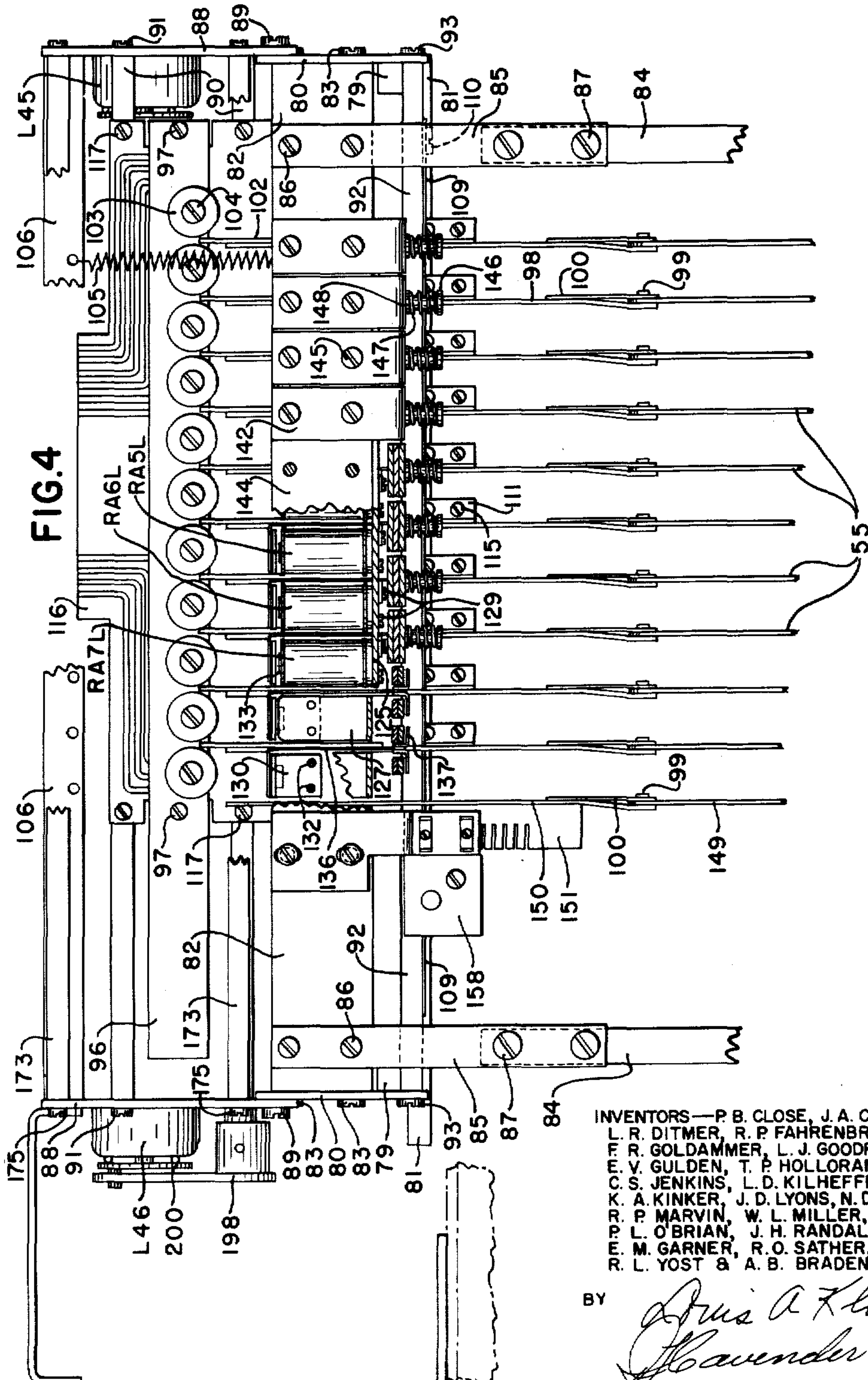


FIG. 4

INVENTORS—P. B. CLOSE, J. A. COULTER,
L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK,
E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER,
K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER,
P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER,
R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
[signature]

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 6

FIG. 5A

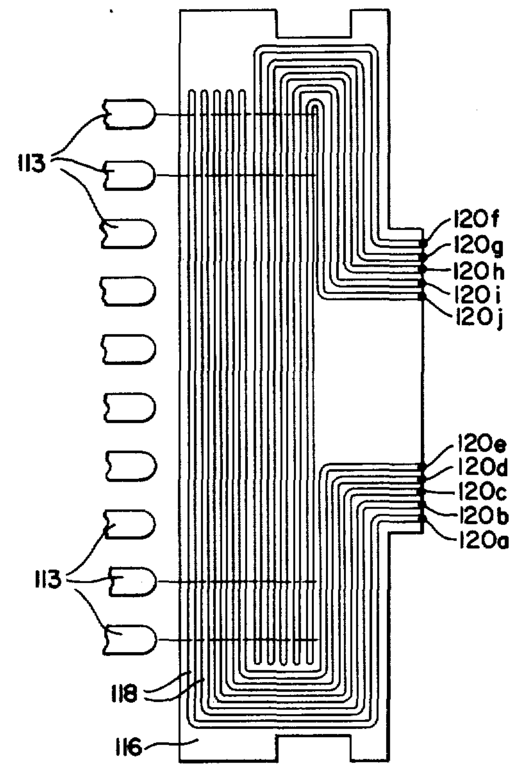


FIG. 5B

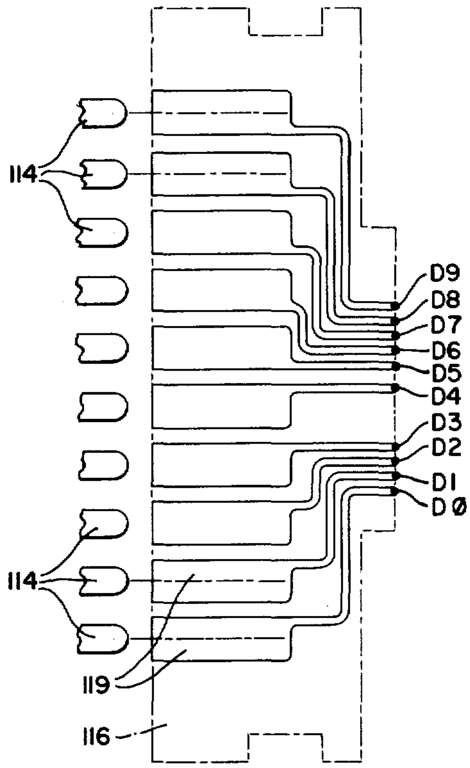
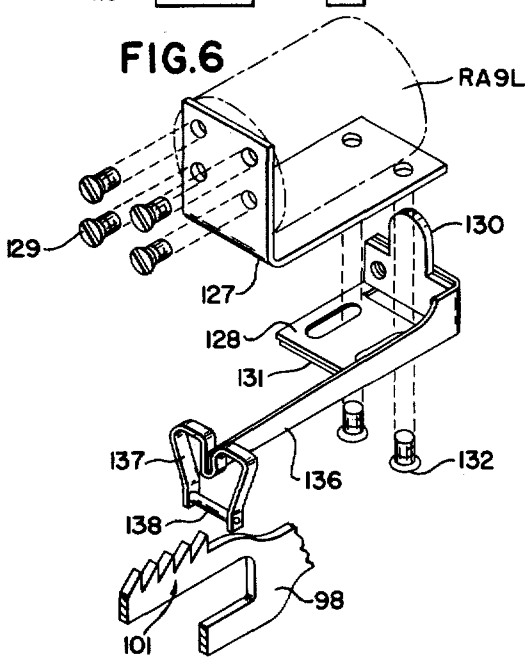


FIG. 6



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline

J Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 7

FIG.7

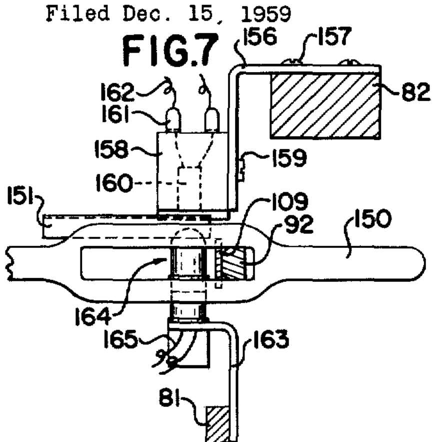


FIG.8

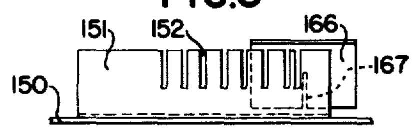


FIG.10

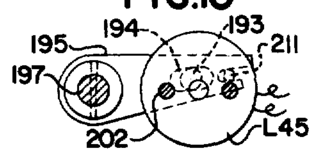


FIG.9

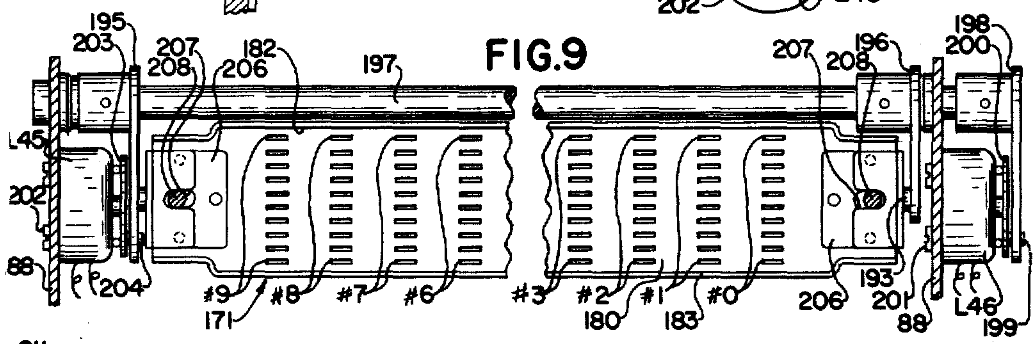


FIG.11

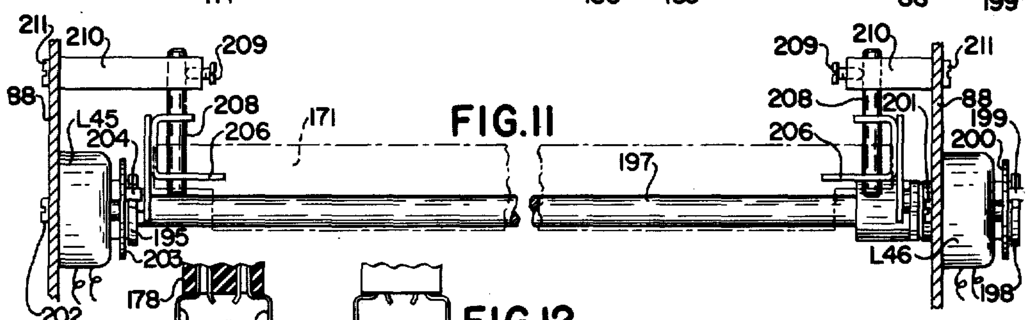
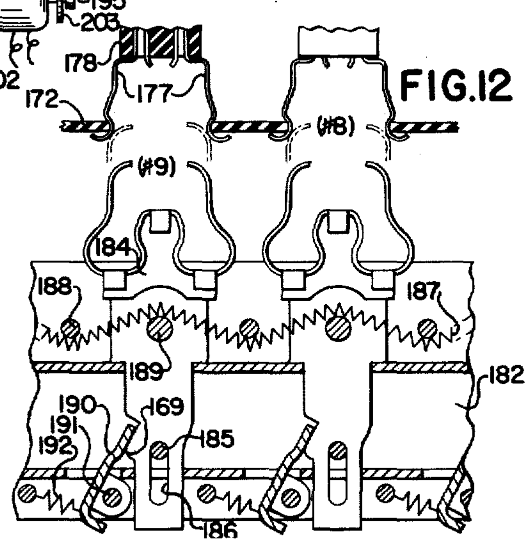


FIG.12

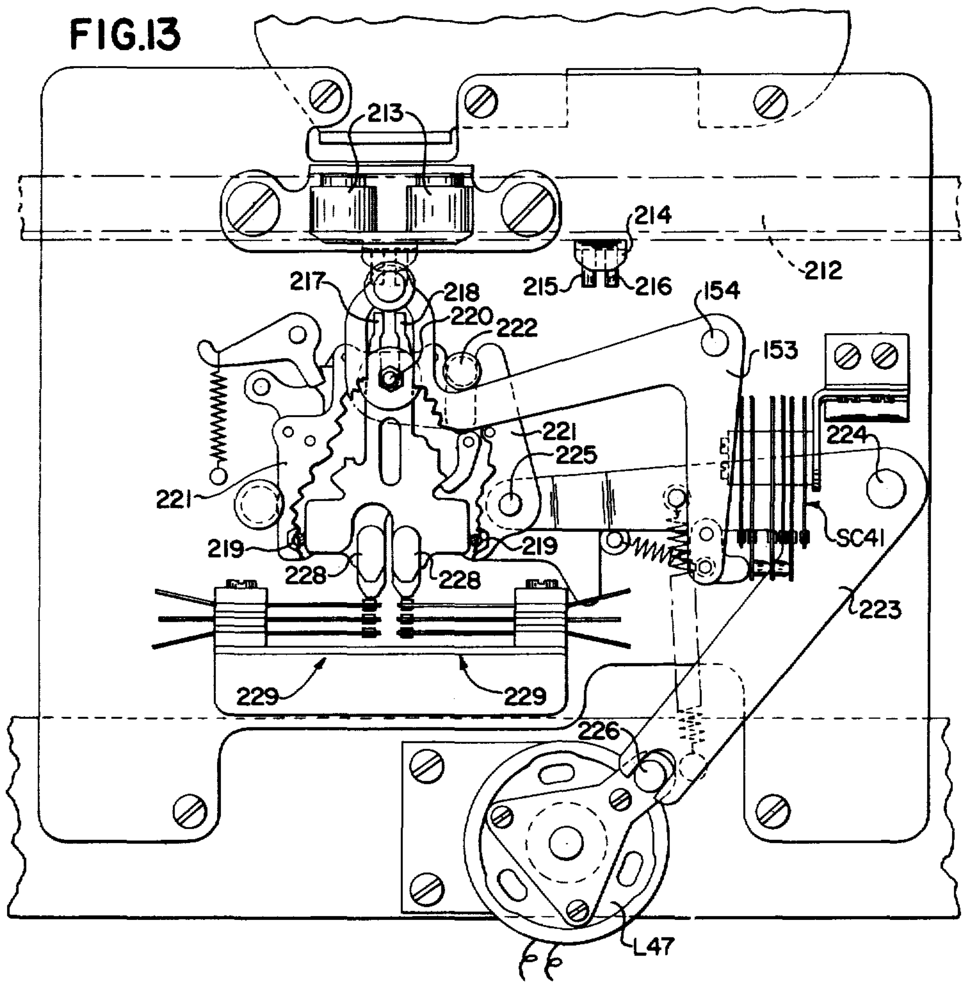


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Hawender

THEIR ATTORNEYS

FIG.13



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

FIG.14

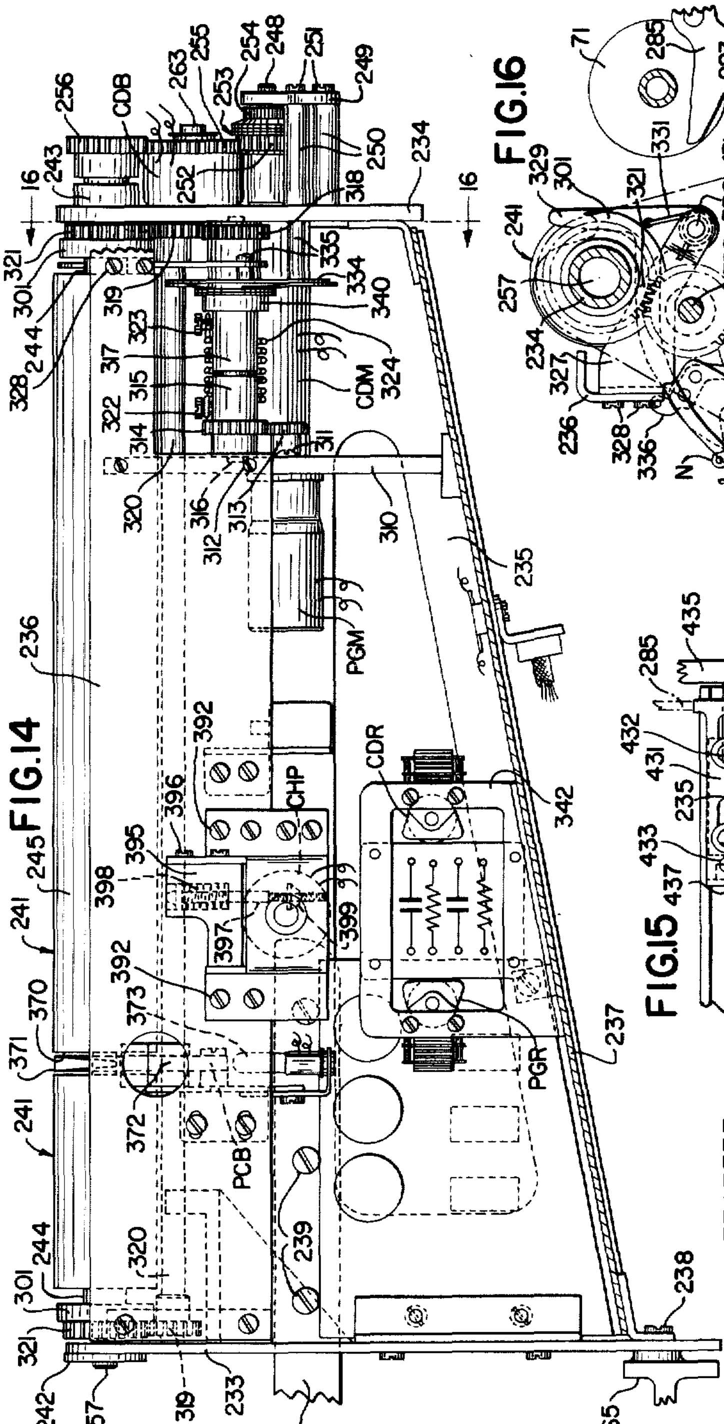


FIG.16

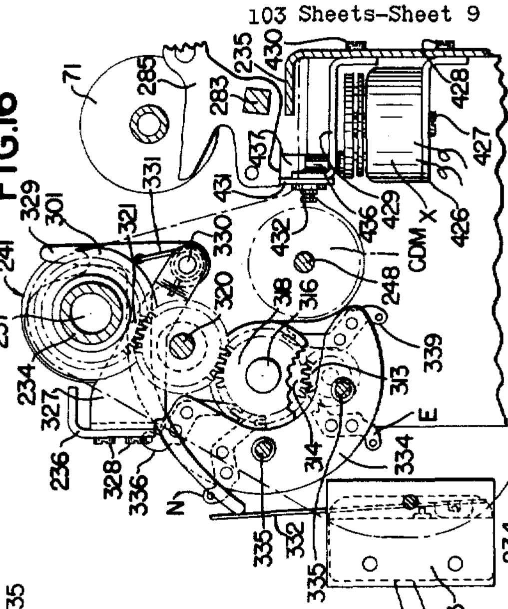
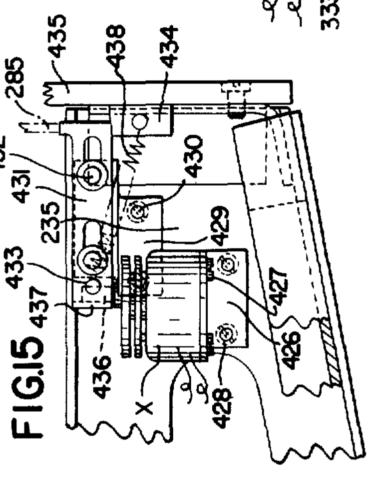


FIG.15



INVENTORS
P. B. CLOSE, J. A. COULTER,
L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK,
E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER,
K. A. KINKER, J. D. LYONS,
N. D. MANOR, R. P. MARVIN,
W. L. MILLER, P. L. O'BRIAN,
J. H. RANDALL, E. M. GARNER,
R. O. SATHER, R. L. YOST &
A. B. BRADEN, JR.

BY Louis A Kline
J Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 10

FIG.17

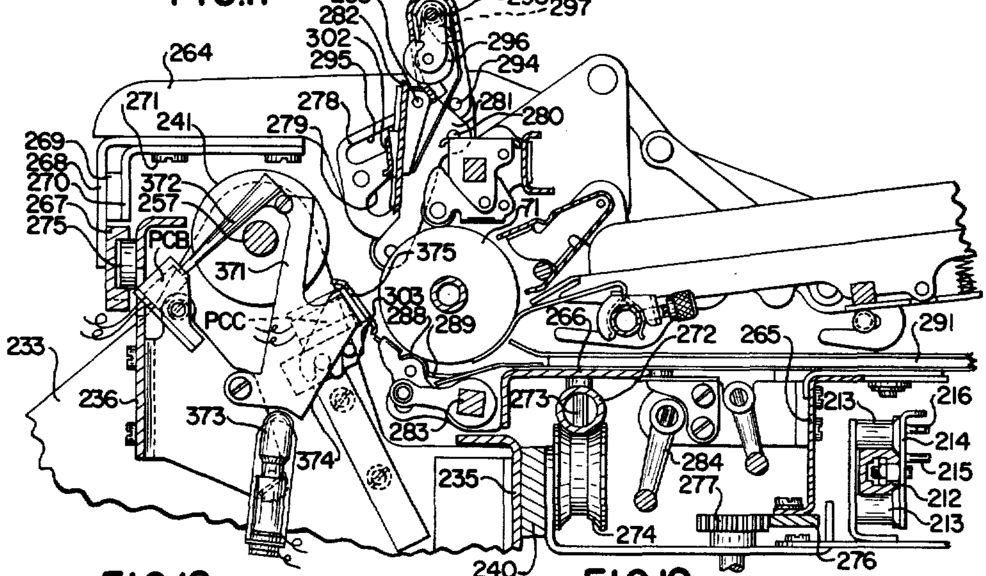


FIG.18

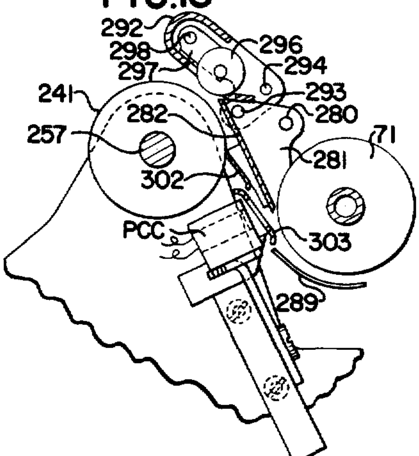


FIG.19

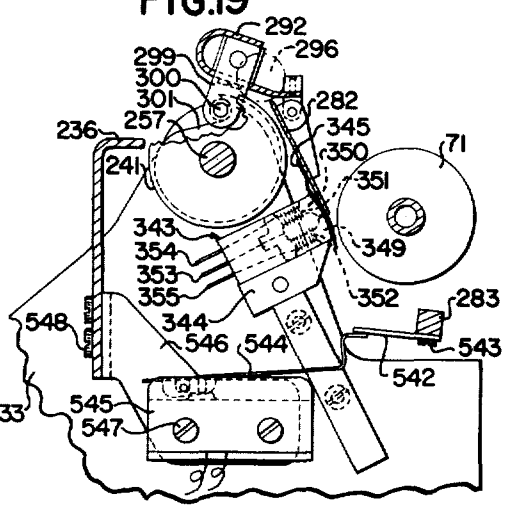


FIG.20

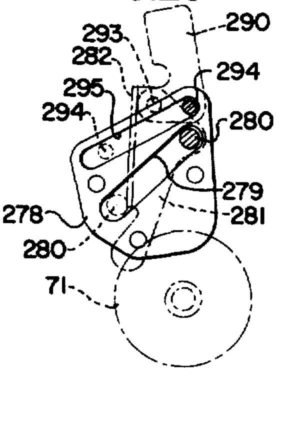
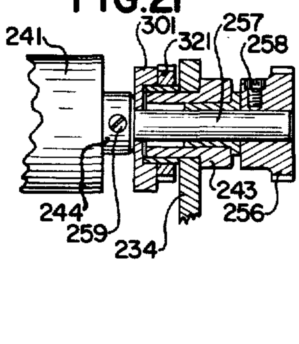


FIG.21



INVENTORS
P. B. CLOSE, J. A. COULTER,
L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK,
E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER,
K. A. KINKER, J. D. LYONS,
N. D. MANOR, R. P. MARVIN,
W. L. MILLER, P. L. O'BRIAN,
J. H. RANDALL, E. M. GARNER,
R. O. SATHER, R. L. YOST &
A. B. BRADEN, JR.

BY Louis A Kline
J Haeunder

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

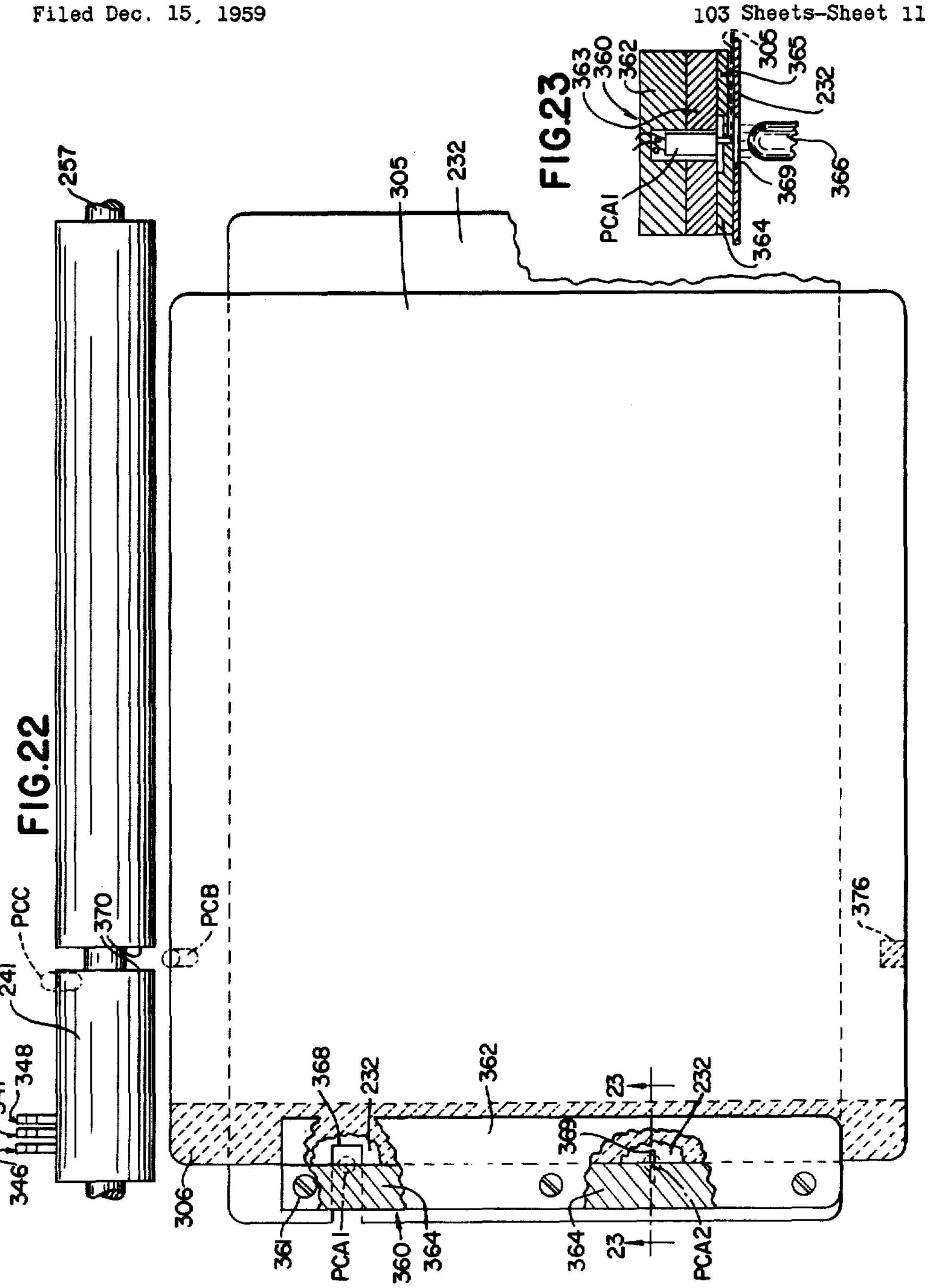


FIG.22

FIG.23

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 12

FIG.24

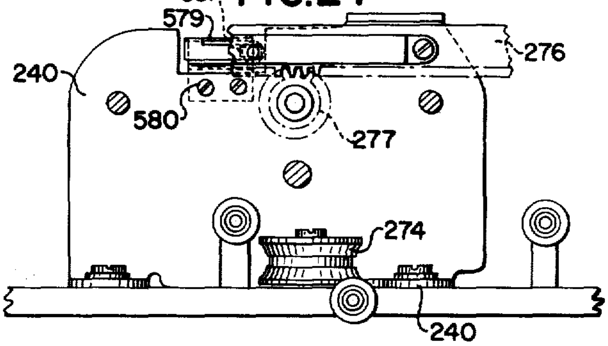


FIG.25

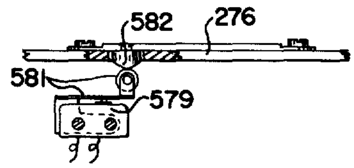


FIG.26

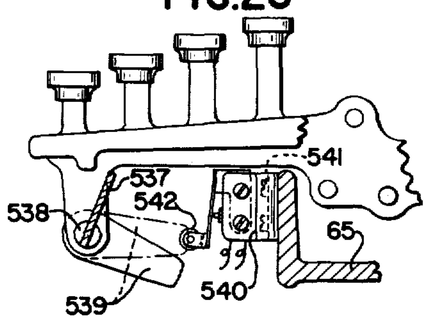


FIG.27

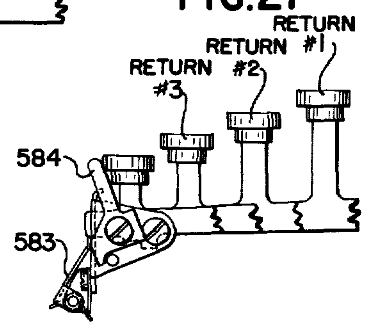


FIG.28A

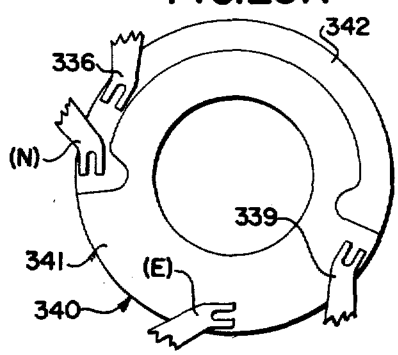
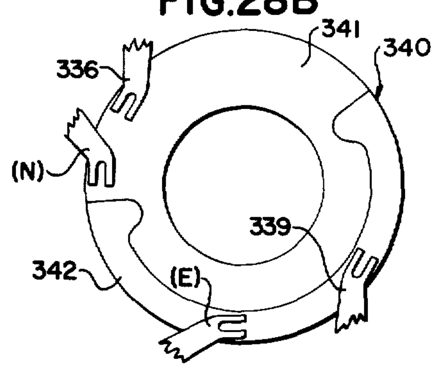


FIG.28B



INVENTORS——P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J. Havender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 13

FIG.29

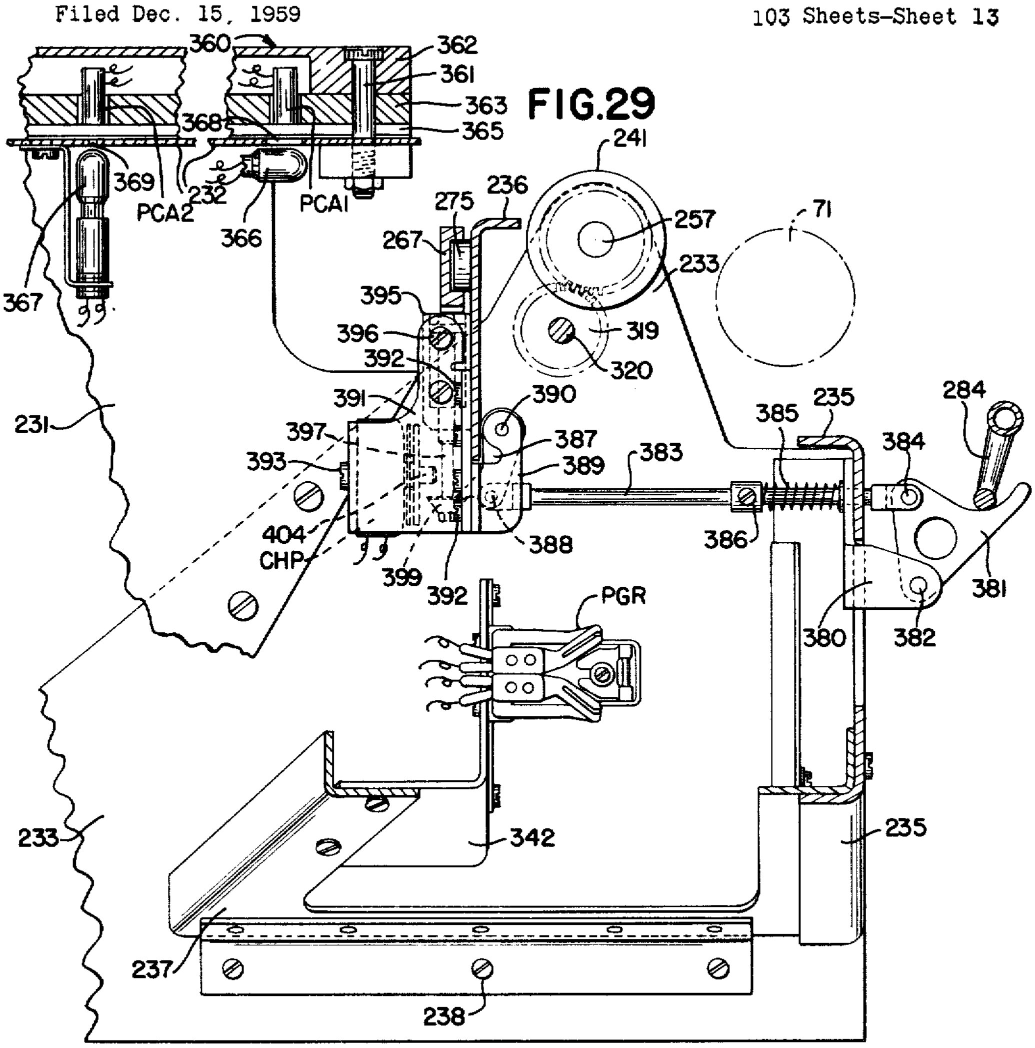


FIG.30

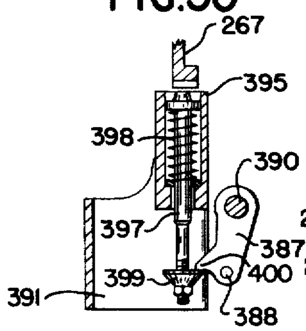
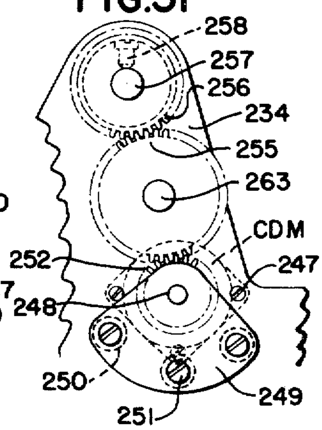


FIG.31



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline

J Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 14

FIG.32

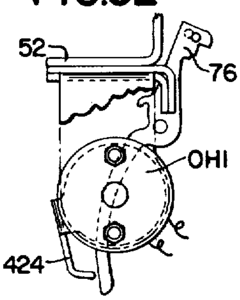


FIG.33

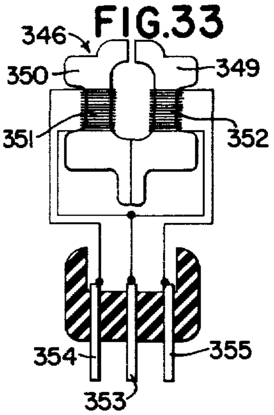


FIG.34

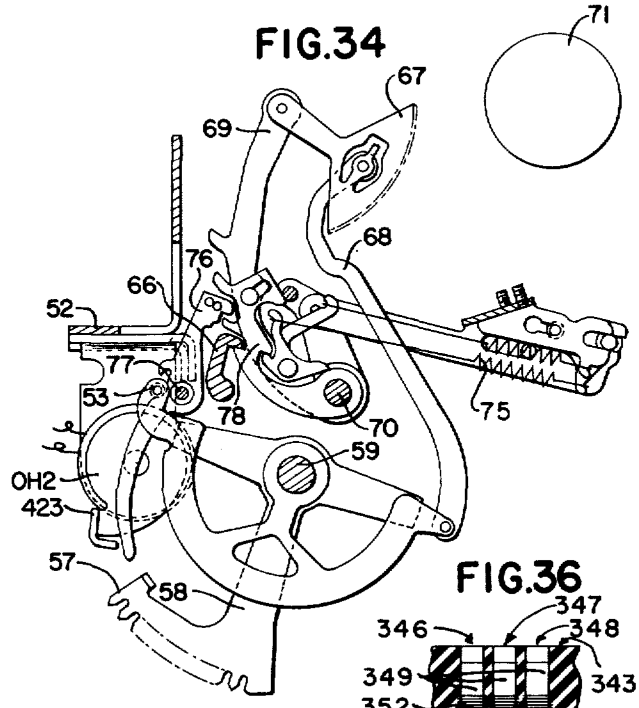


FIG.36

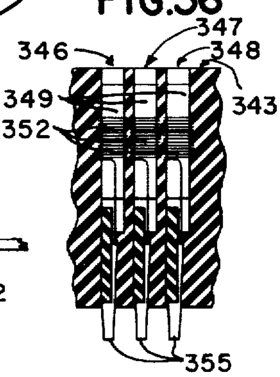


FIG.35

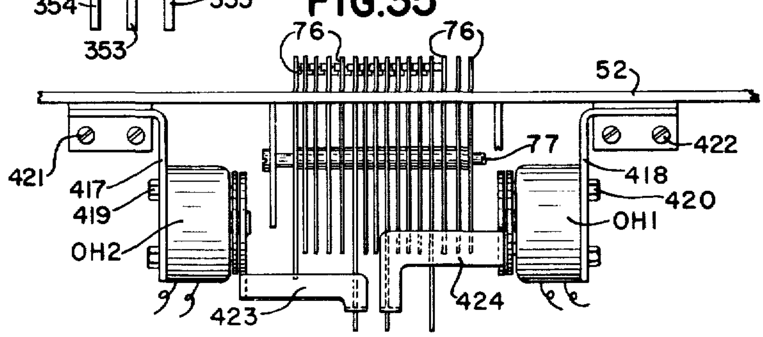
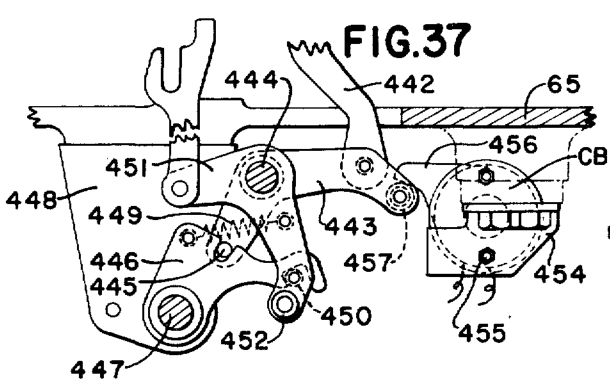


FIG.37



INVENTORS—P. B. CLOSE, J. A. COULTER,
L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK,
E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER,
K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER,
P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER,
R. L. YOST & A. B. BRADEN, JR.

BY

Louis A Kline
J Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 15

FIG.38

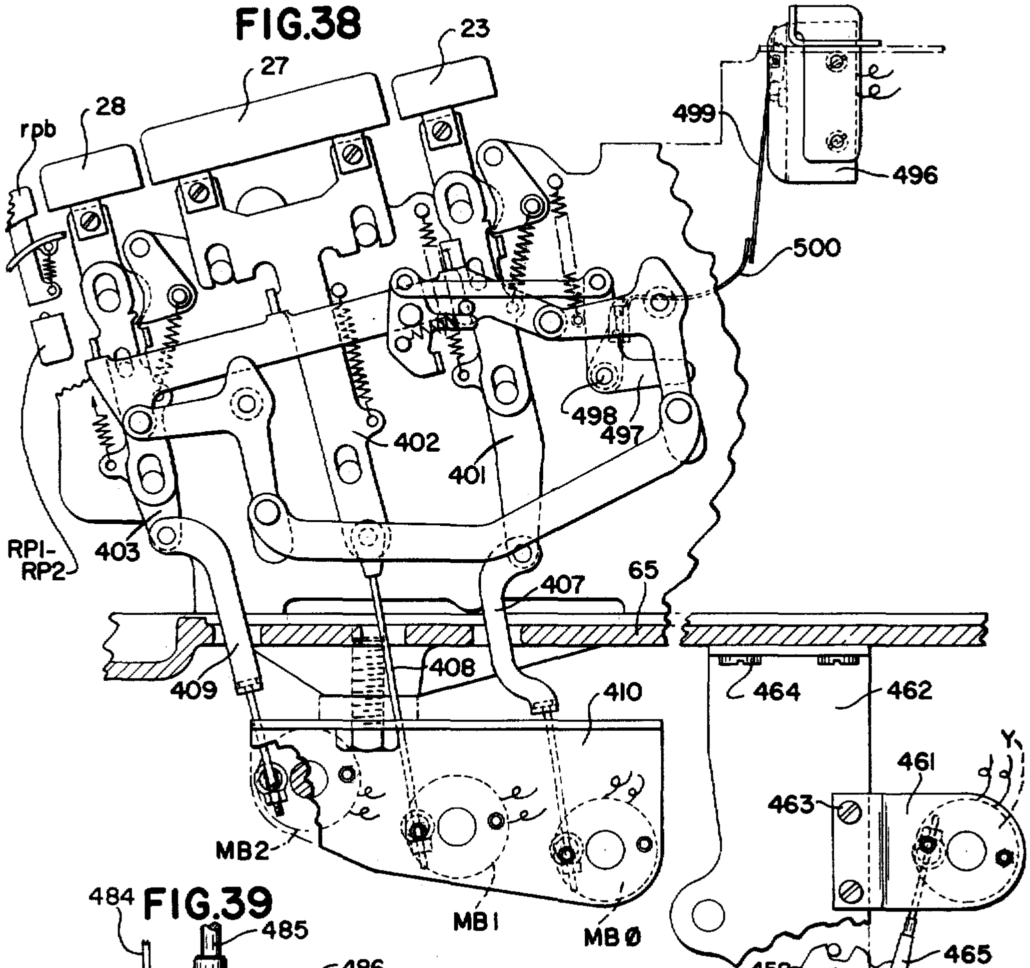


FIG.39

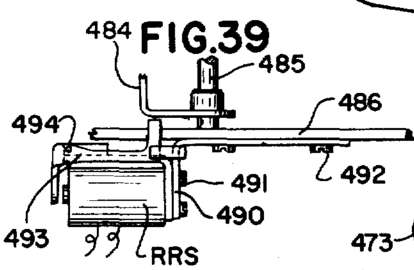
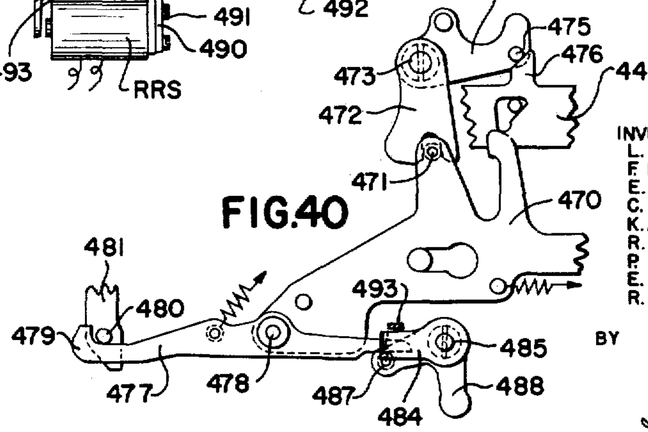


FIG.40



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Hcavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 16

FIG.41

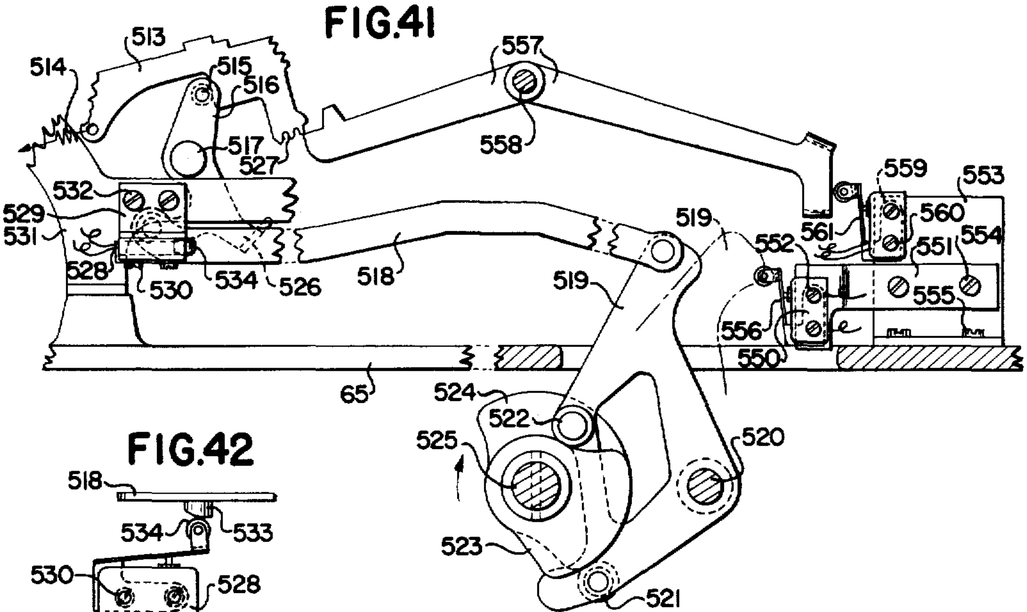


FIG.42

FIG.43

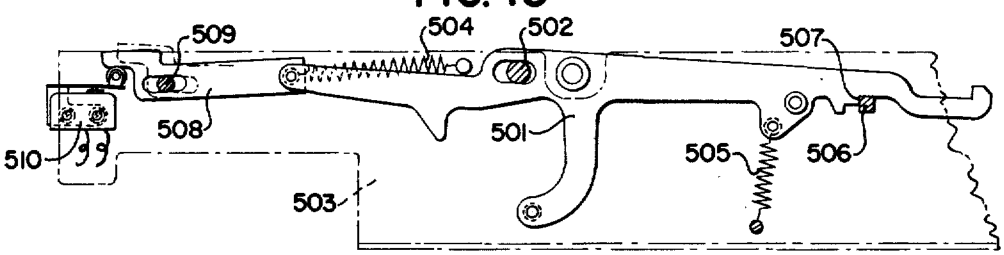
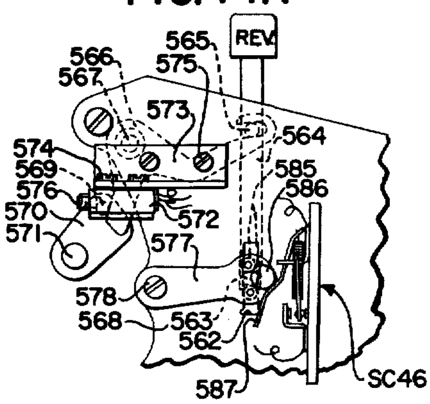


FIG.44A



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Havender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 17

FIG. 44B

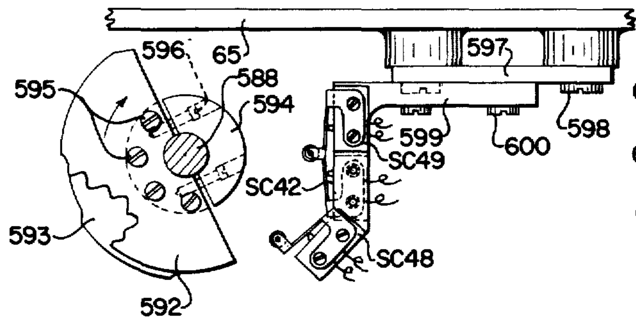


FIG. 44C

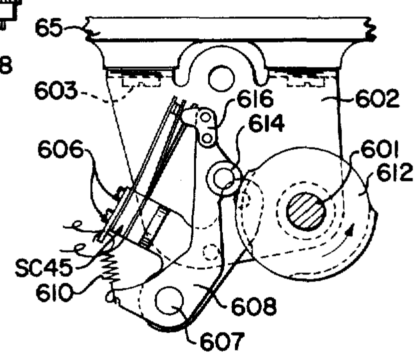


FIG. 44D

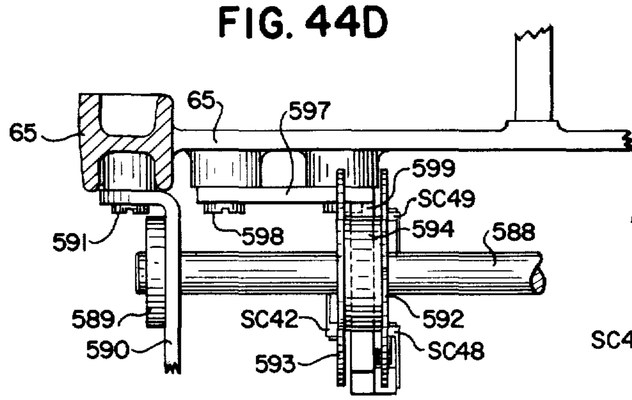


FIG. 44E

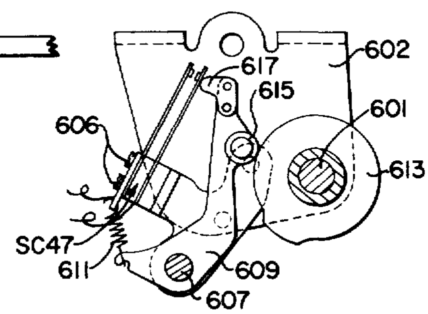


FIG. 44F

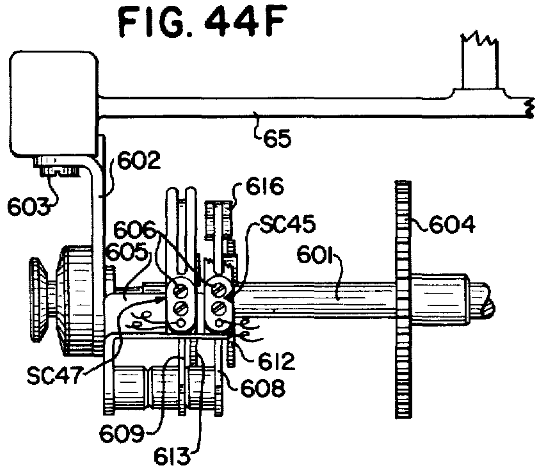
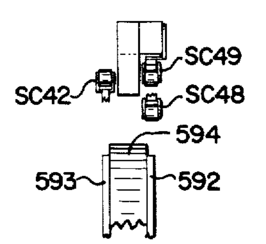


FIG. 44G



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Saunder

THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 18

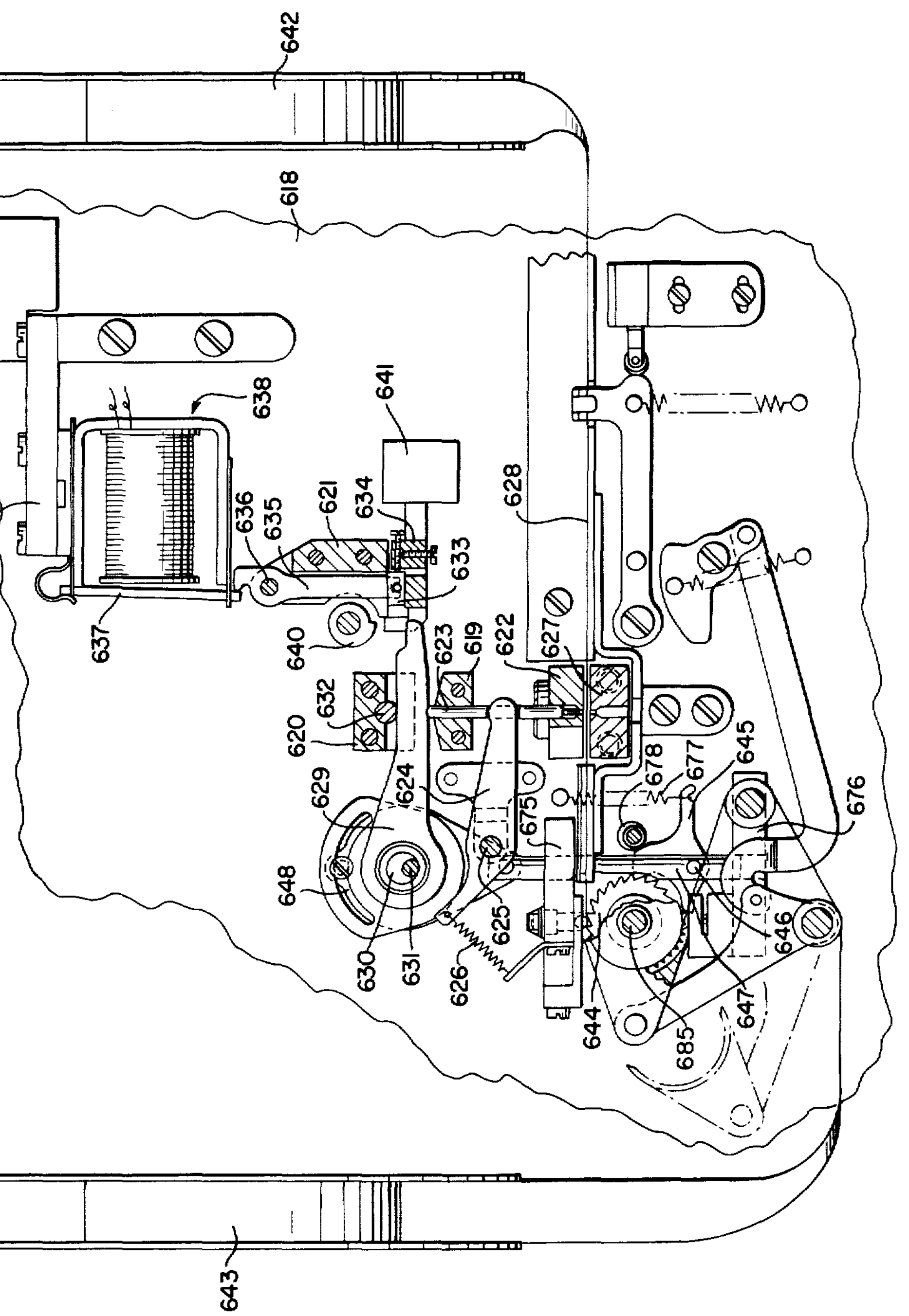


FIG. 44H

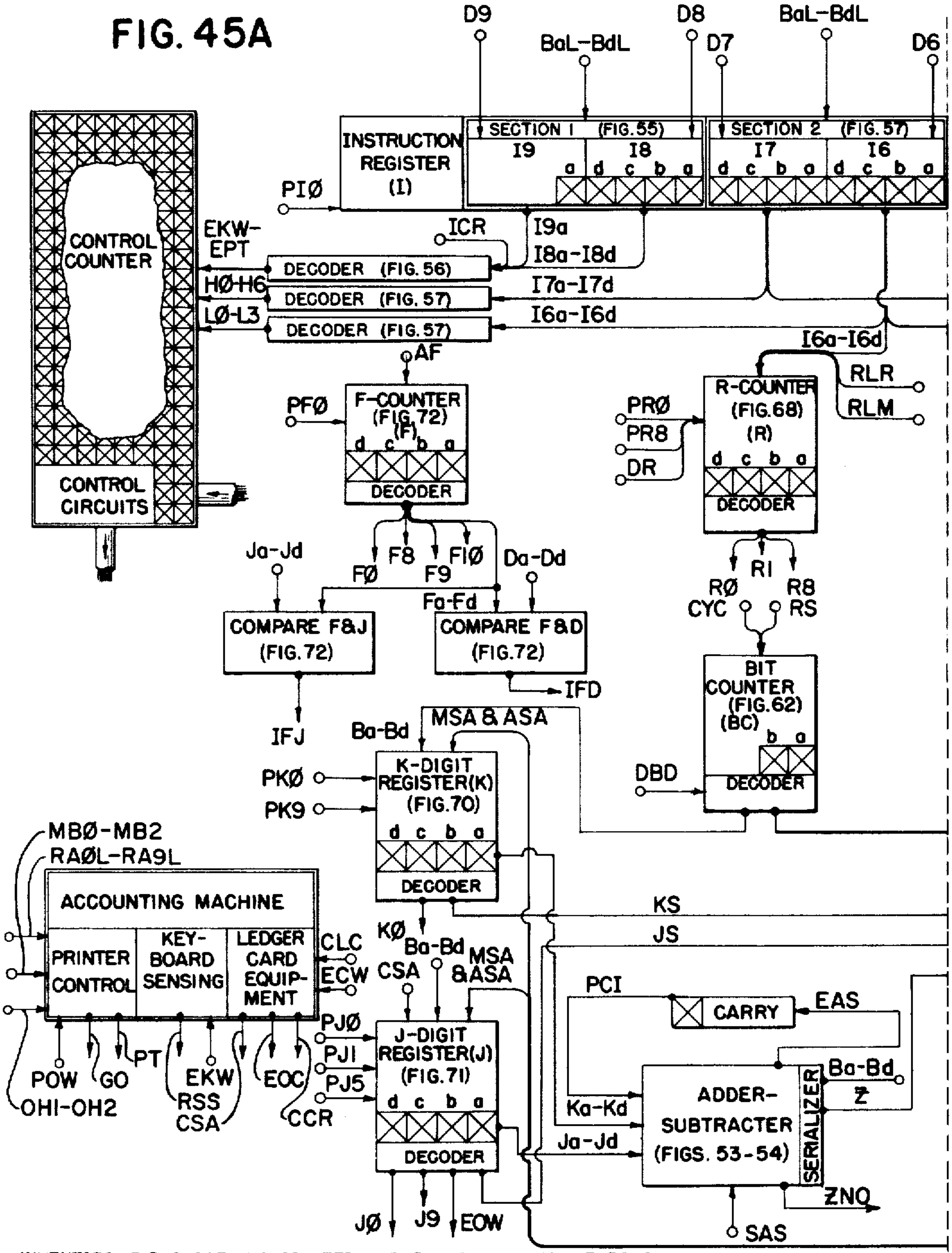
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Lauren Lee

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

FIG. 45A



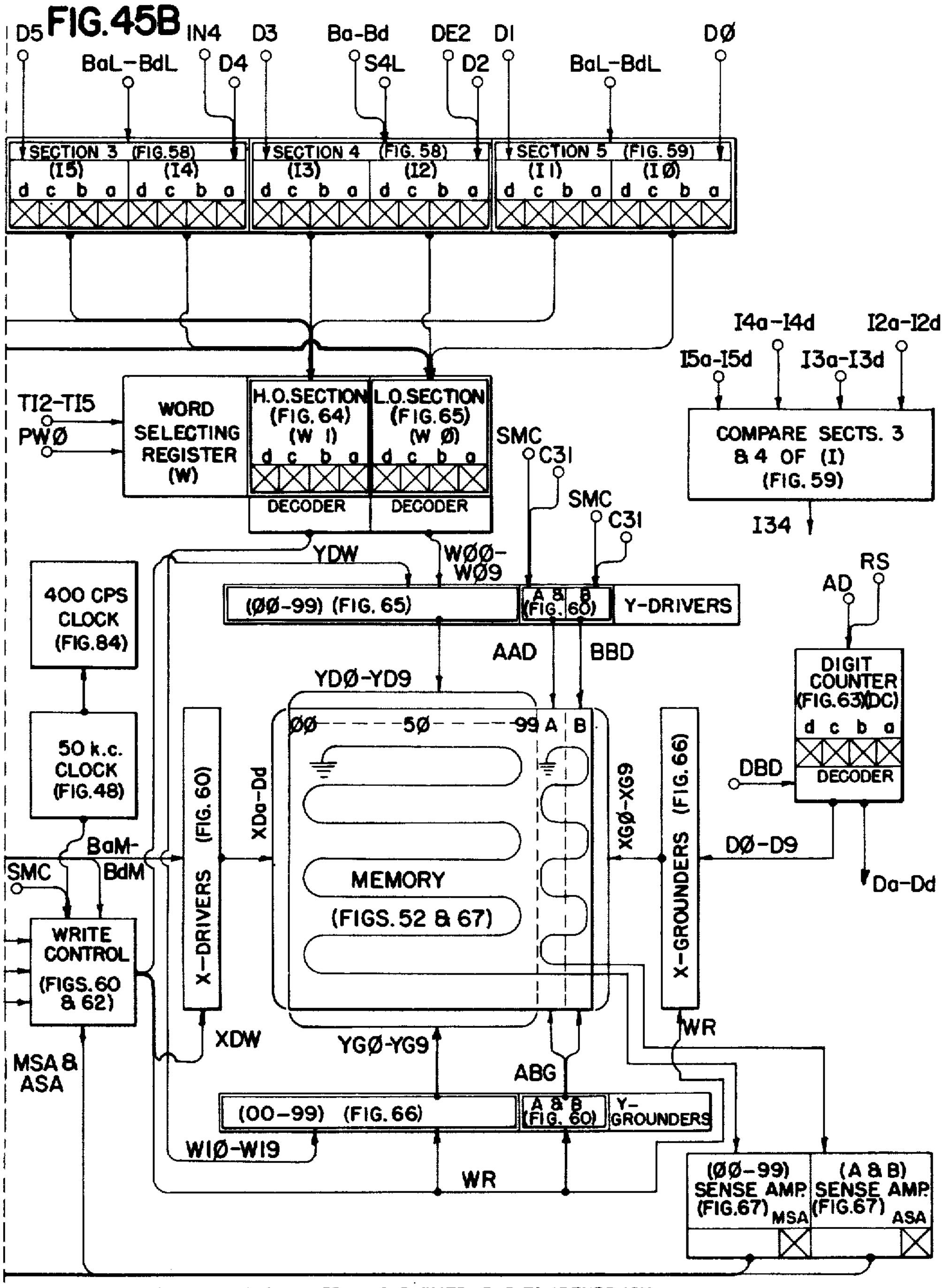
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST, & A. B. BRADEN, JR.

Louis A Kline

BY J. Cavender

THEIR ATTORNEYS

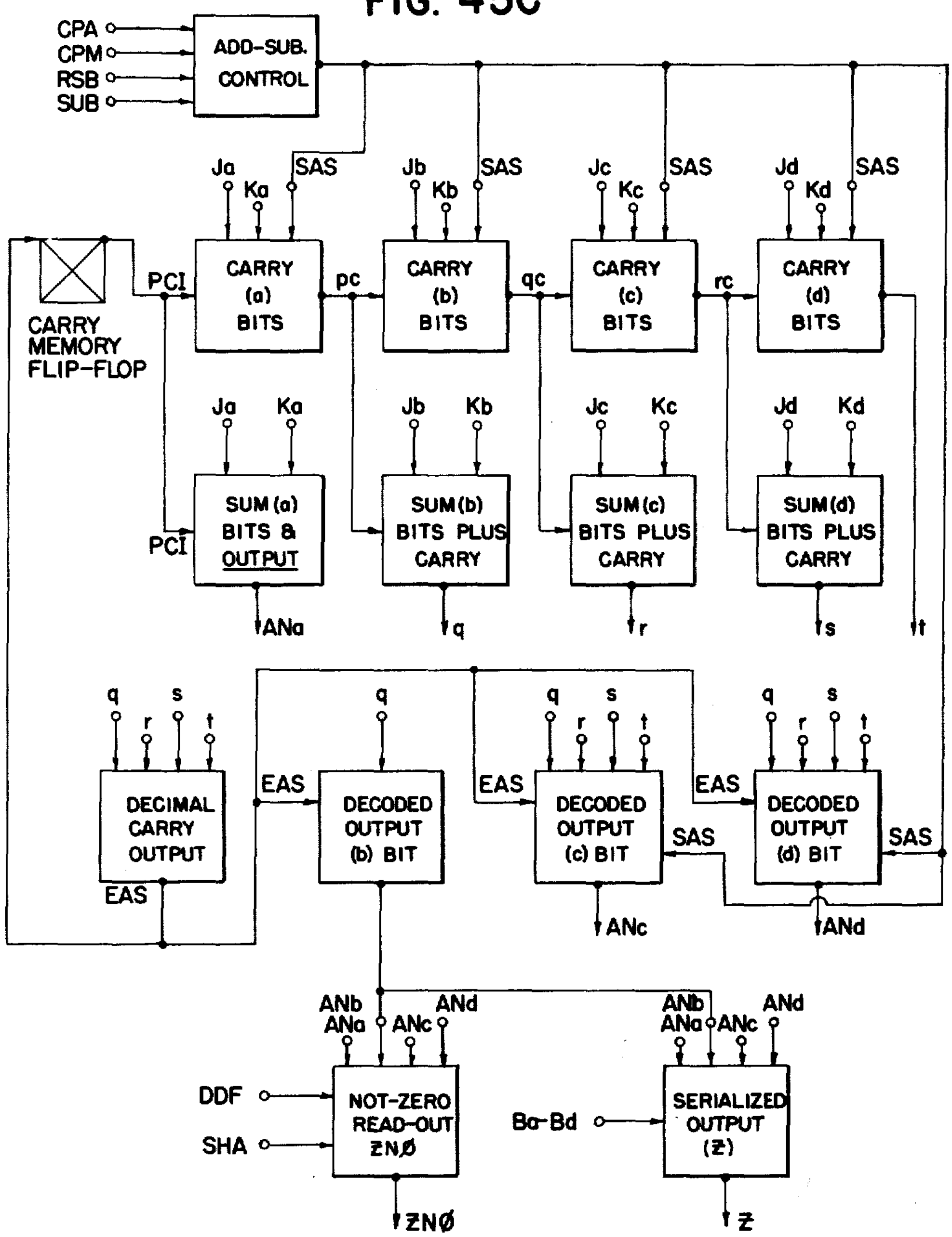
FIG. 45B



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Cavender
THEIR ATTORNEYS

FIG. 45C



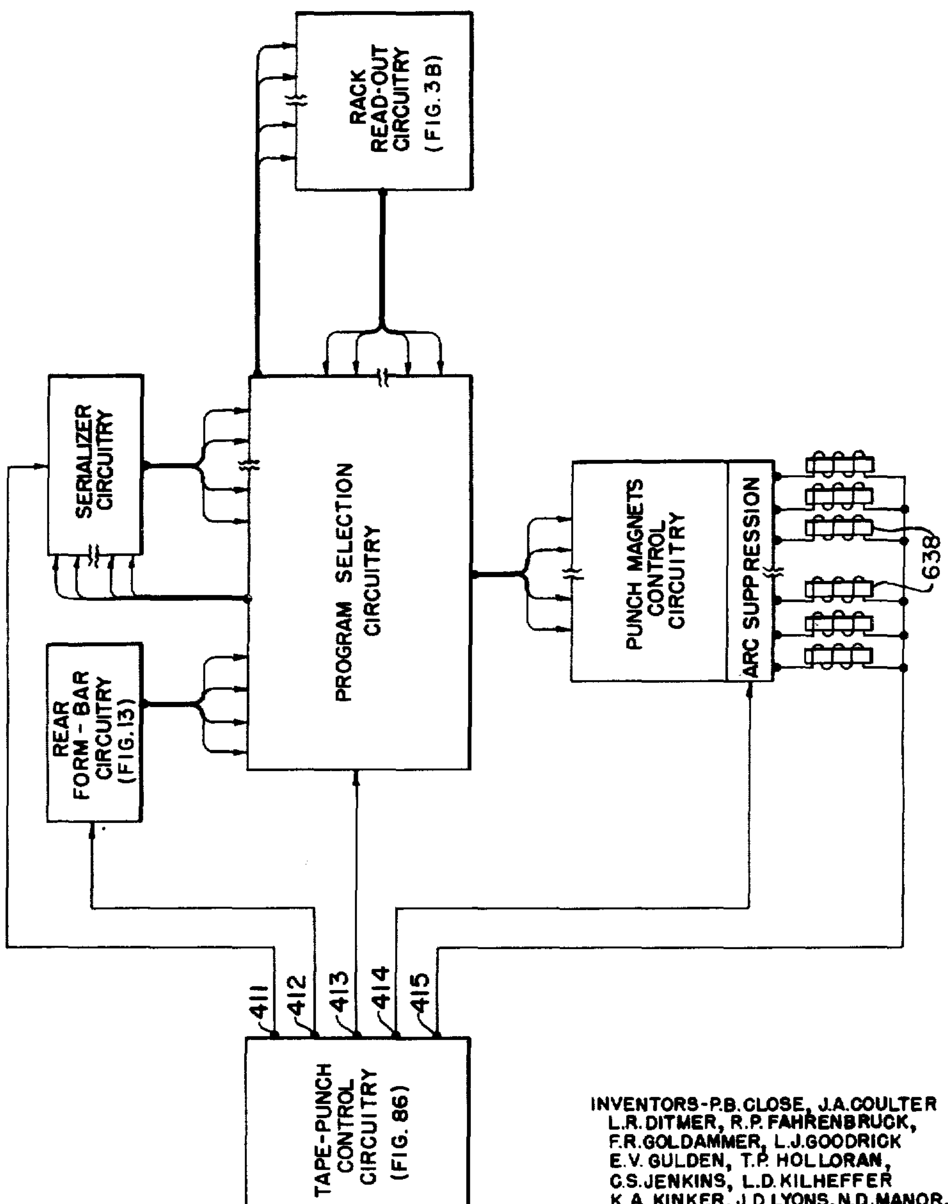
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Hlavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 22

FIG. 45 D

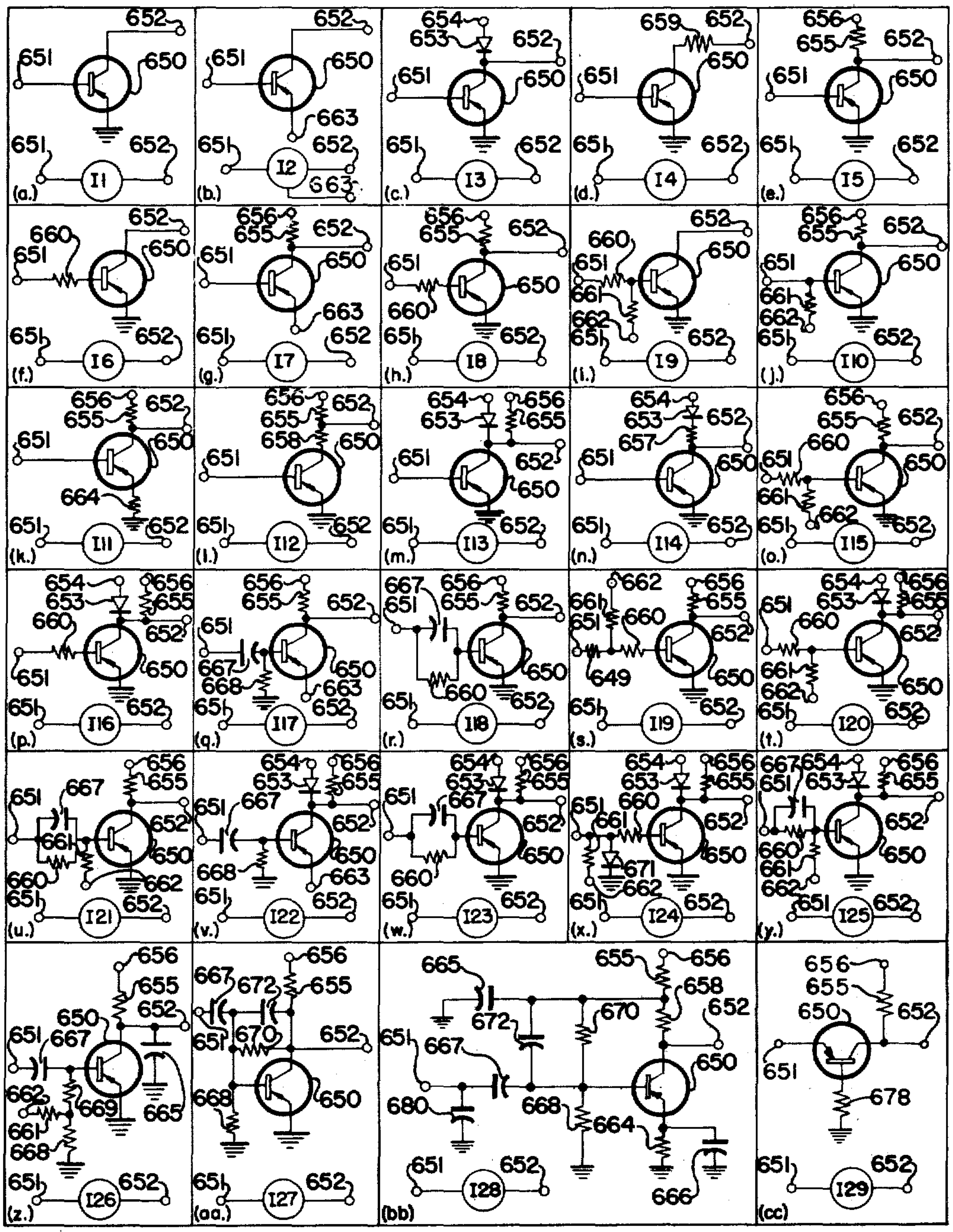


INVENTORS-P.B. CLOSE, J.A. COULTER
L.R. DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOODRICK
E.V. GULDEN, T.P. HOLLORAN,
C.S. JENKINS, L.D. KILHEFFER
K.A. KINKER, J.D. LYONS, N.D. MANOR,
R.P. MARVIN, W.L. MILLER,
P.L. O'BRIAN, J.H. RANDALL,
E.M. GARNER, R.O. SATHER,
R.L. YOST & A.B. BRADEN, JR.

BY

Louis A. Kline
[signature]
THEIR ATTORNEYS

FIG. 46



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR. BY

Louis A Kline

J. Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 24

FIG. 47A

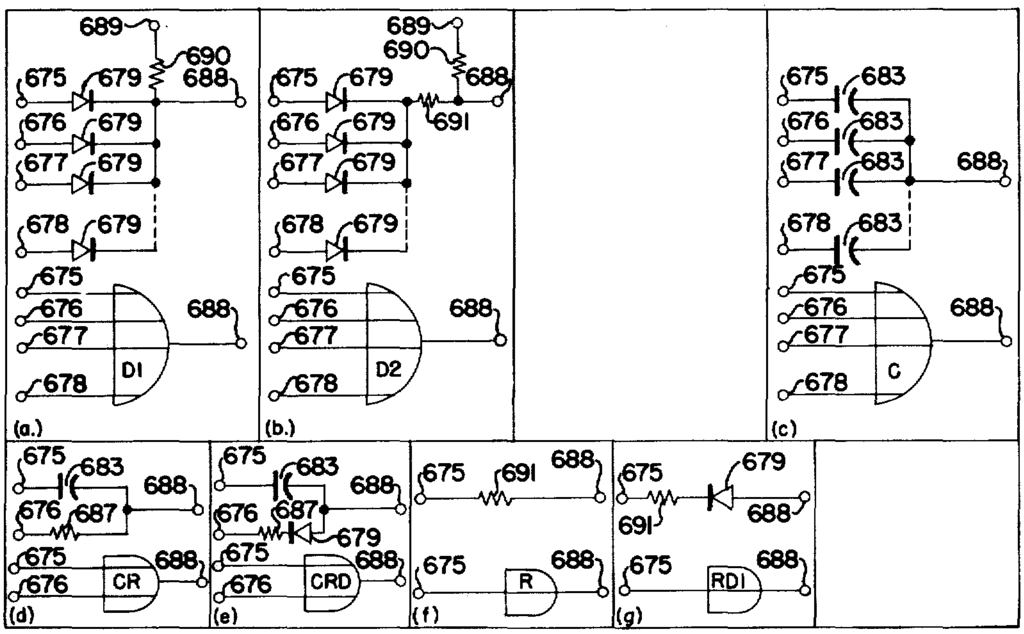
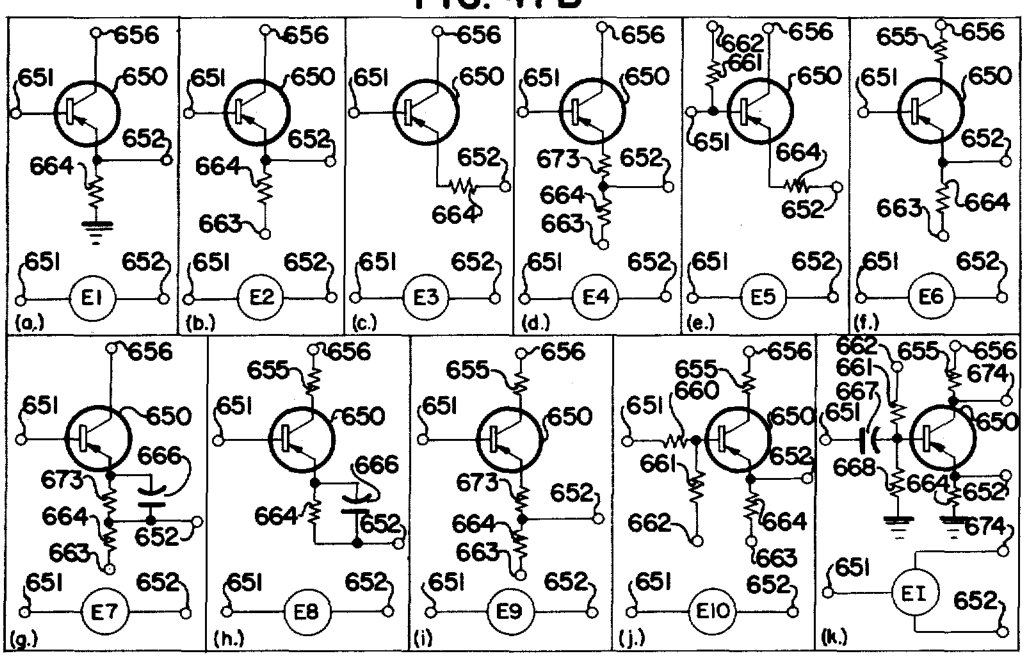


FIG. 47B

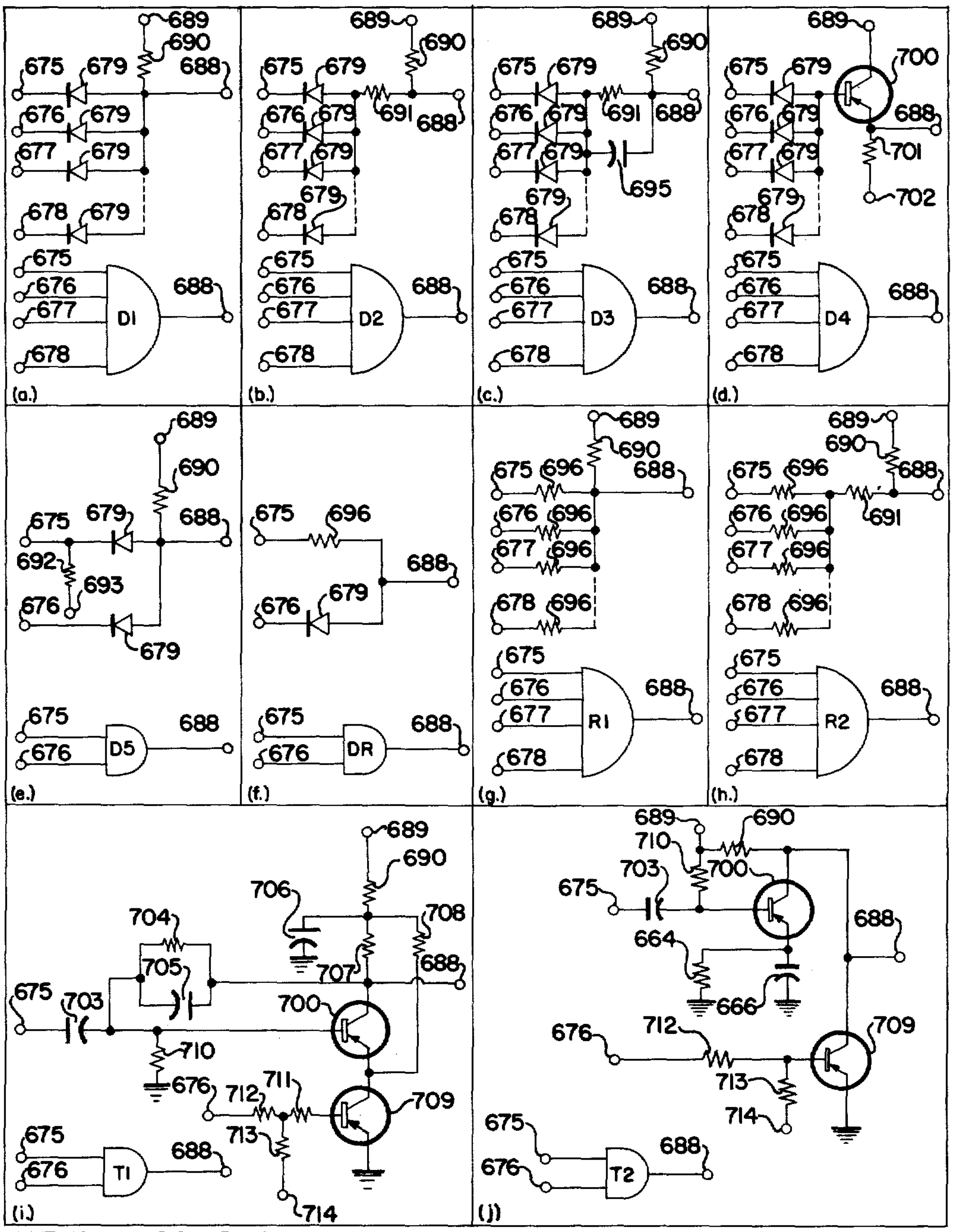


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Cavender

THEIR ATTORNEYS

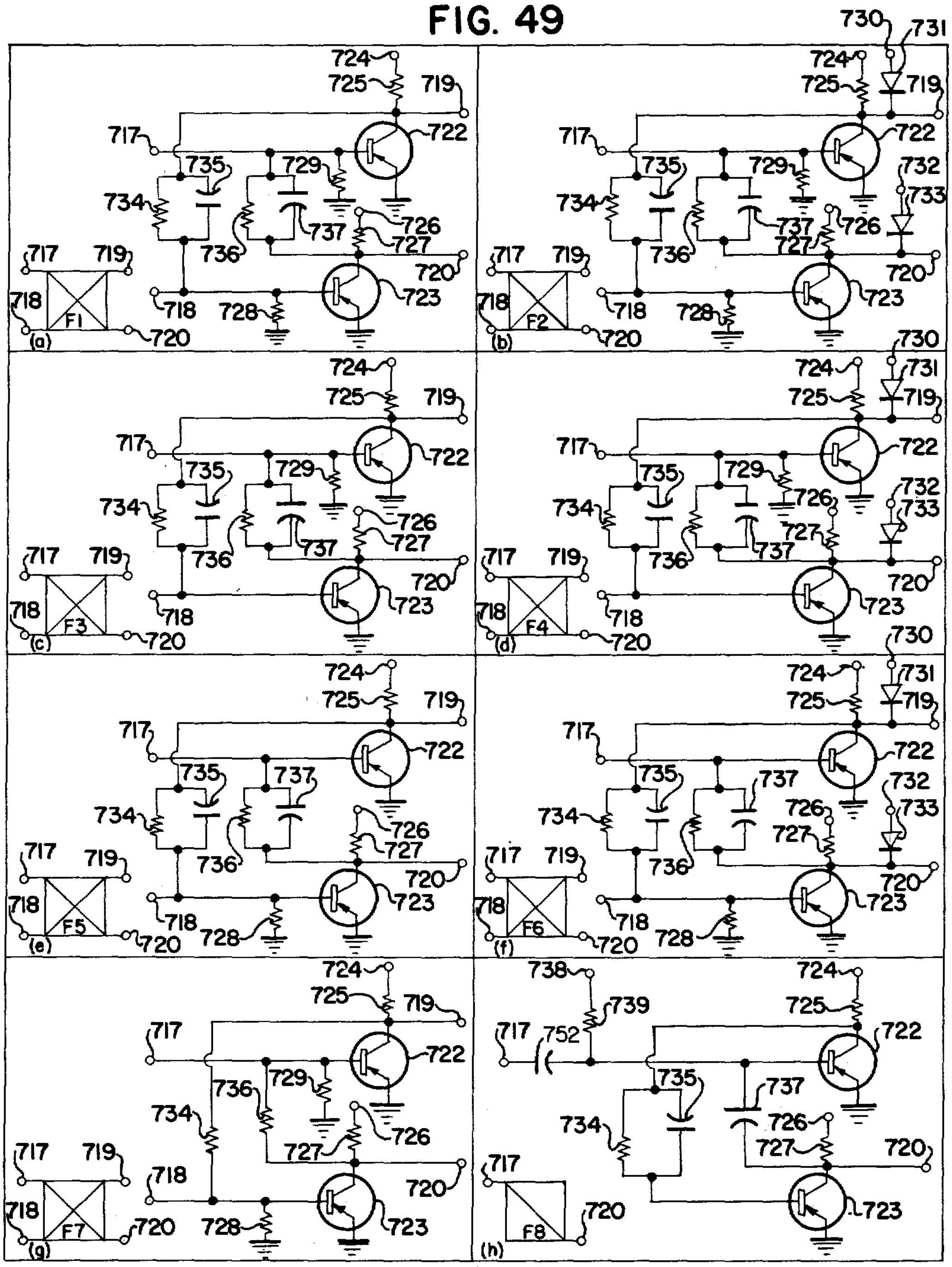
FIG. 48



INVENTORS—P.B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR. BY

Louis A Kline
[signature]
THEIR ATTORNEYS

FIG. 49



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

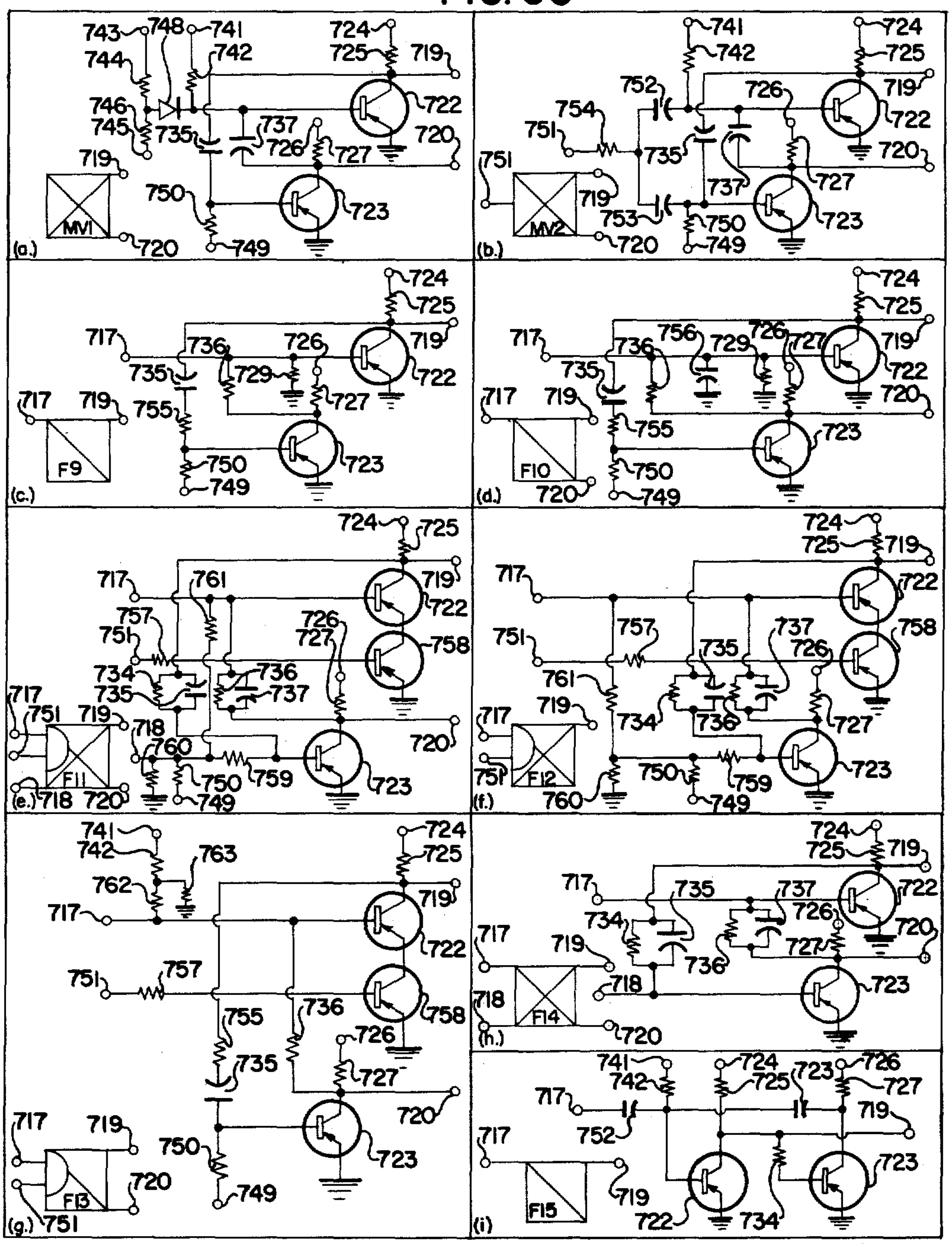
BY Louis A Kline
Hawender
THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 27

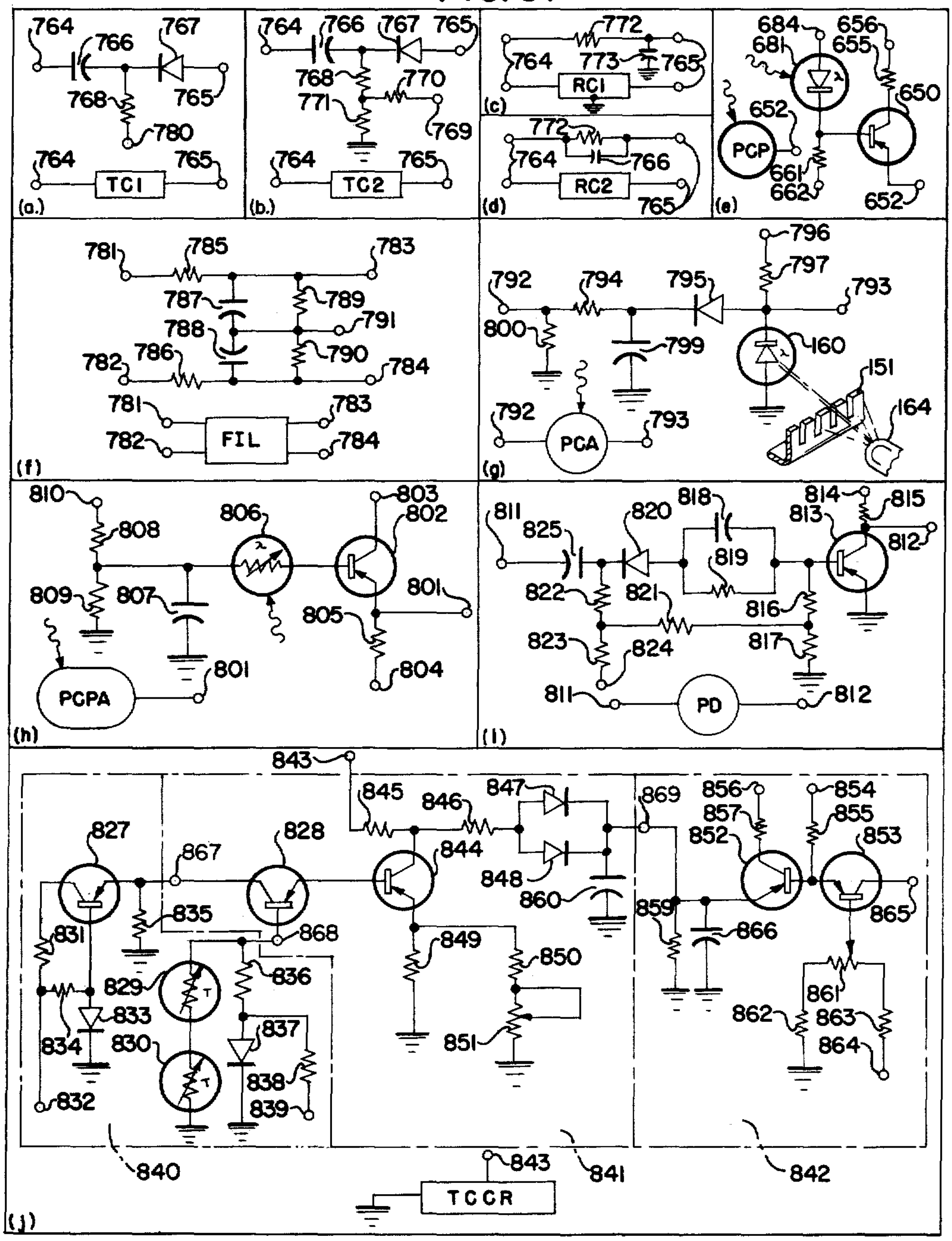
FIG. 50



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
H. Cavender
THEIR ATTORNEYS

FIG. 51



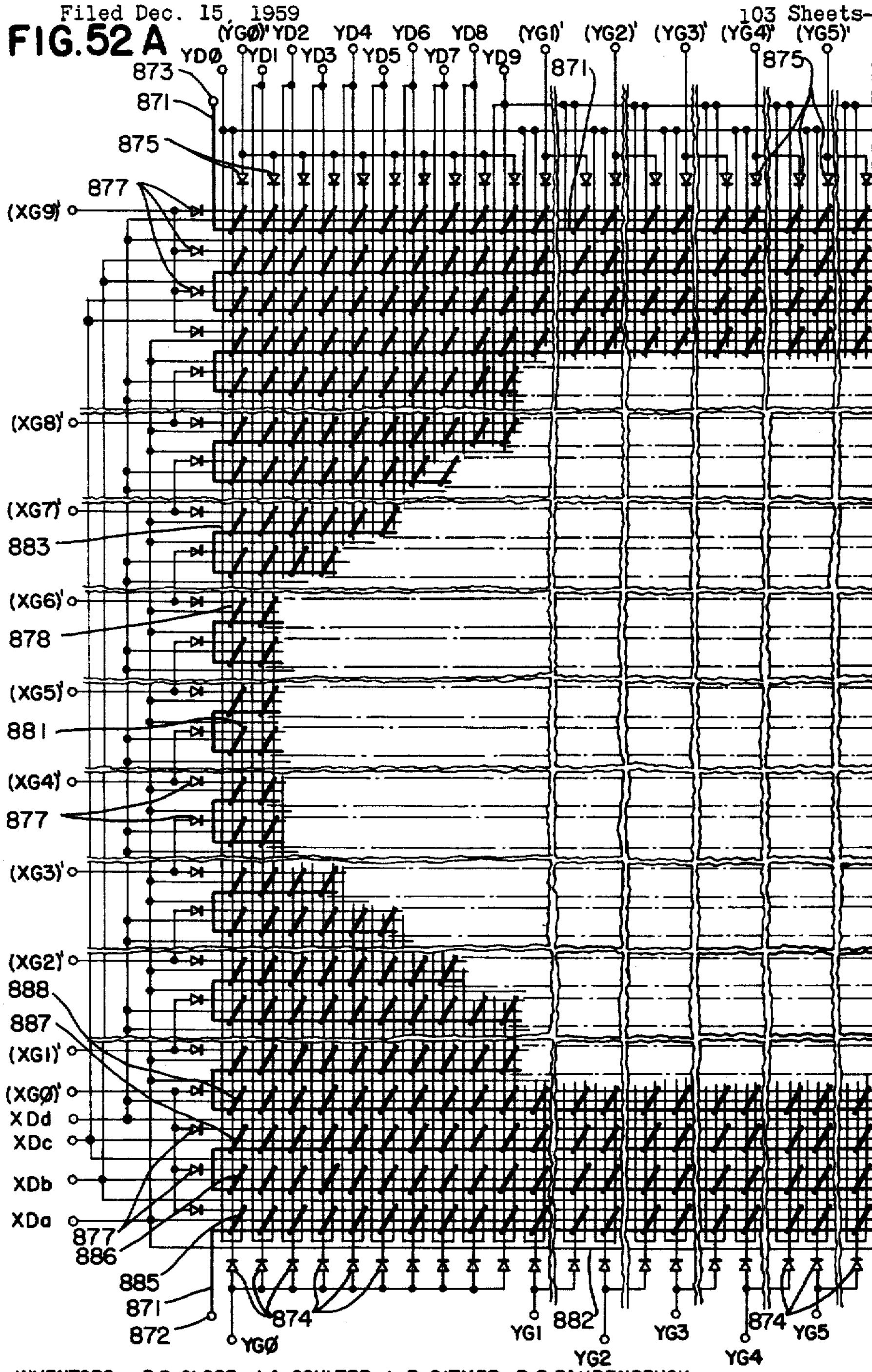
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. [illegible]ST & A. B. BRADEN, JR.

BY Louis A Kline
Havender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 29

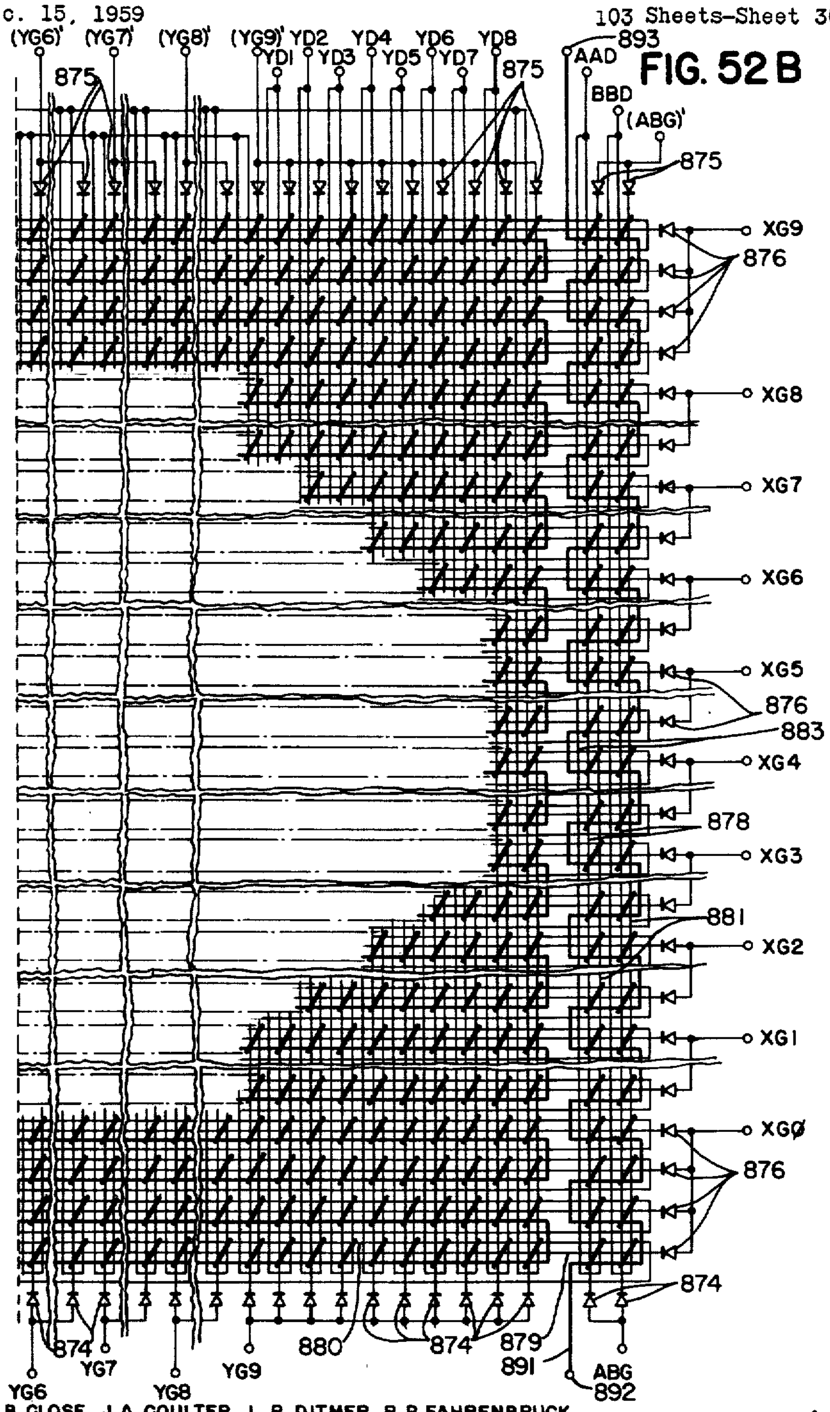
FIG.52A



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Stavender
THEIR ATTORNEYS

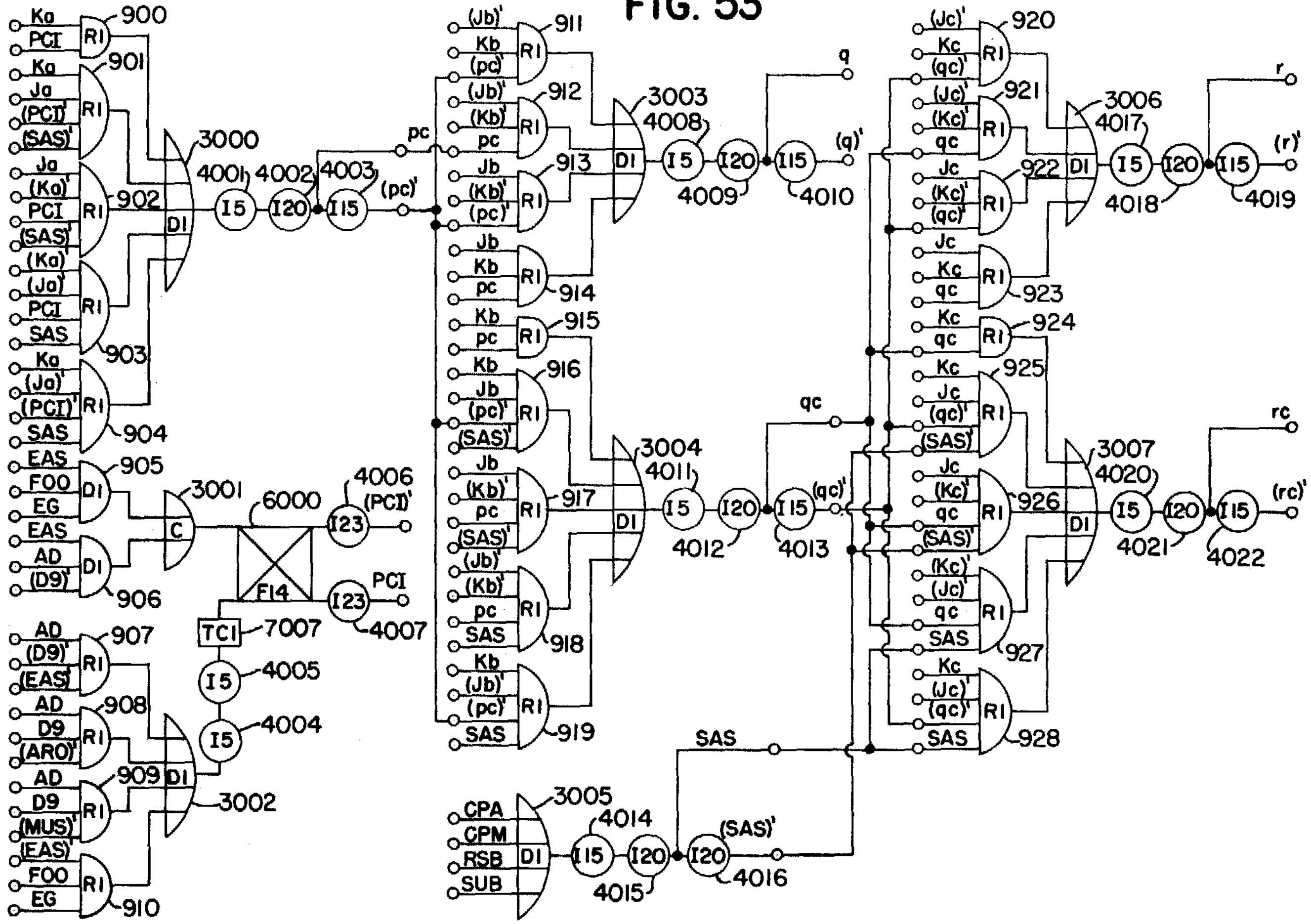
FIG. 52 B



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

FIG. 53



INVENTORS—P.B.CLOSE, J.A.COULTER, L.R.DITMER, R.P.FAHRENBRUCK, F.R.GOLDAMMER, L.J.GOODRICK, E.V.GULDEN, T.P.HOLLORAN, C.S.JENKINS, L.D.KILHEFFER, K.A.KINKER, J.D.LYONS, N.D.MANOR, R.P.MARVIN, W.L.MILLER, P.L.O'BRIAN, J.H.RANDALL, E.M.GARNER, R.O.SATHER, R.L.YOST & A.B.BRADEN, JR.

BY [signature]

THEIR ATTORNEYS

Nov. 26, 1963

P. B. CLOSE ETAL

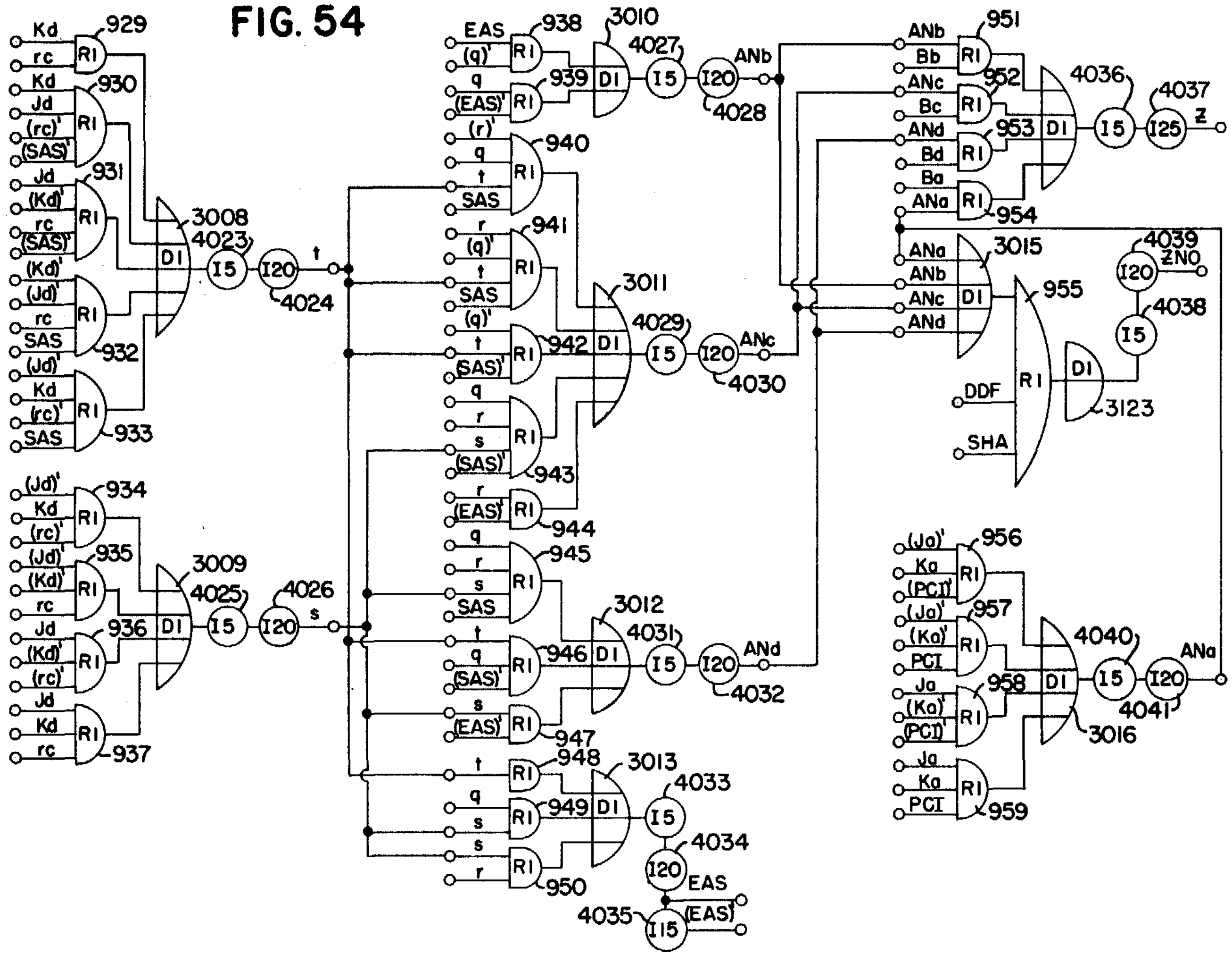
3,112,394

Filed Dec. 15, 1959

ELECTRONIC COMPUTING MACHINE

103 Sheets-Sheet 32

FIG. 54



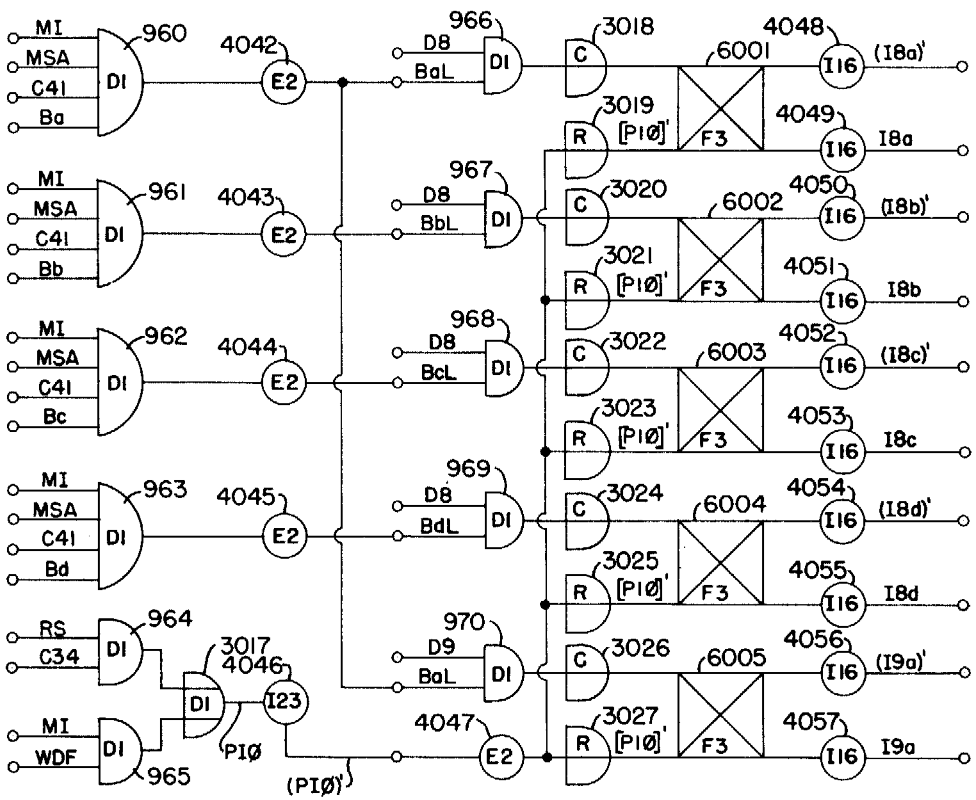
INVENTORS—P.B.CLOSE, J.A.COULTER, L.R.DITMER, R.P.FAHRENBRUCK, F.R.GOLDAMMER, L.J.GOODRICK, E.V.GULDEN, T.P.HOLLORAN, C.S.JENKINS, L.D.KILHEFFER, K.A.KINKER, J.D.LYONS, N.D.MANOR, R.P.MARVIN, W.L.MILLER, P.L.O'BRIAN, J.H.RANDALL, E.M.GARNER, R.O.SATHER, R.L.YOST & A.B.BRADEN, JR.

BY

THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394
Filed Dec. 15, 1959 ELECTRONIC COMPUTING MACHINE 103 Sheets-Sheet 33

FIG. 55



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

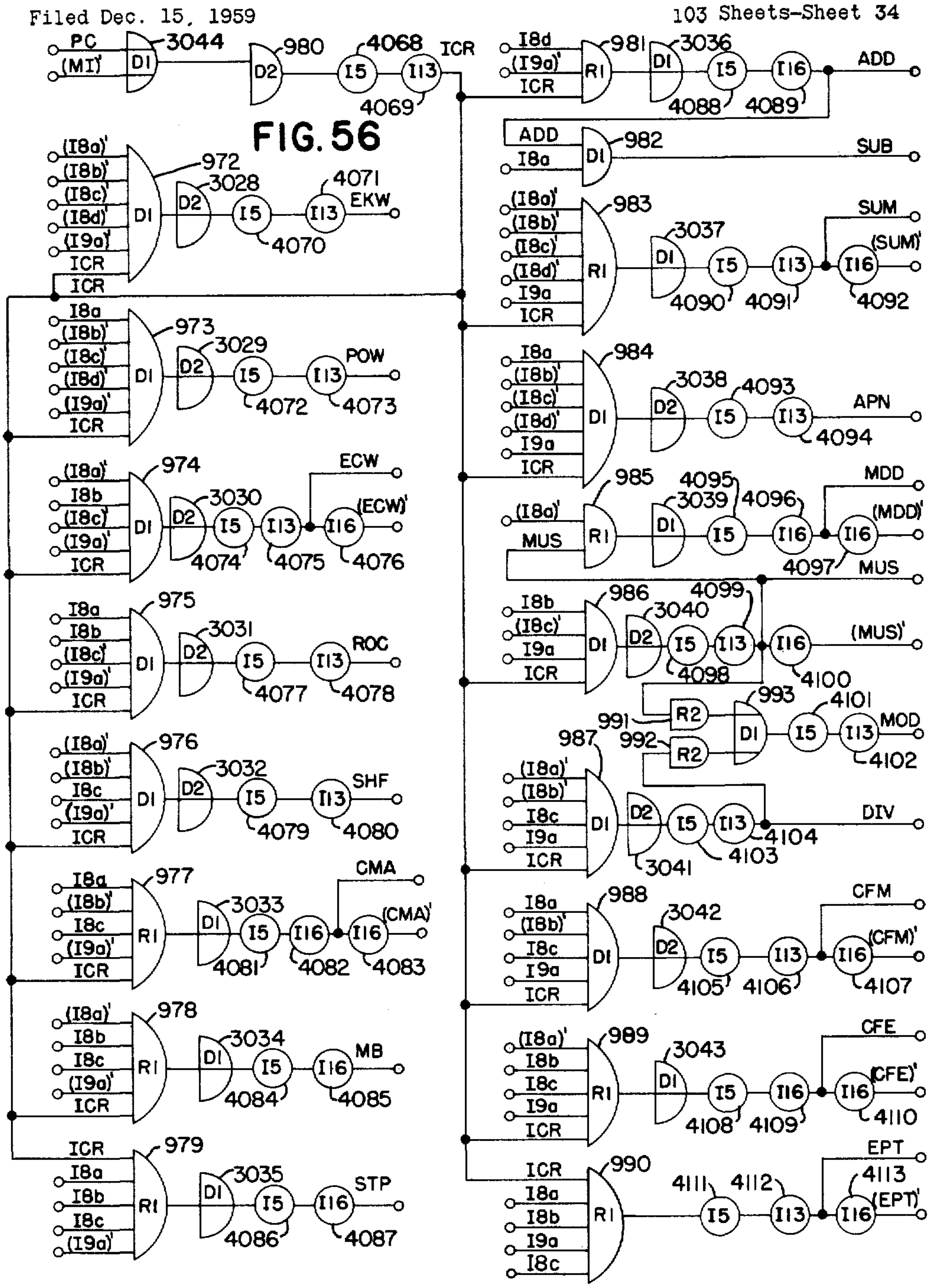


FIG. 56

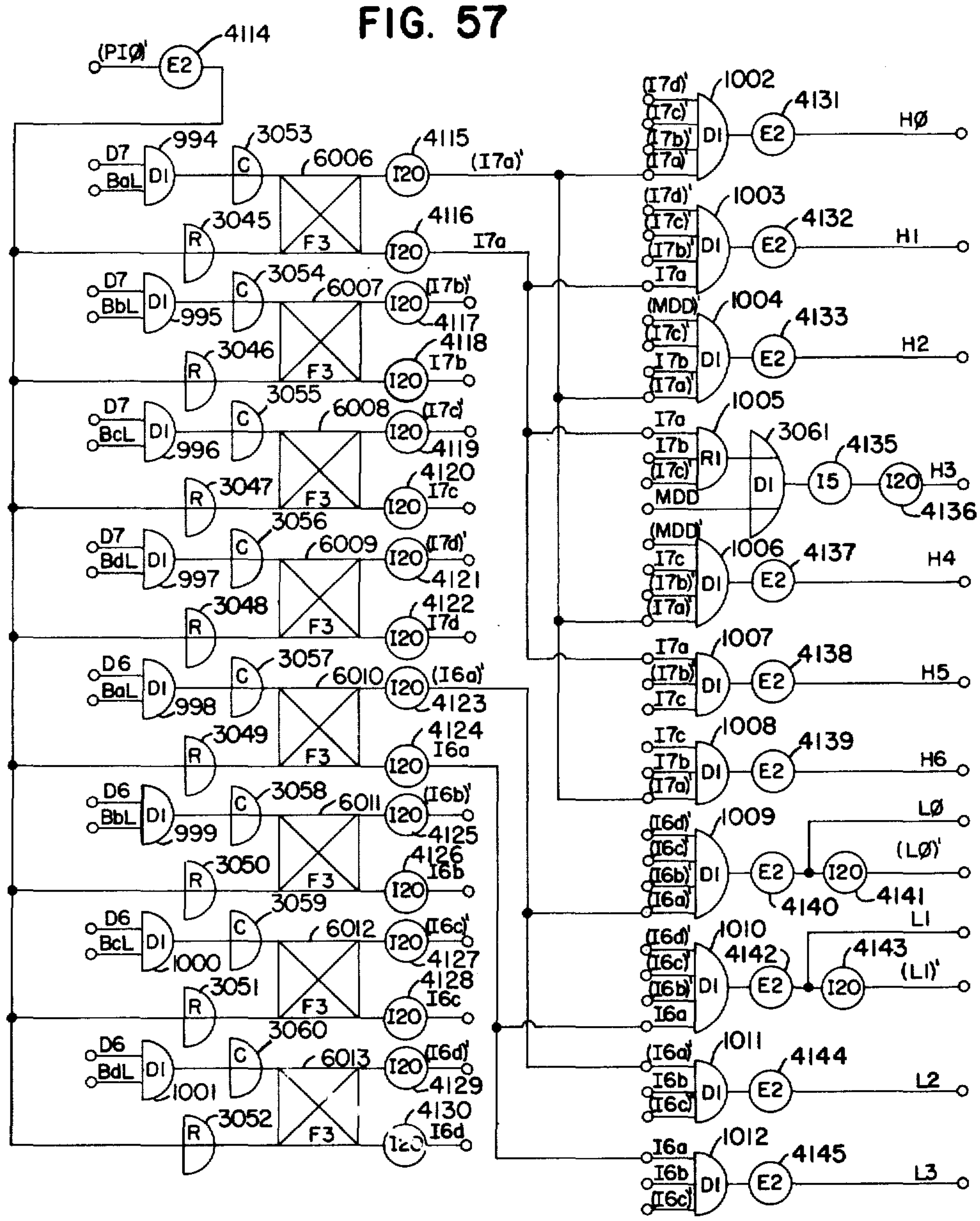
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Cavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 35

FIG. 57



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Bavender
THEIR ATTORNEYS

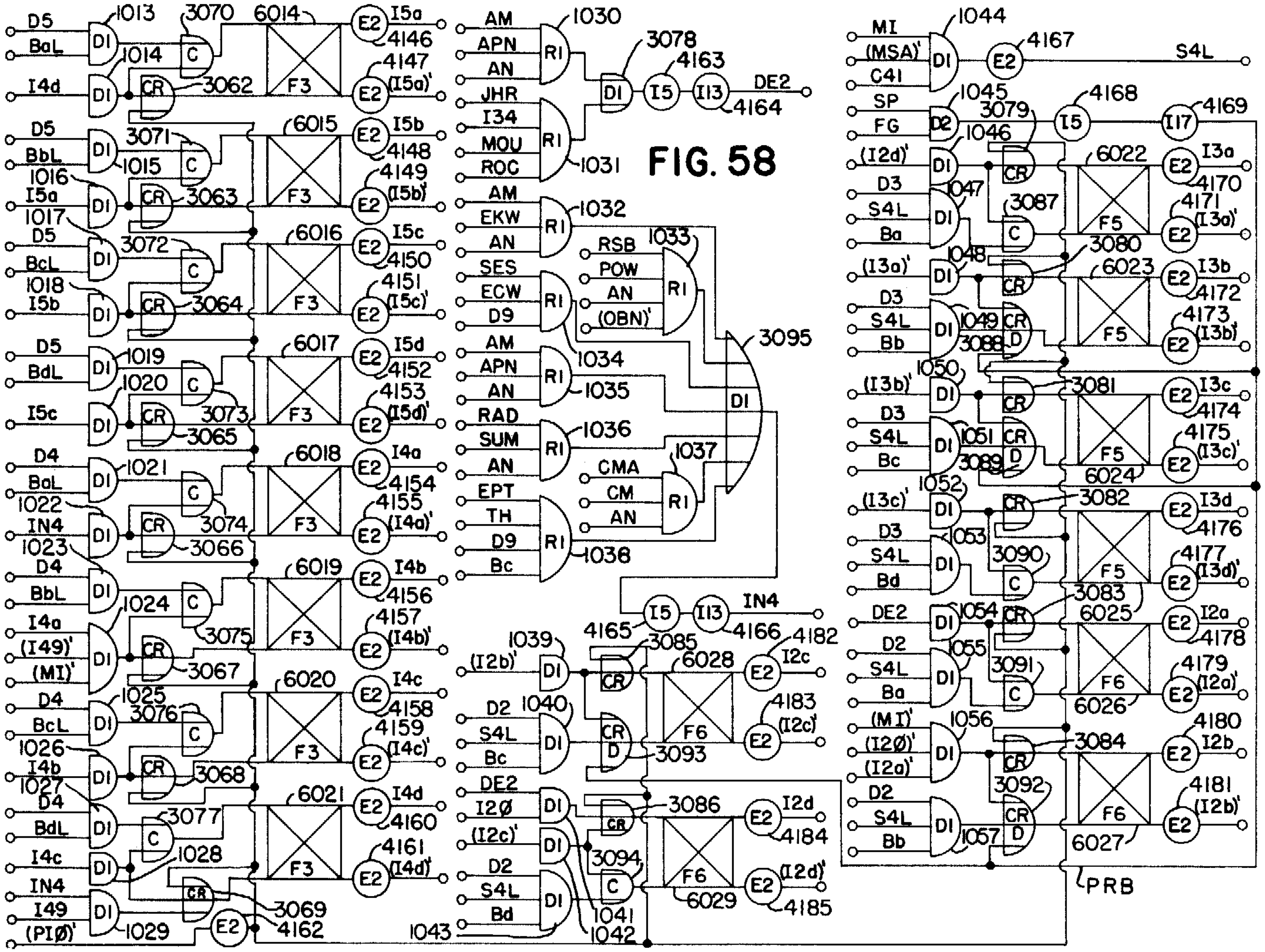
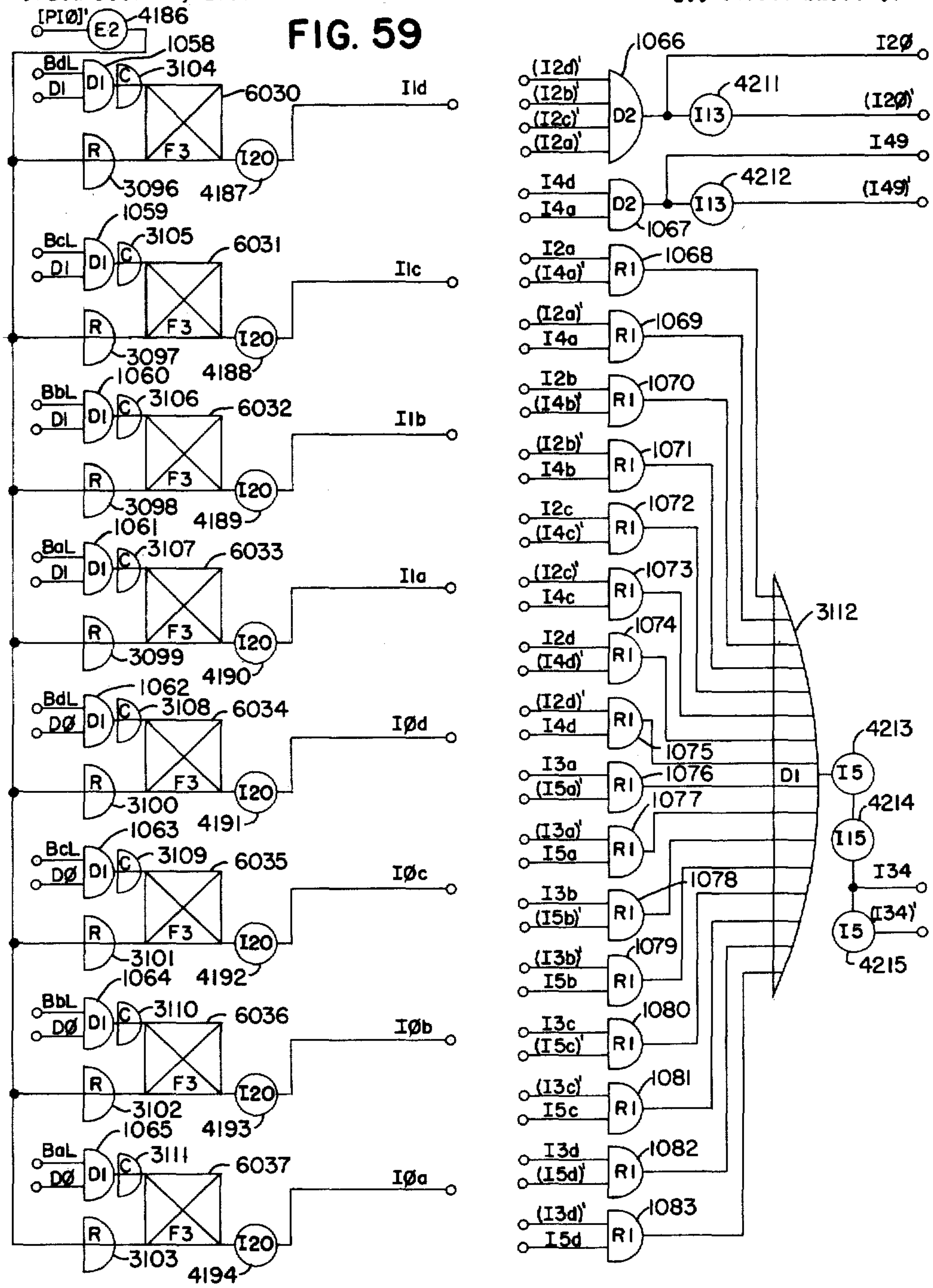


FIG. 58

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

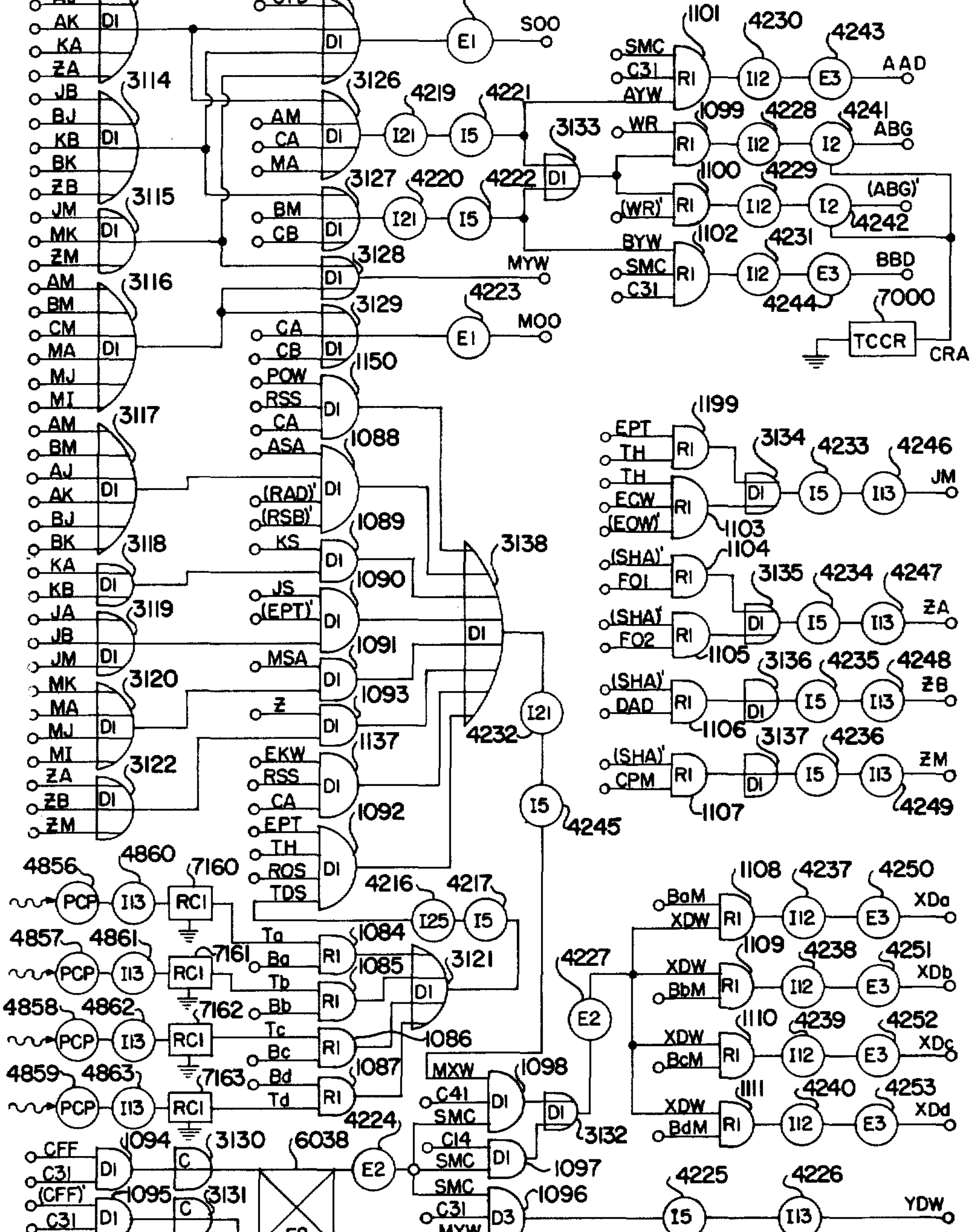
FIG. 59



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Hoavender
THEIR ATTORNEYS

FIG. 60

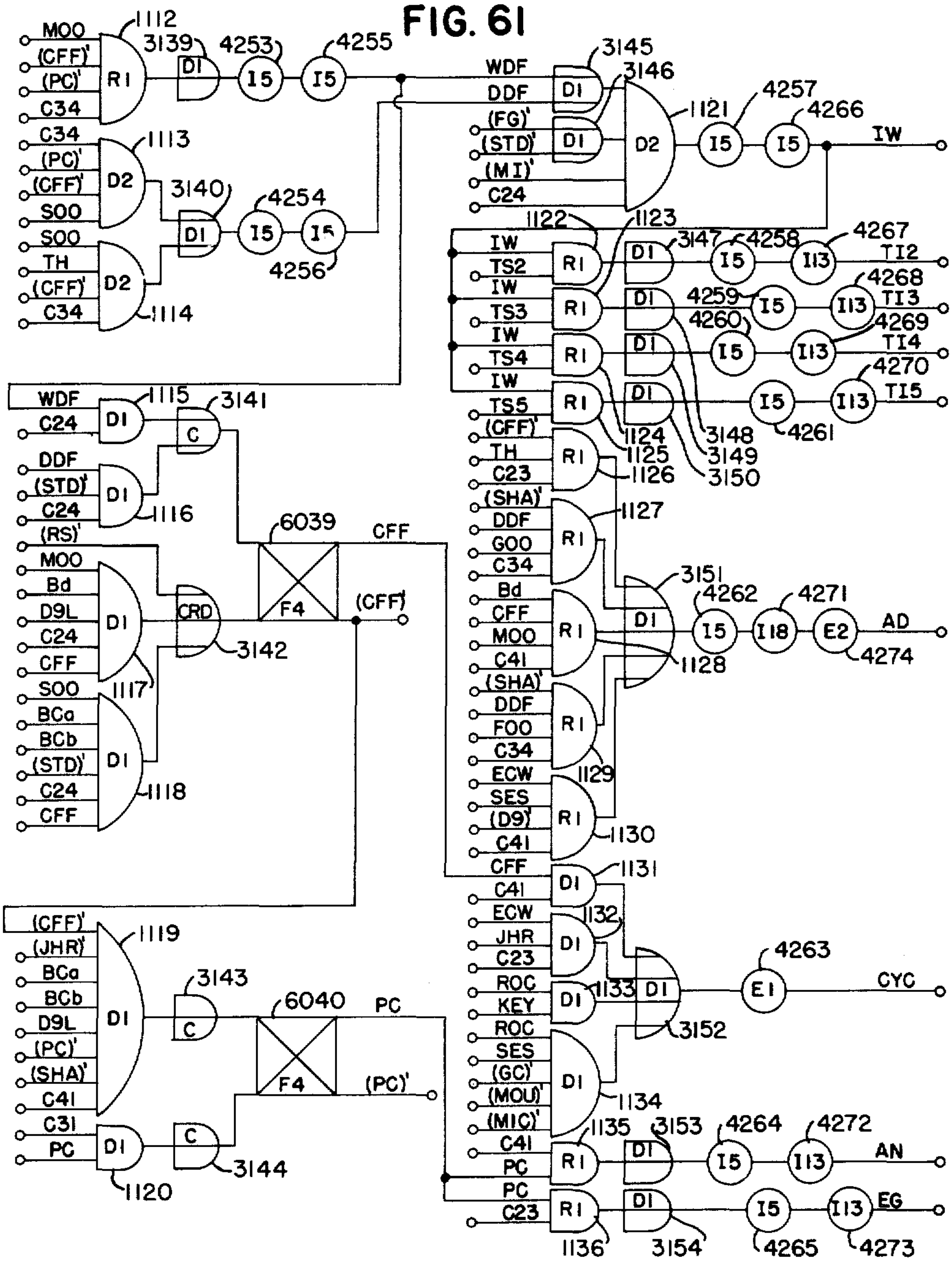


INVENTORS- P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN
C.S. JENKINS, L.D. KILLHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR,
R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL
E.M. GARNER, R.O. SATHER, R.L. YOST, & A.B. BRADEN, JR.
BY Louis A. Kline
Havender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 39

FIG. 61



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E.V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K.A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Lavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 40

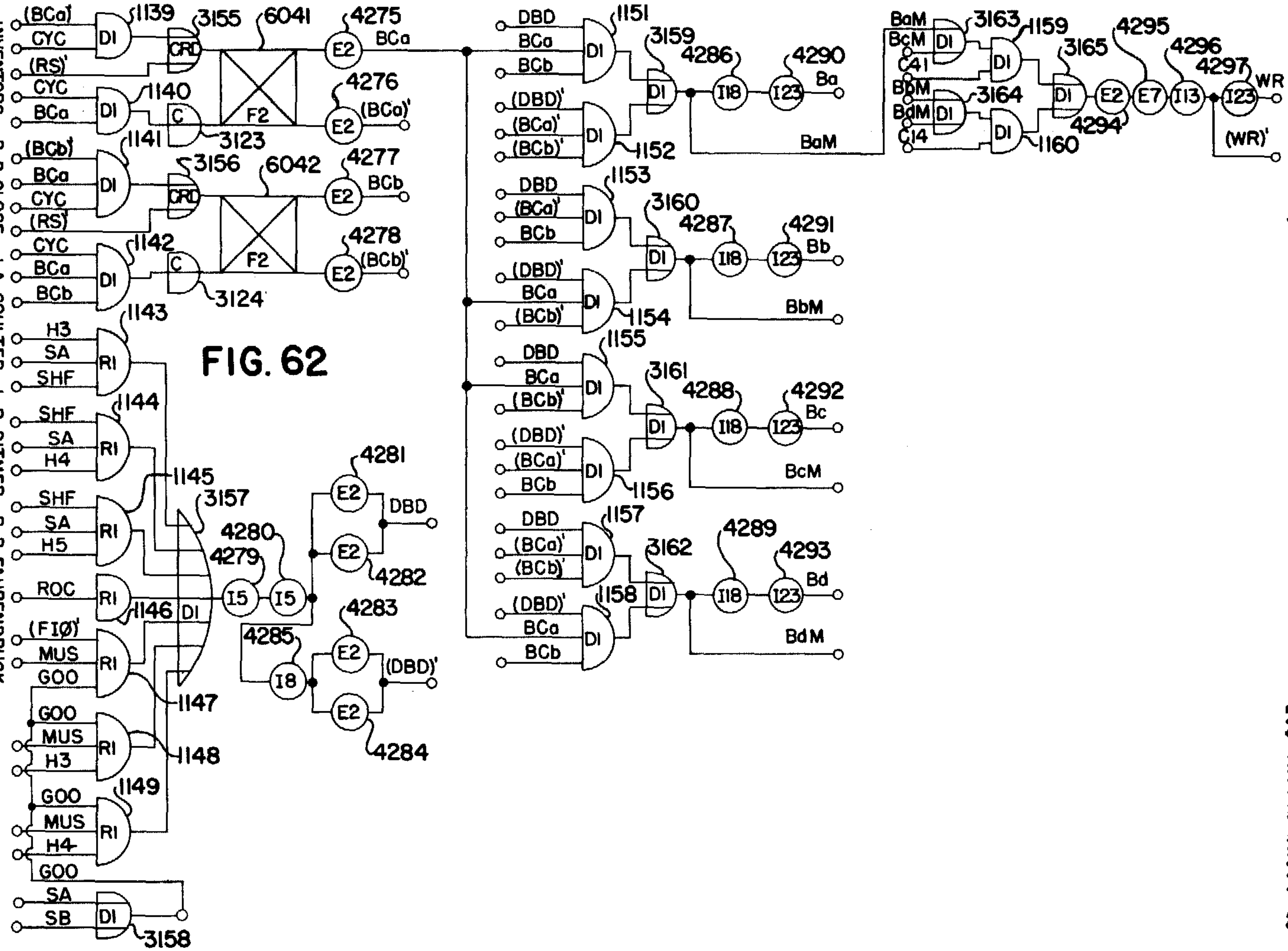


FIG. 62

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

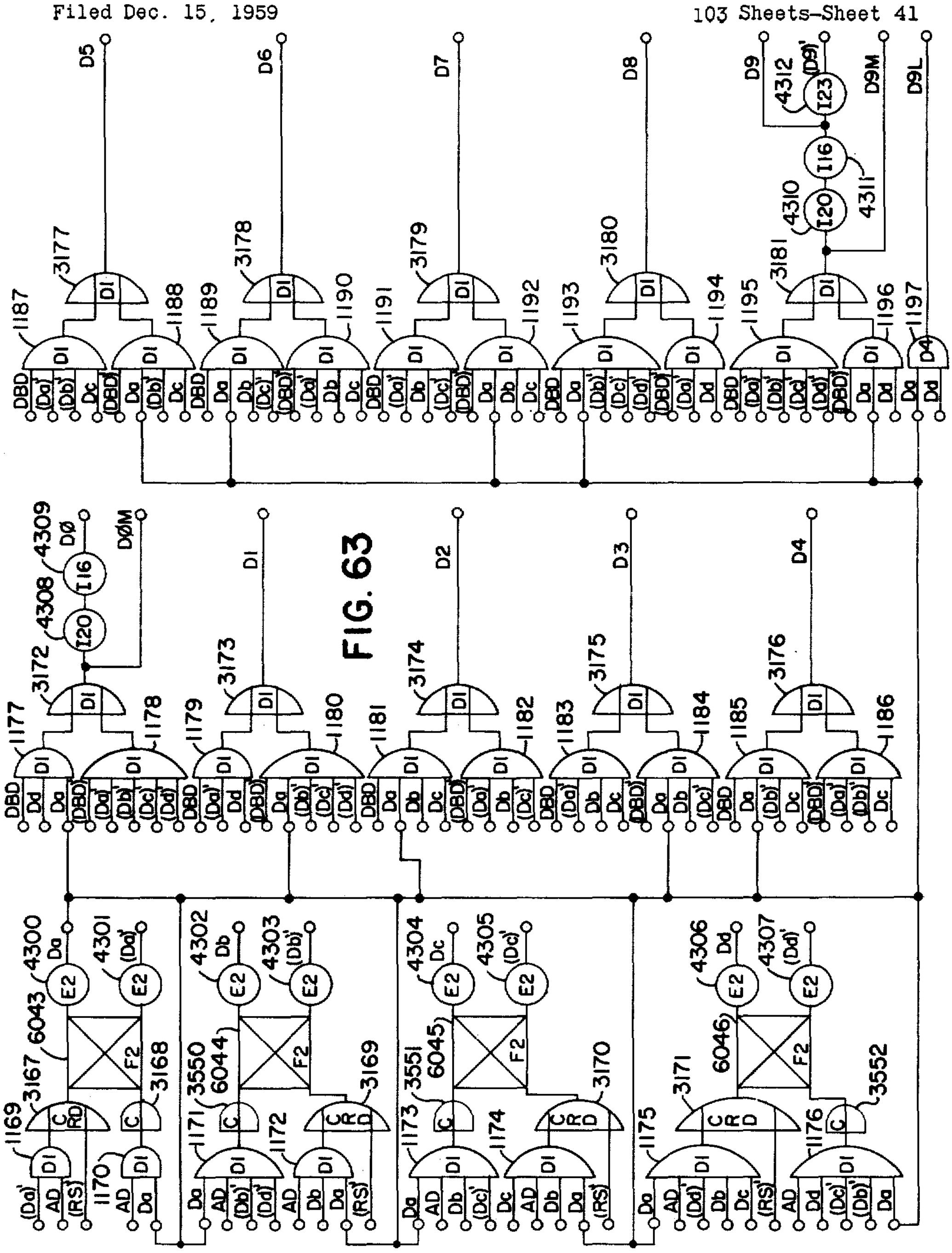


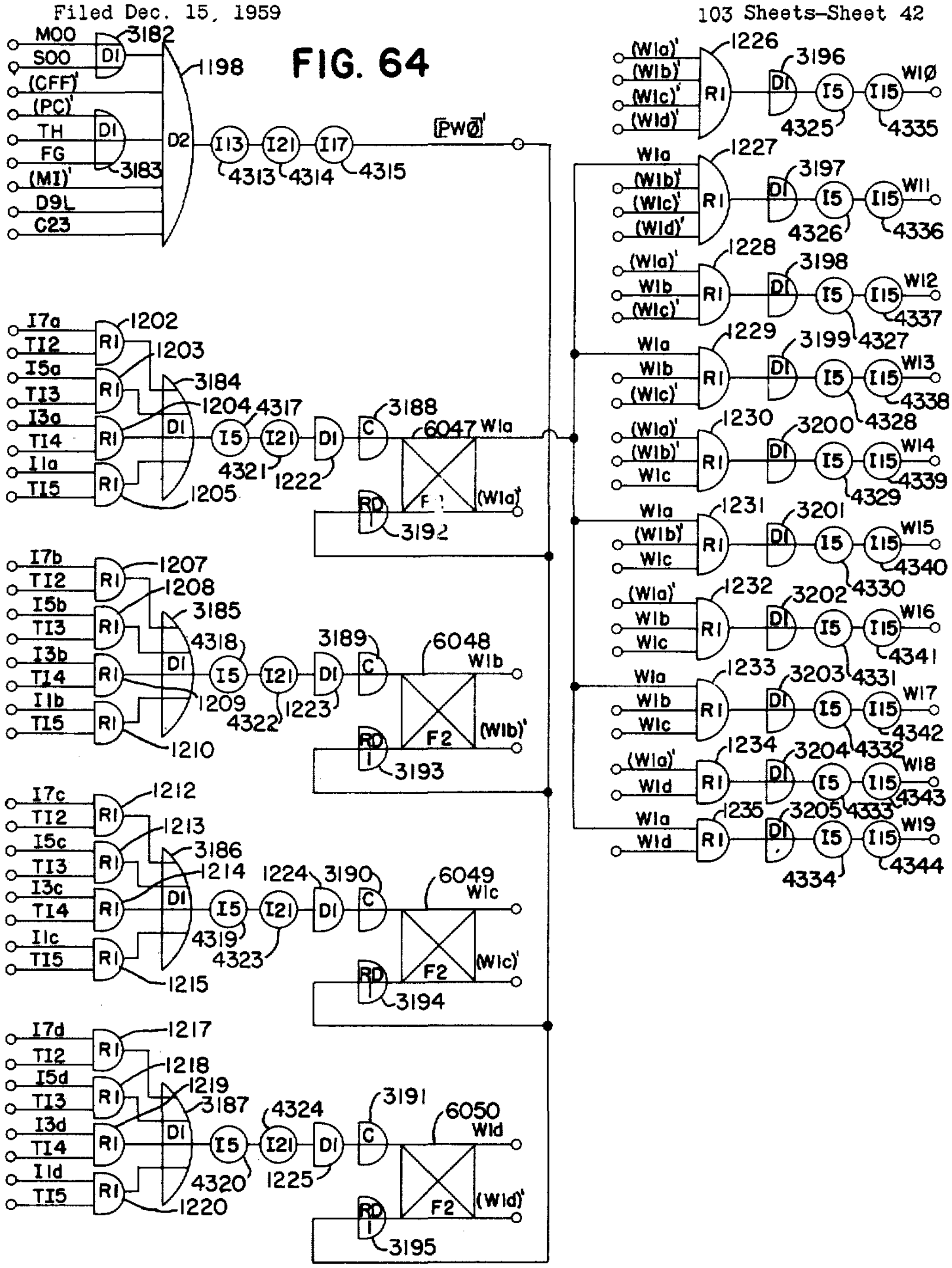
FIG. 63

INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK, F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN, C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR, R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL, E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.

BY Louis A. Kline
Lavender

THEIR ATTORNEYS

FIG. 64



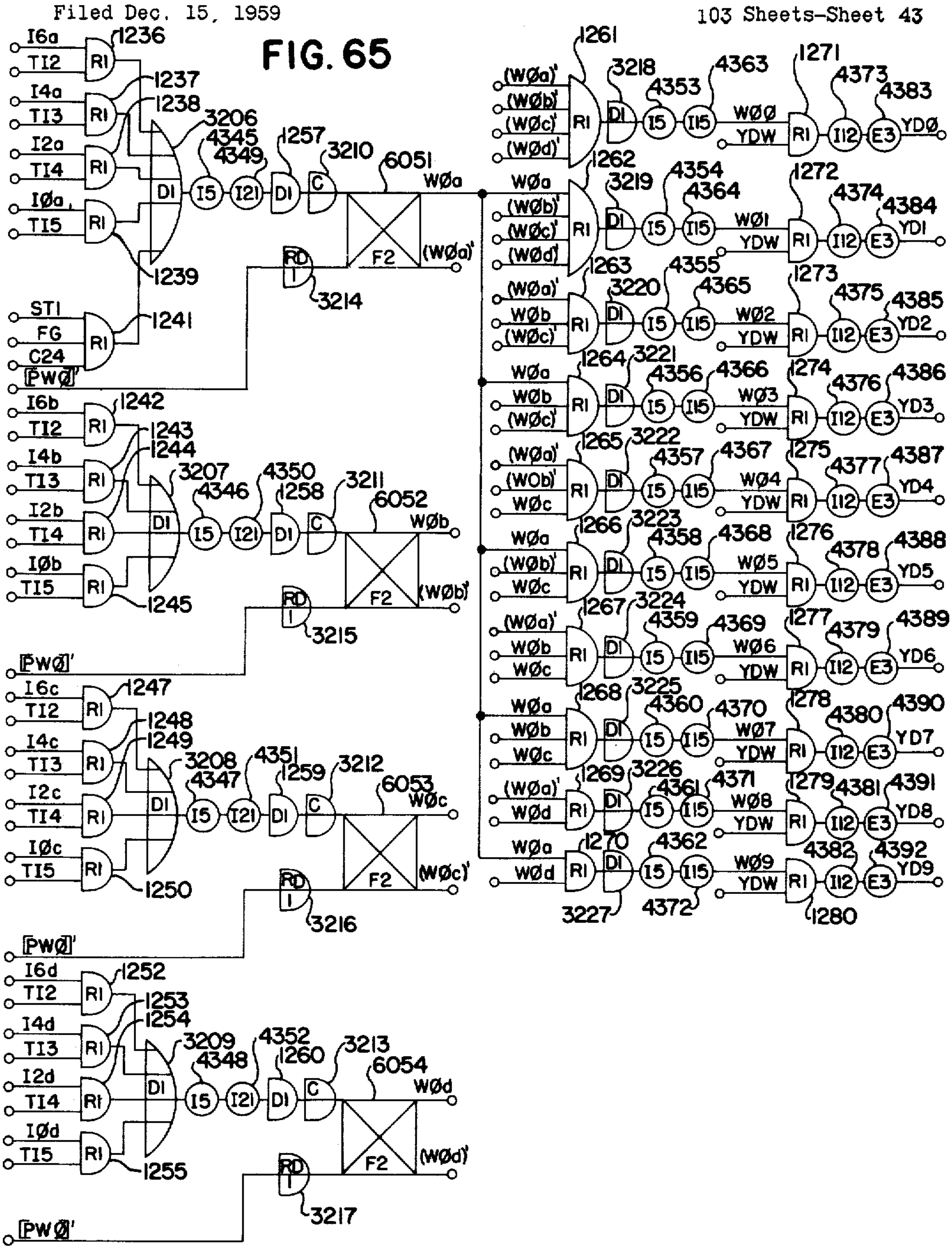
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, G. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Havender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 43

FIG. 65



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J. Cavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 44

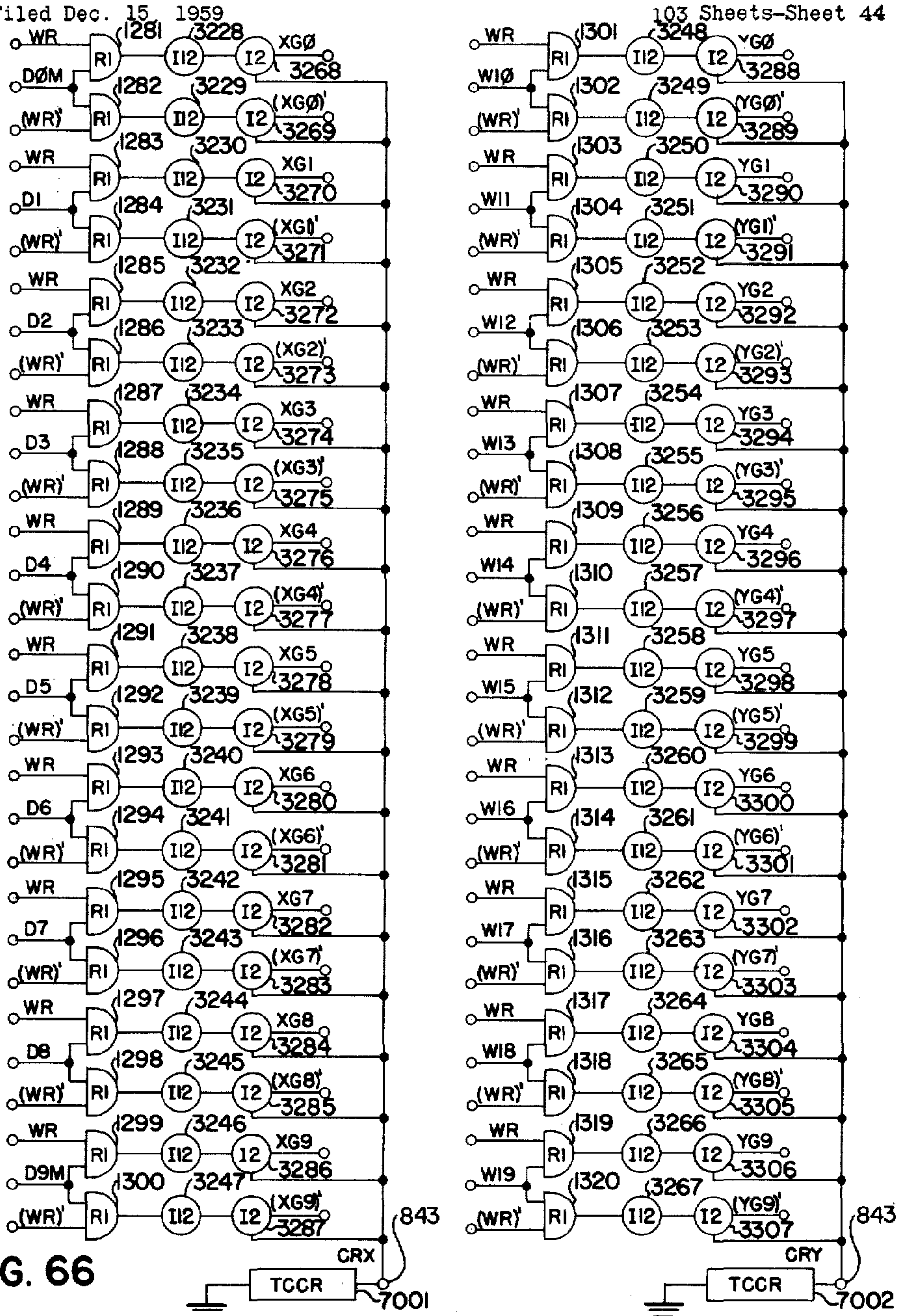


FIG. 66

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C.S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J. Cavender
THEIR ATTORNEYS

FIG. 67A

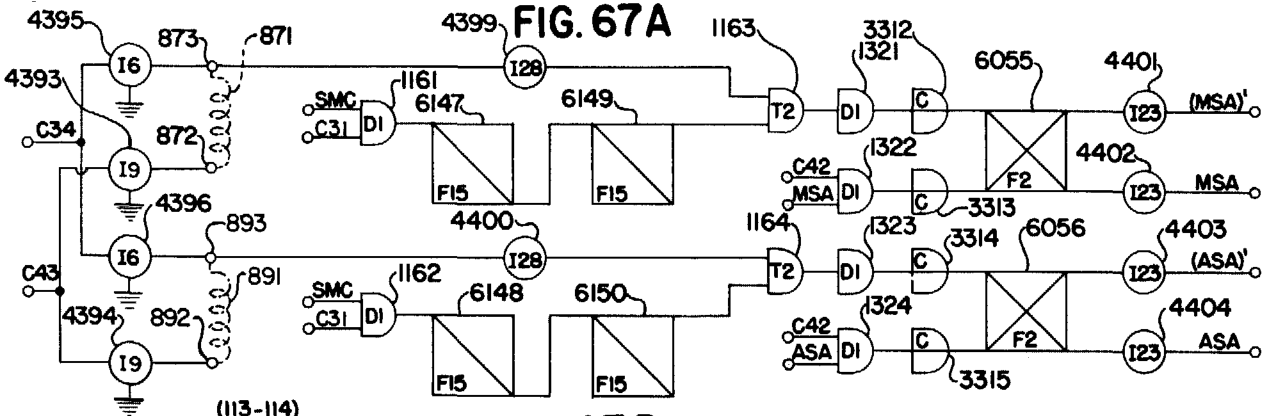
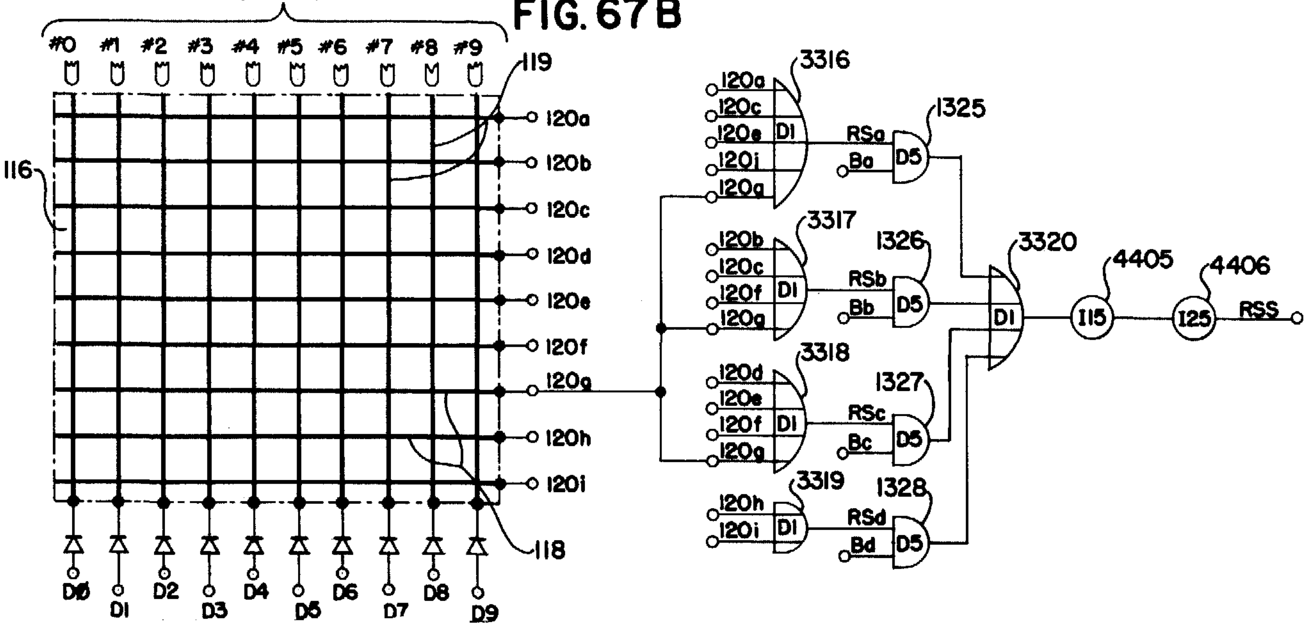


FIG. 67B



INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK, F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN, C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR, R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL, E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.

BY

THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 46

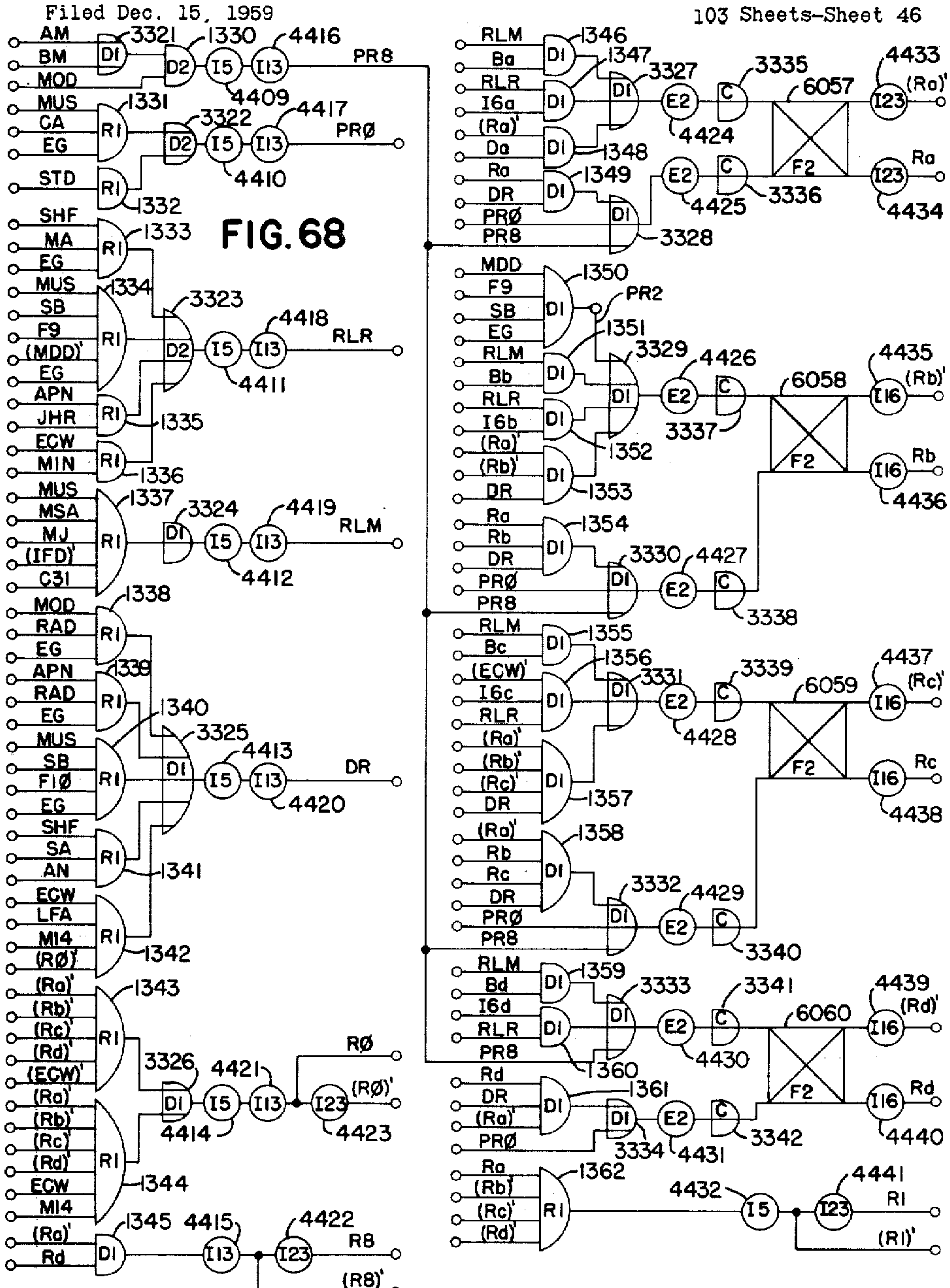


FIG. 68

INVENTORS — P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline

J Cavender

THEIR ATTORNEYS

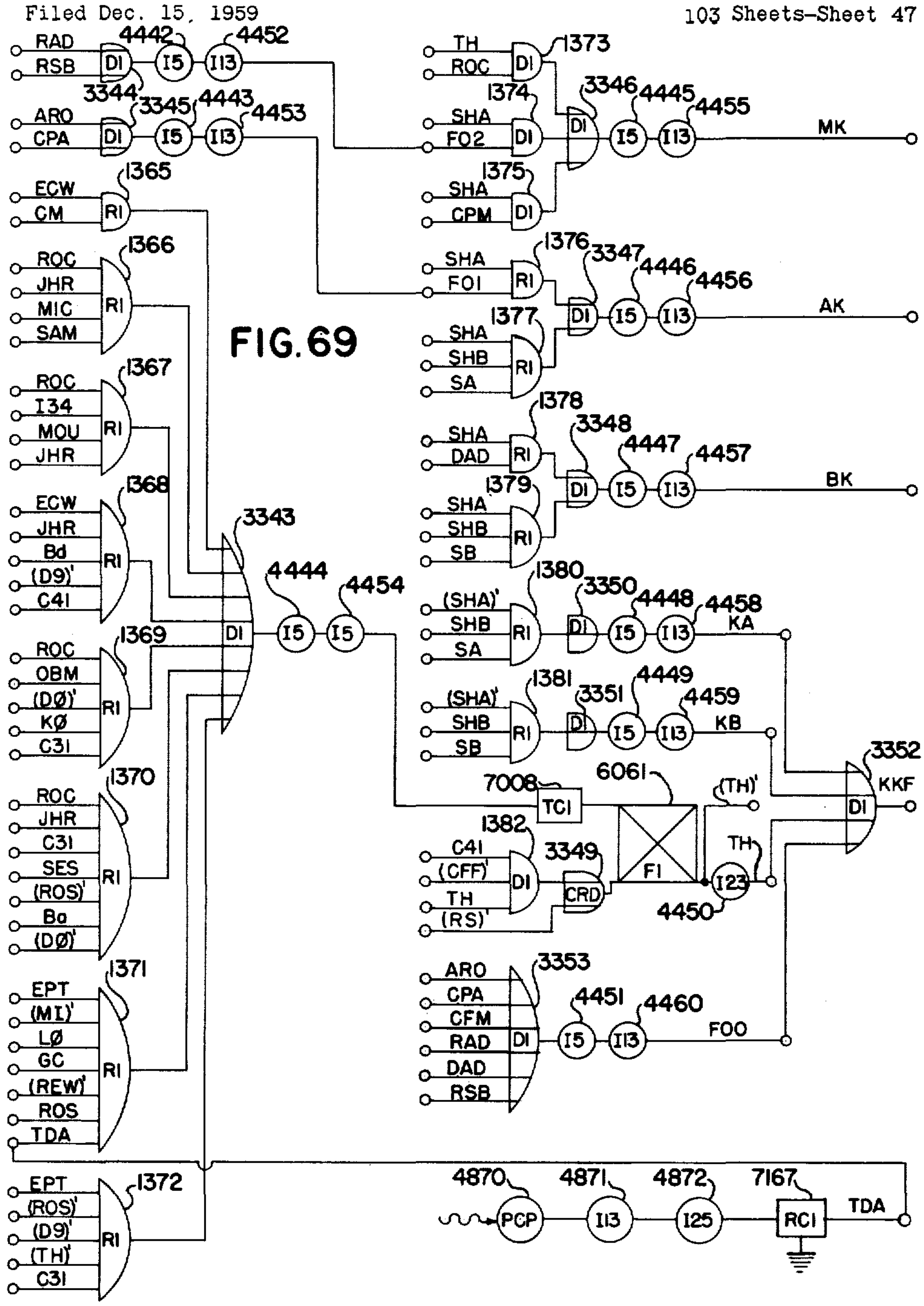
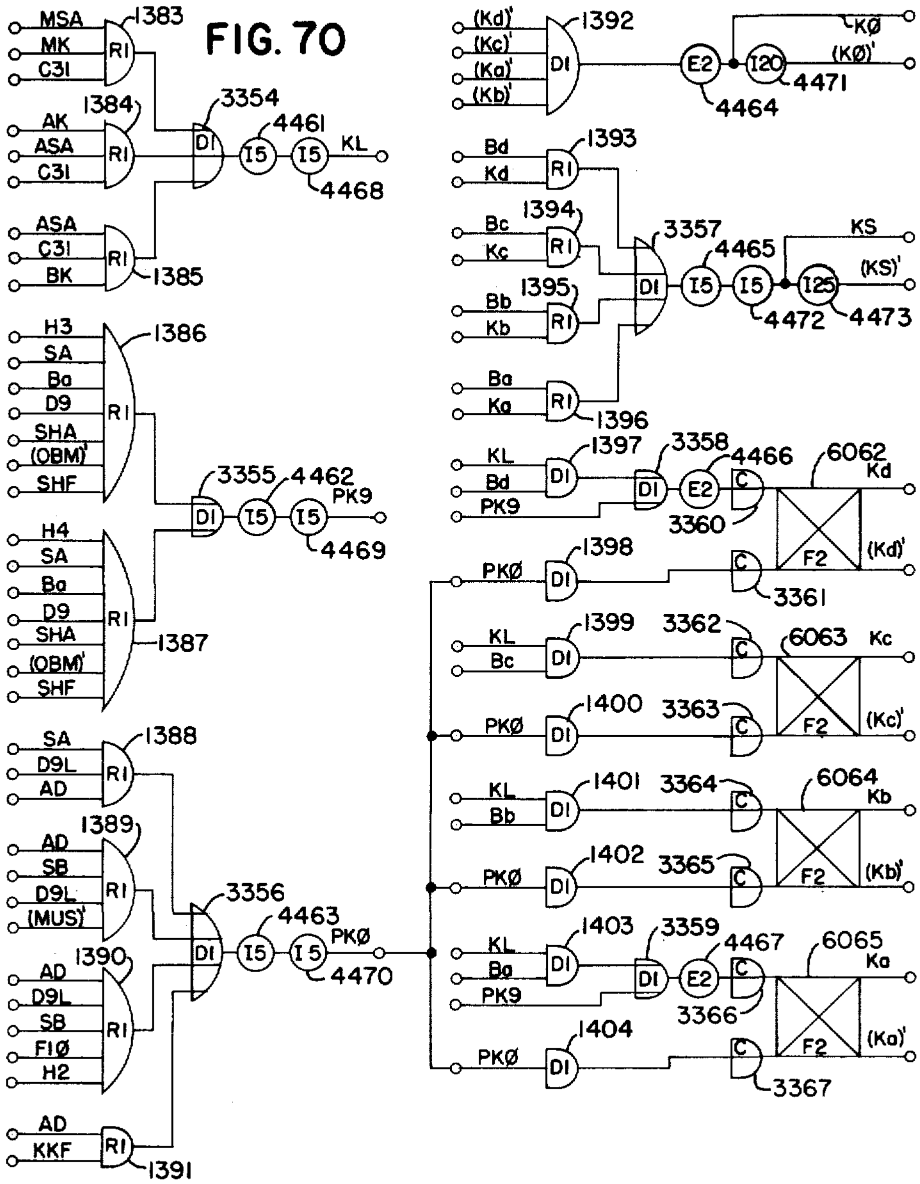


FIG. 69

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

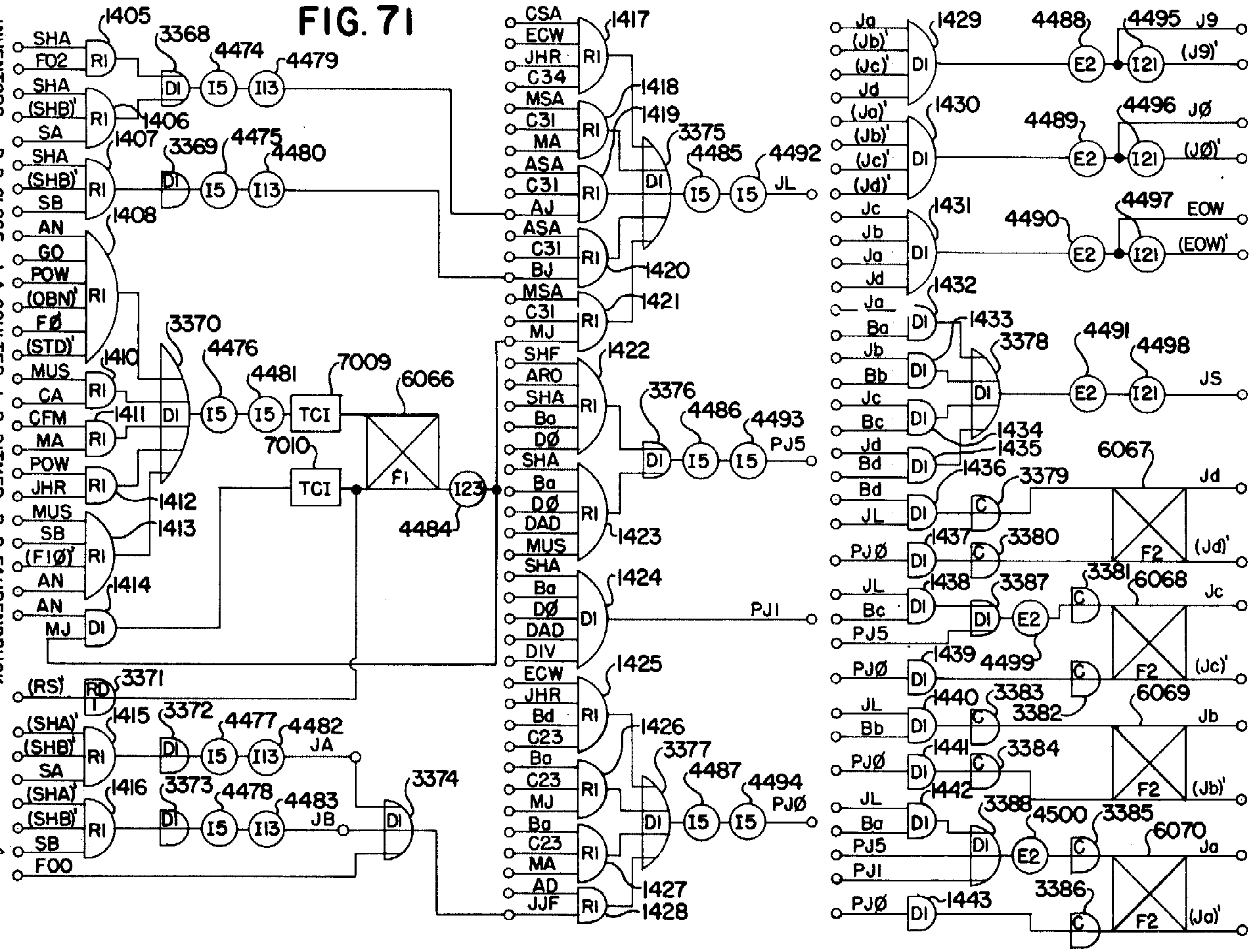
FIG. 70



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, G.S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Hauender
THEIR ATTORNEYS

FIG. 71

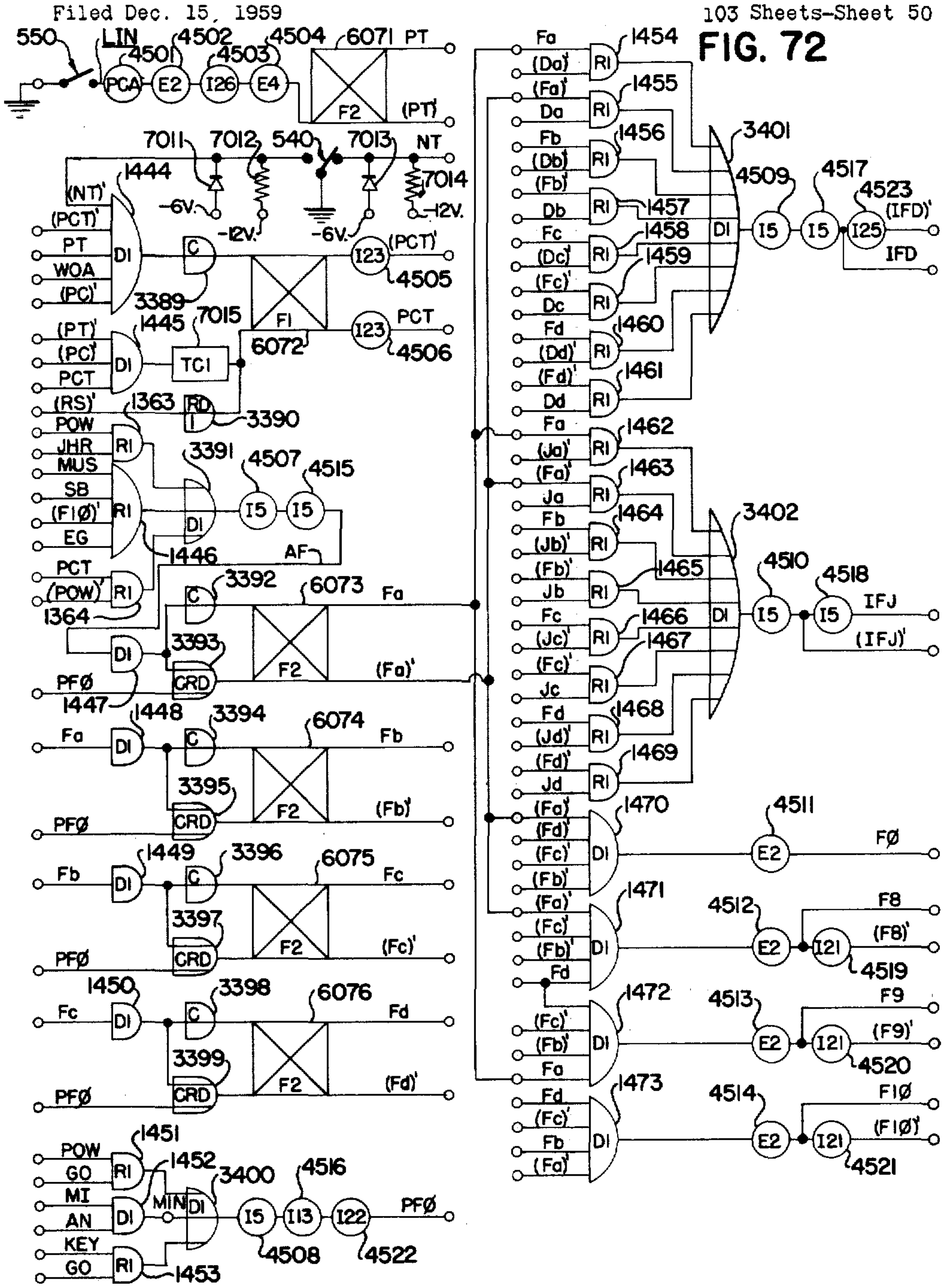


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLHEFFER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 50

FIG. 72



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J. Cavender
THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 51

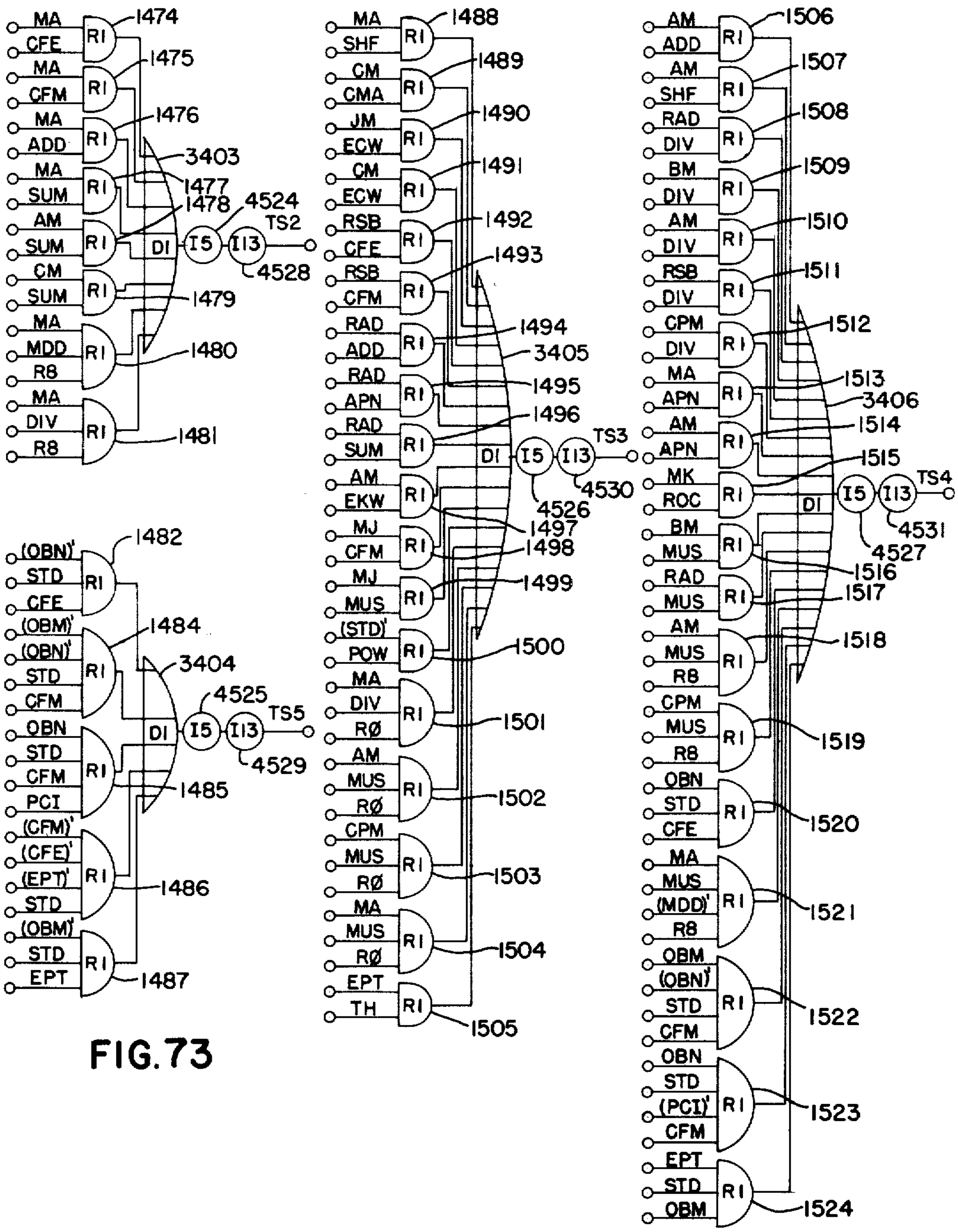


FIG.73

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
[signature]
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

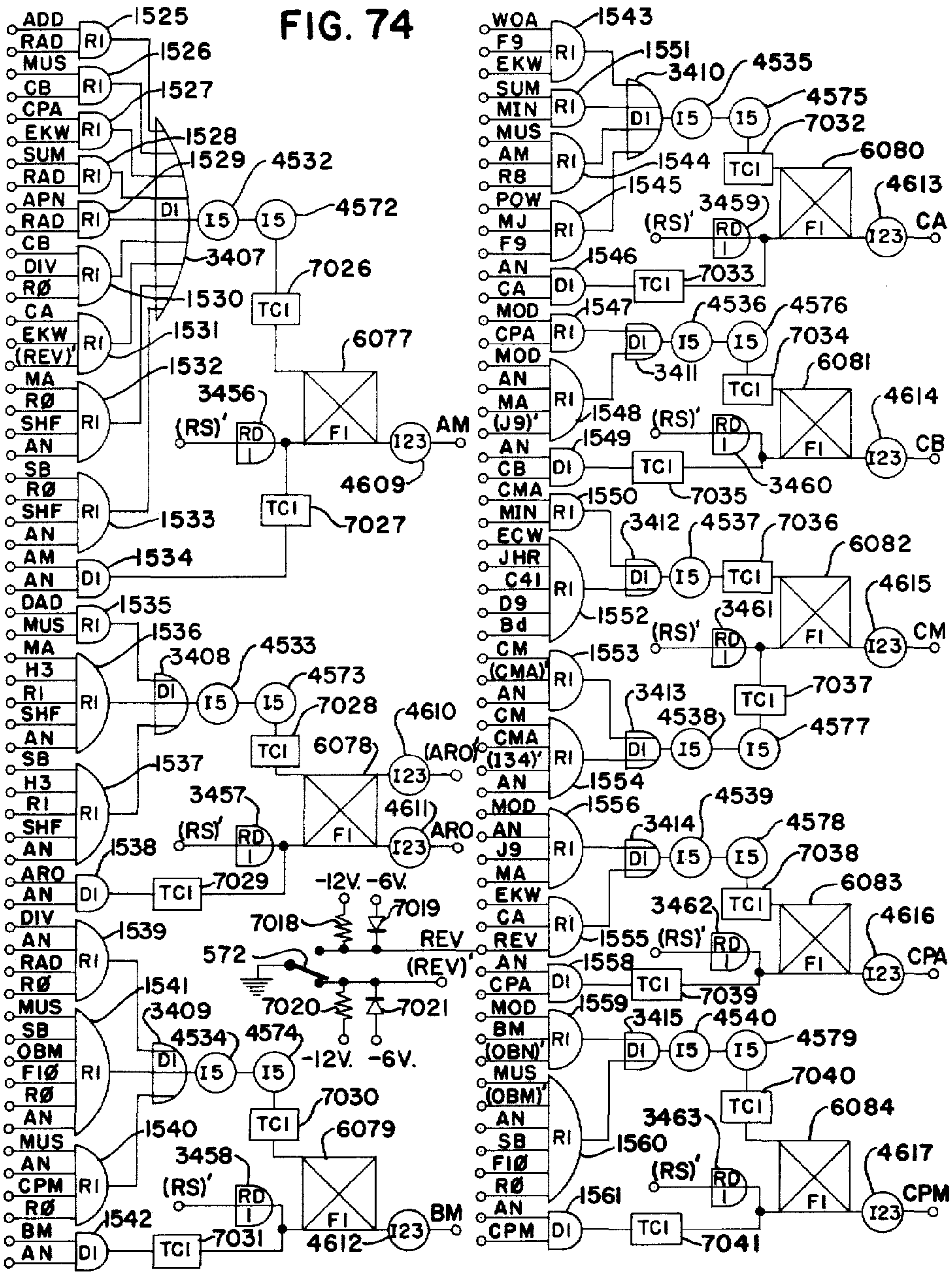


FIG. 74

INVENTORS- P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDAL, E. N. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

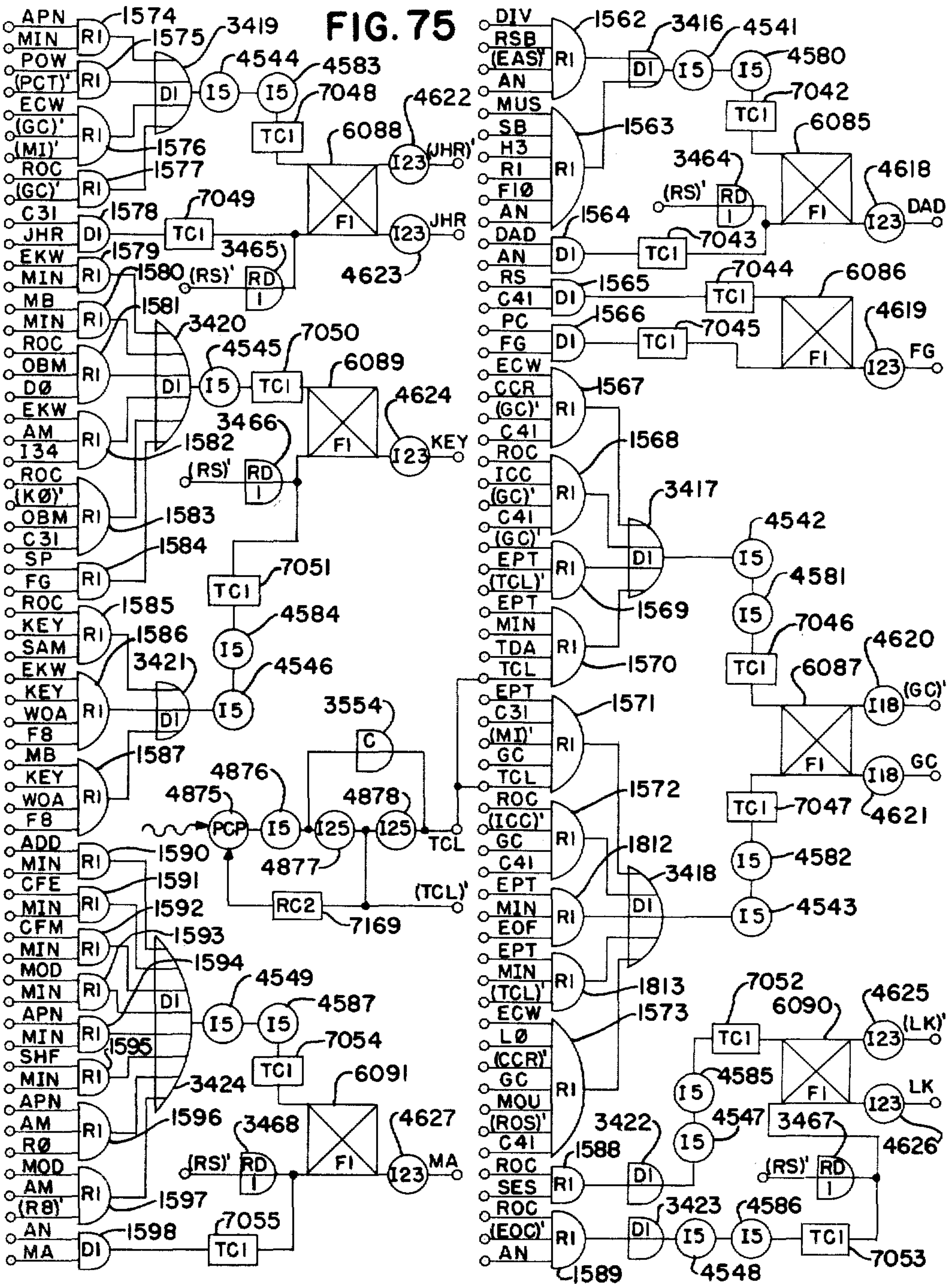
BY Louis A Kline

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 53

FIG. 75



INVENTORS— P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E.V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Havender
THEIR ATTORNEYS

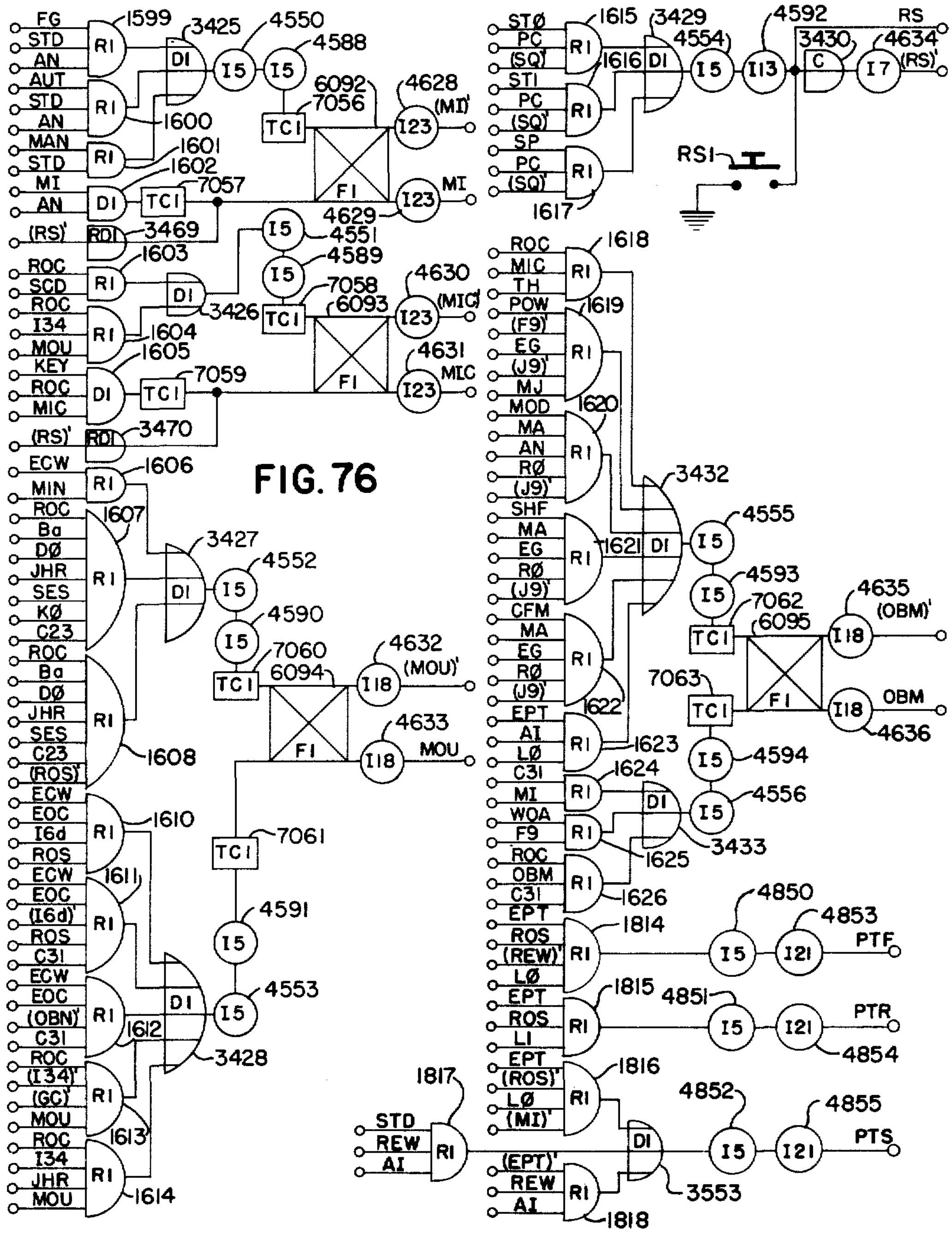
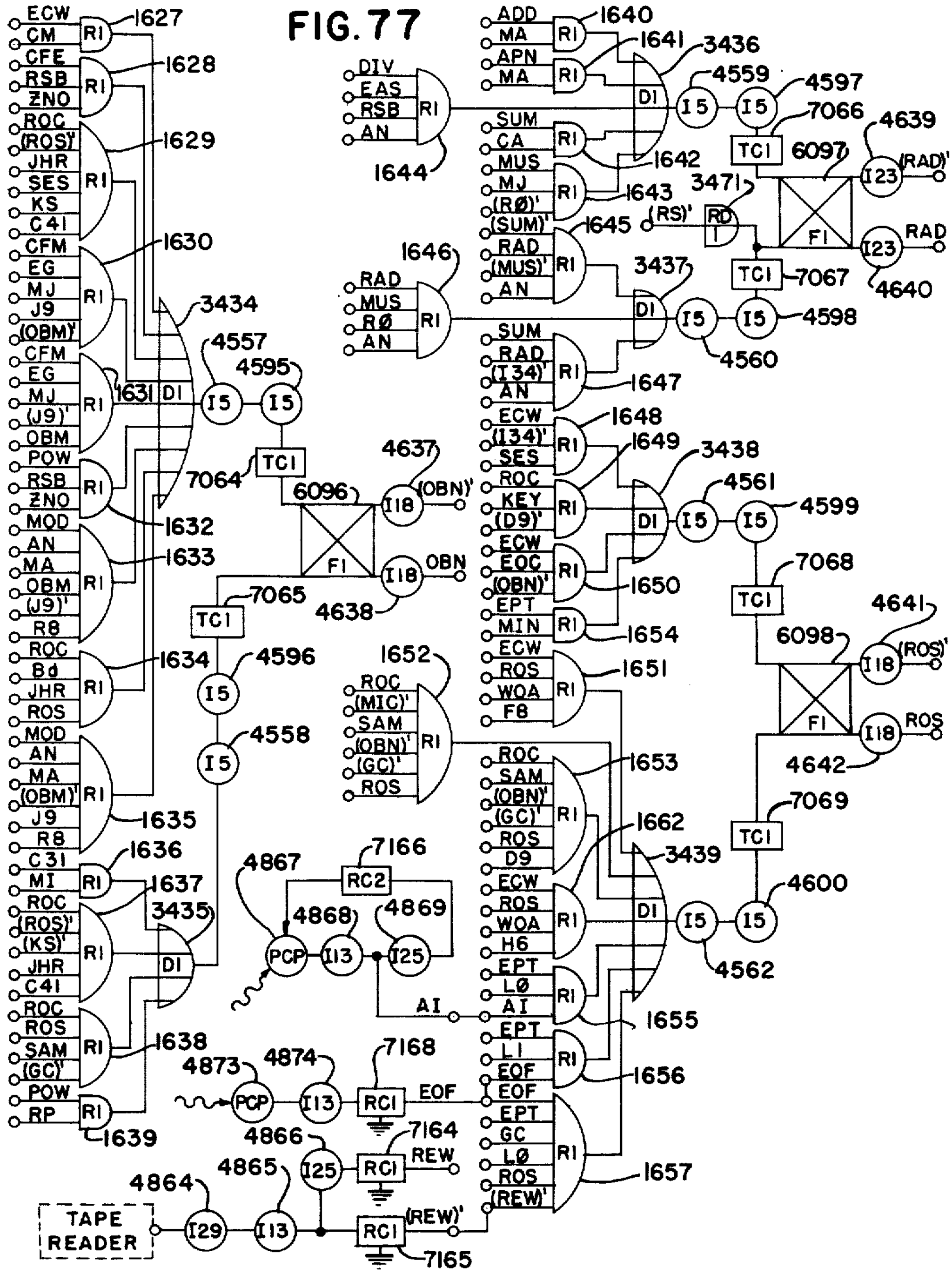


FIG. 76

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Havender
THEIR ATTORNEYS

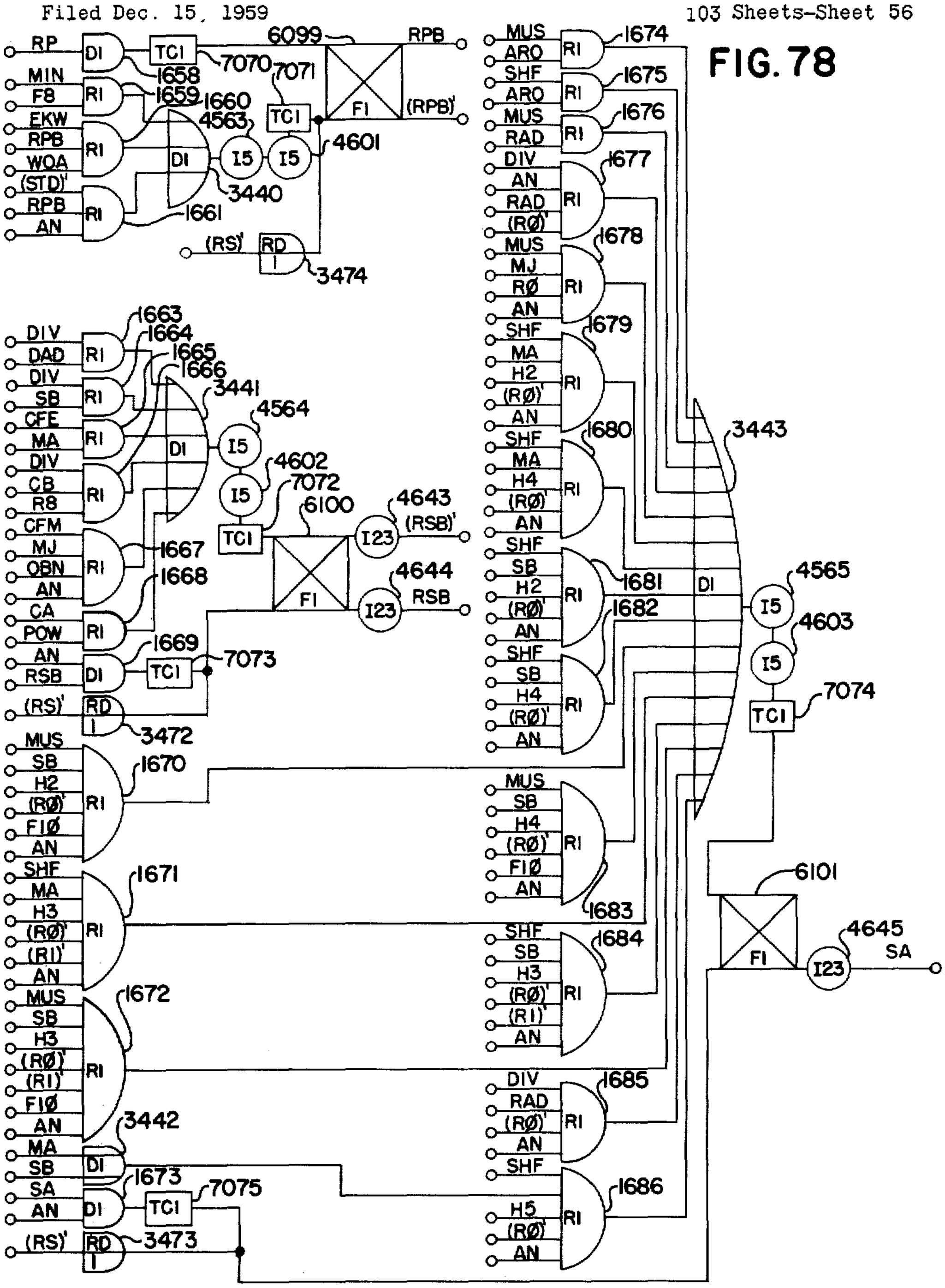
FIG. 77



INVENTORS— P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY [signature]
THEIR ATTORNEYS

FIG. 78

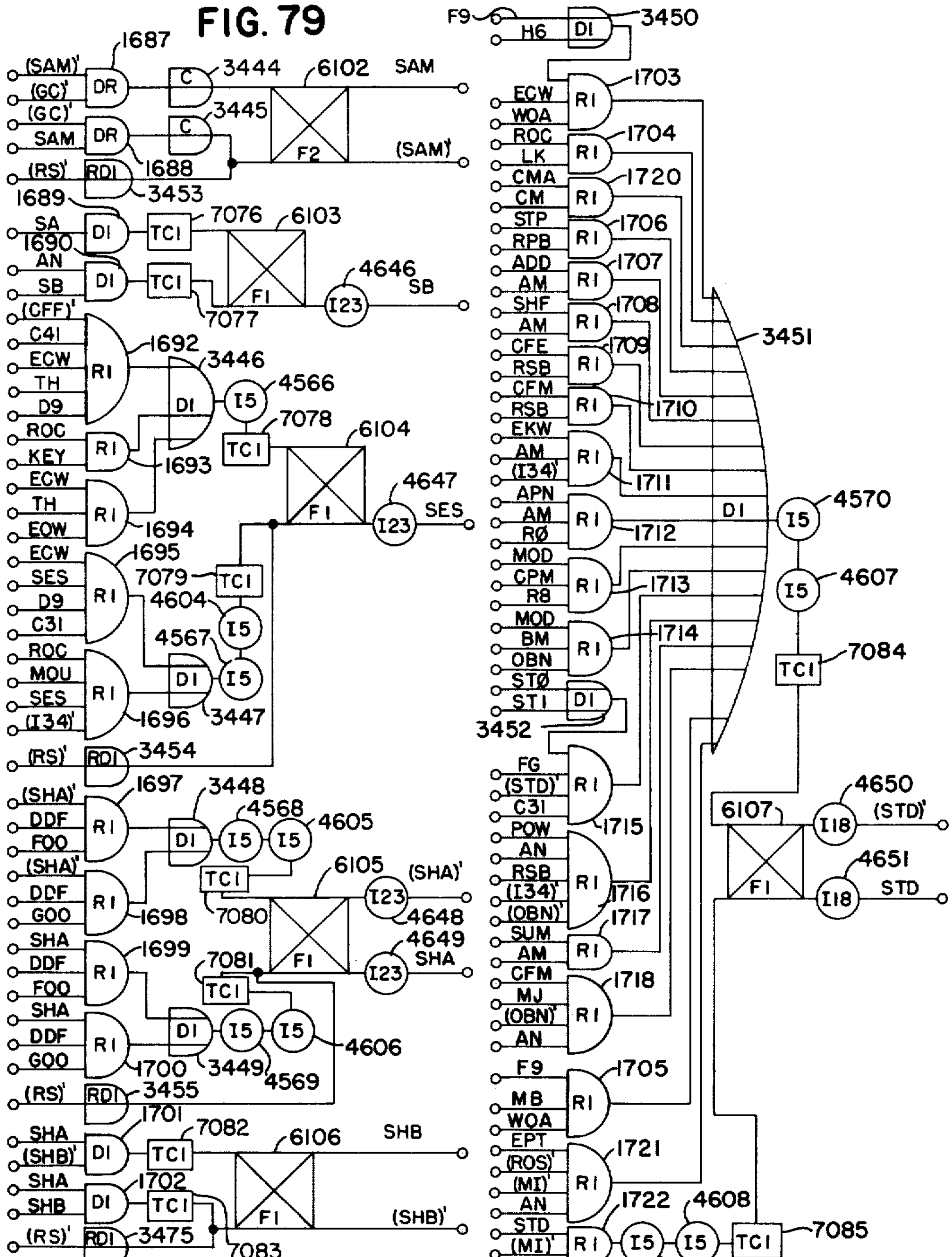


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 79



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

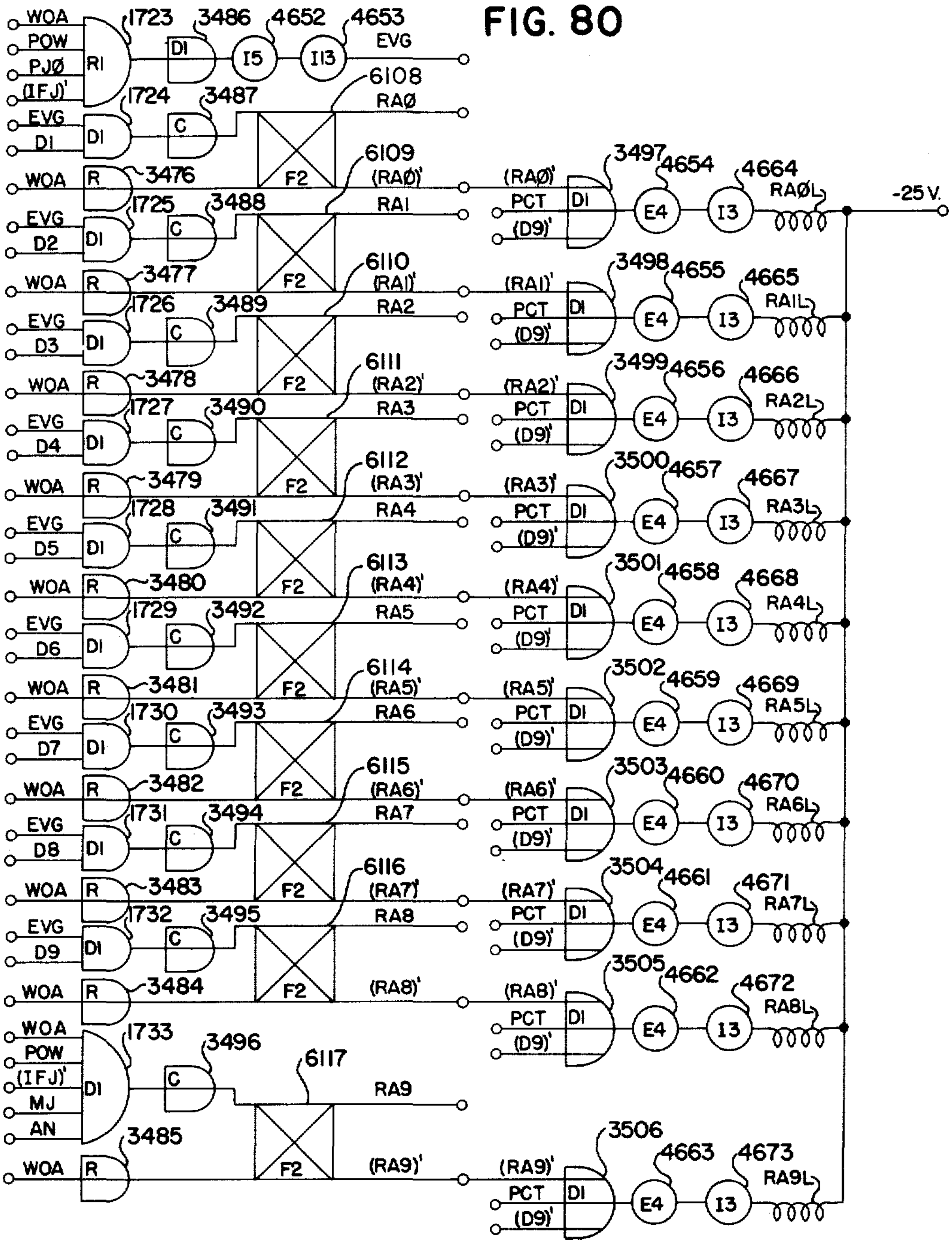
BY Louis A. Kline
J. Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 58

FIG. 80

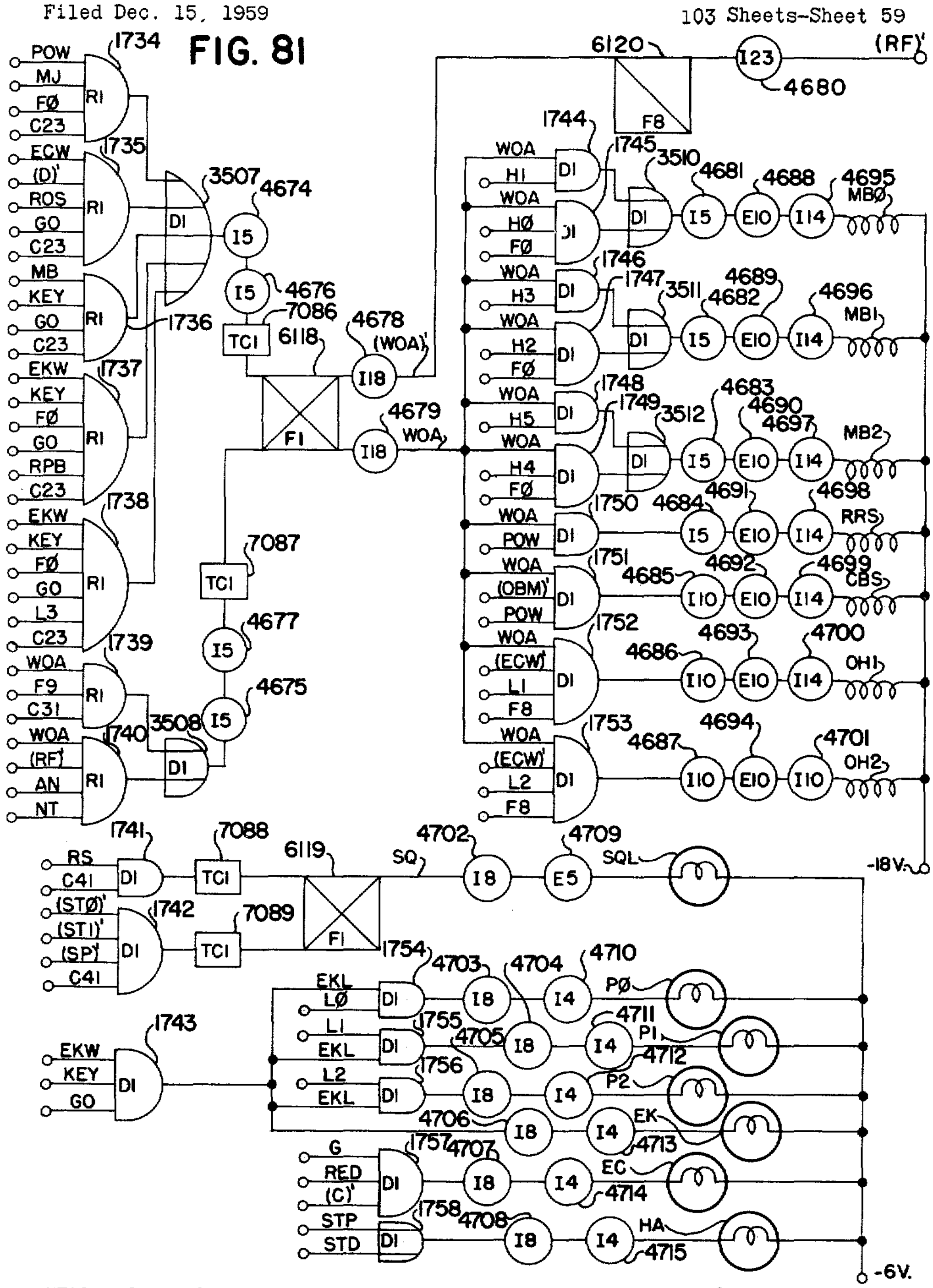


INVENTORS — P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 81

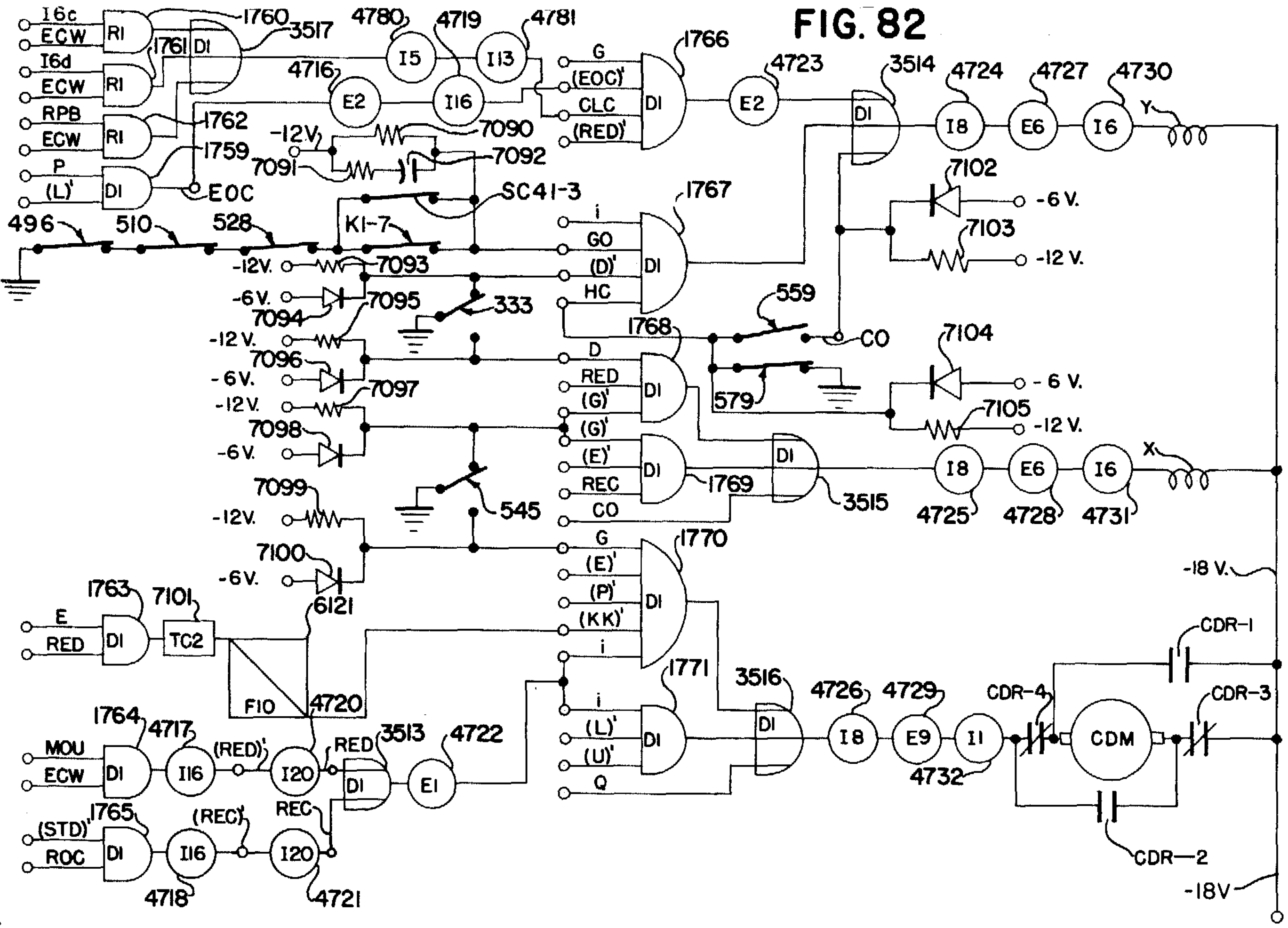


INVENTORS — P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST, & A. B. BRADEN, JR.

BY Louis A Kline
Lavender

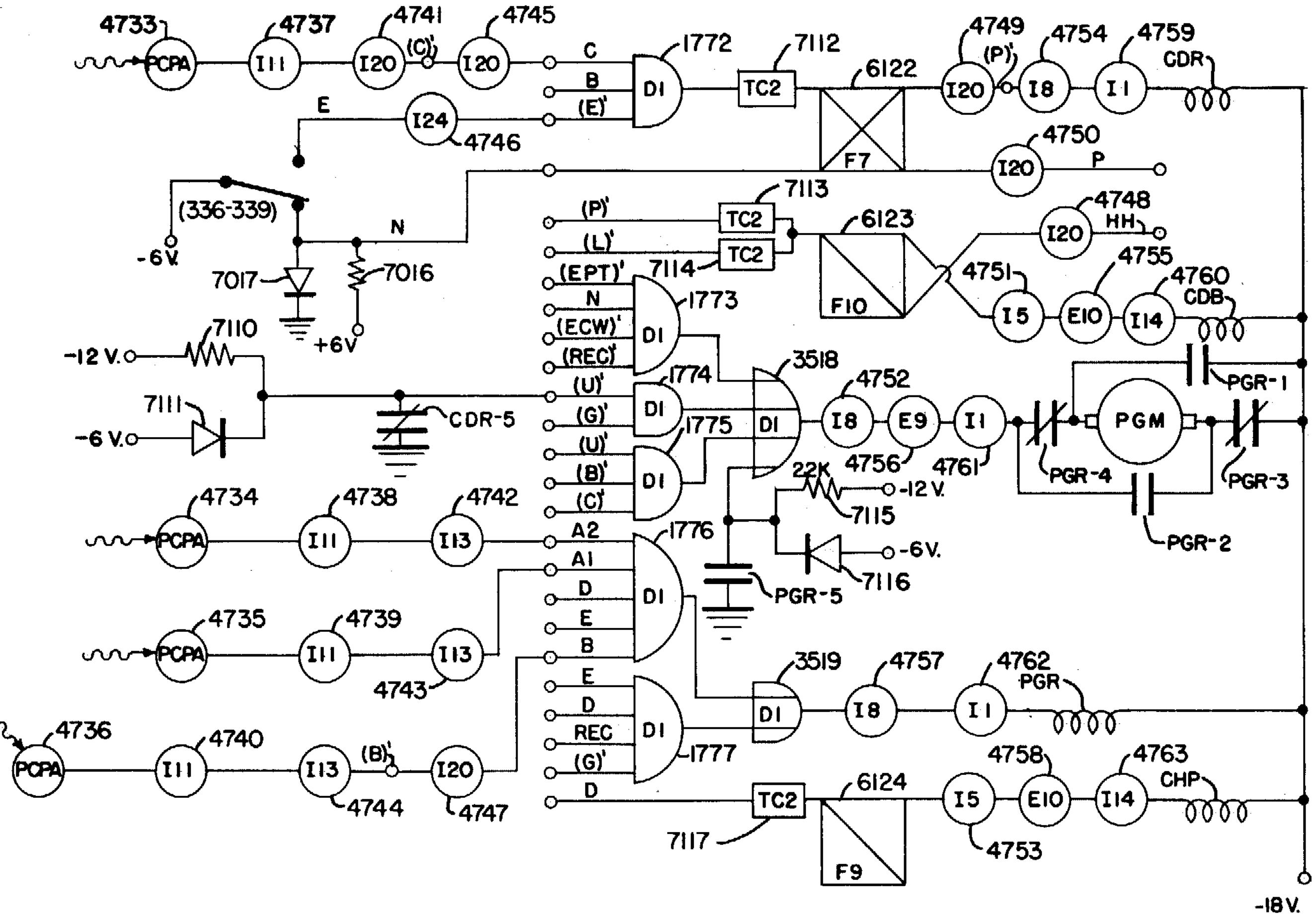
THEIR ATTORNEYS

FIG. 82



INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOLDRICK, E.V. GULDEN, T.P. HOLLORAN,
G.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR,
R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL,
E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.
BY
THEIR ATTORNEYS

FIG. 83



INVENTORS—B. P. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 62

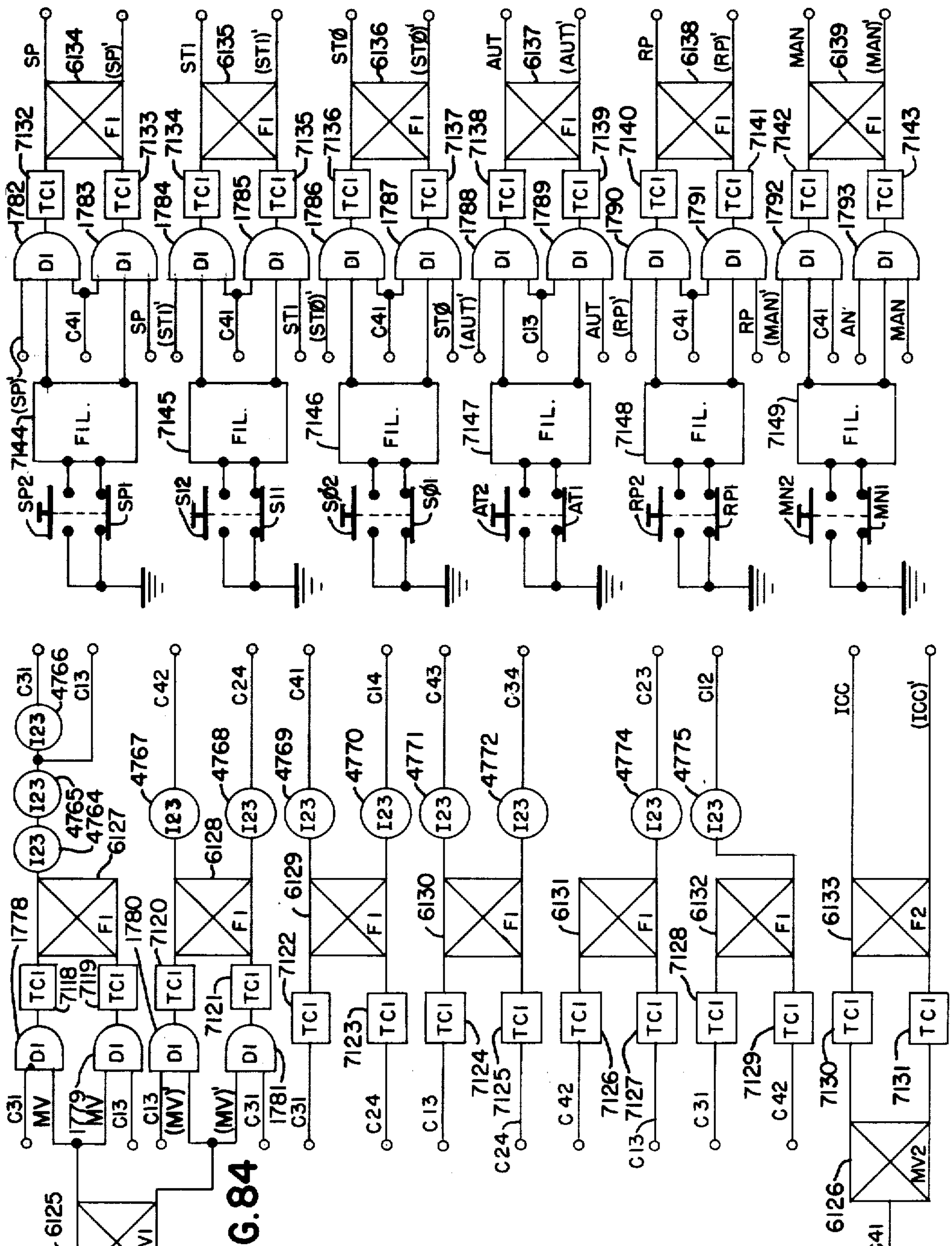


FIG. 84

INVENTORS—P. B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK, F. R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W.L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

Filed Dec. 15, 1959

ELECTRONIC COMPUTING MACHINE

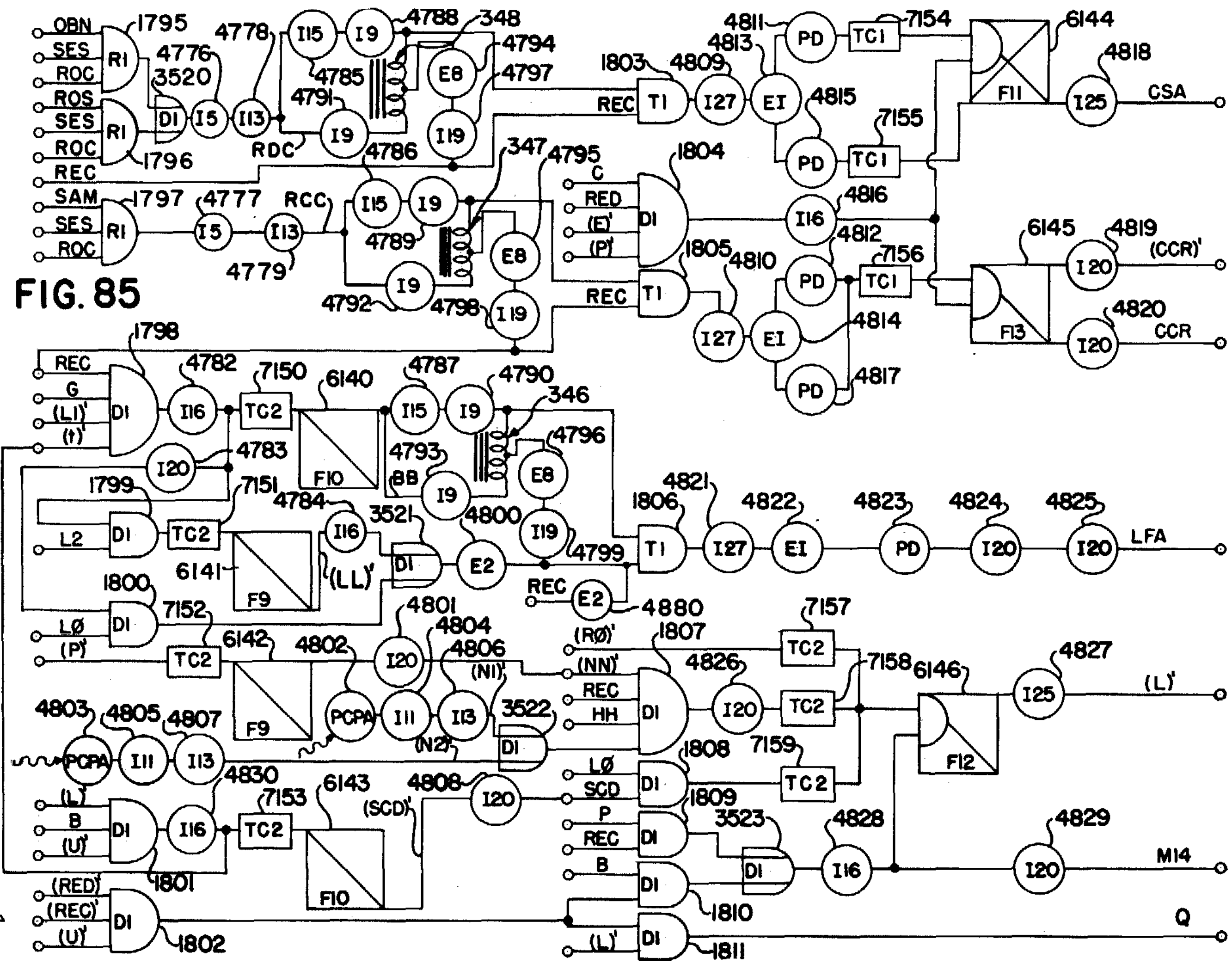


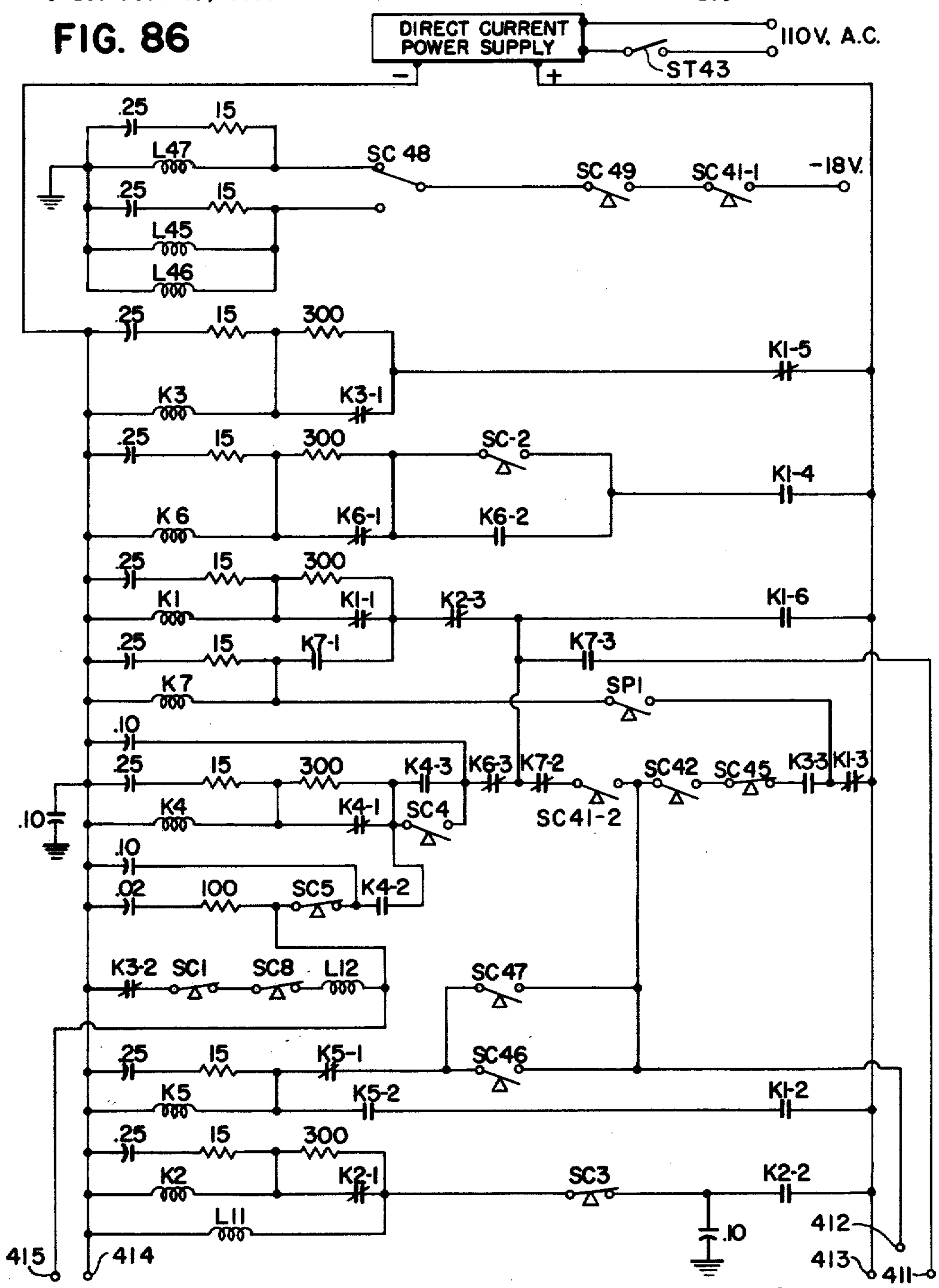
FIG. 85

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 86

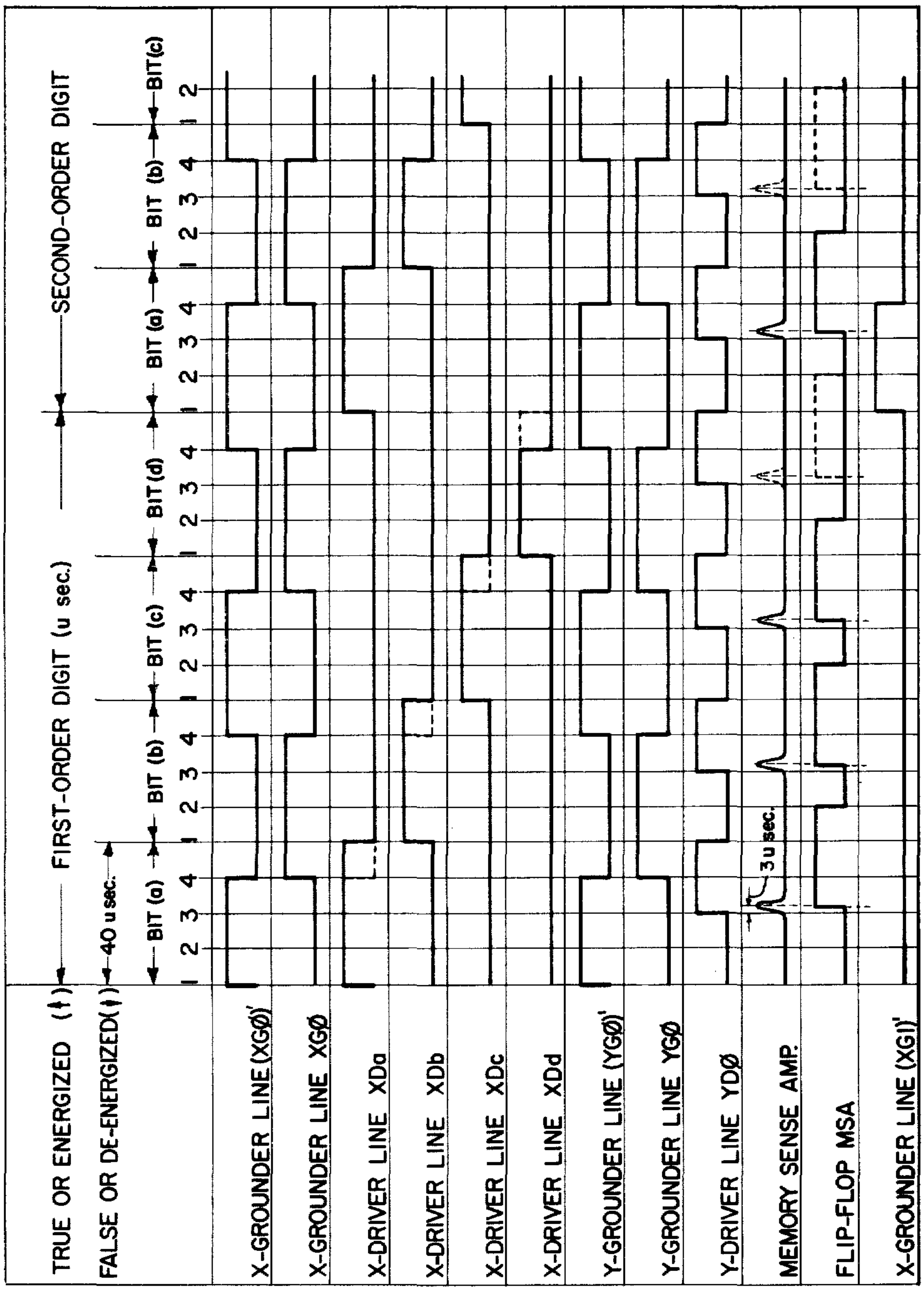


INVENTORS- P.B. CLOSE, J.A.COULTER, L.R.DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN,
C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D.LYONS, N.D.MANOR,
R.P. MARVIN, W.L.MILLER, P.L.O'BRIAN, J.H.RANDAL,
E.N. GARNER, R.O.SATHER, R L. YOST & A.B. BRADEN, JR.
BY
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 65

FIG. 87A



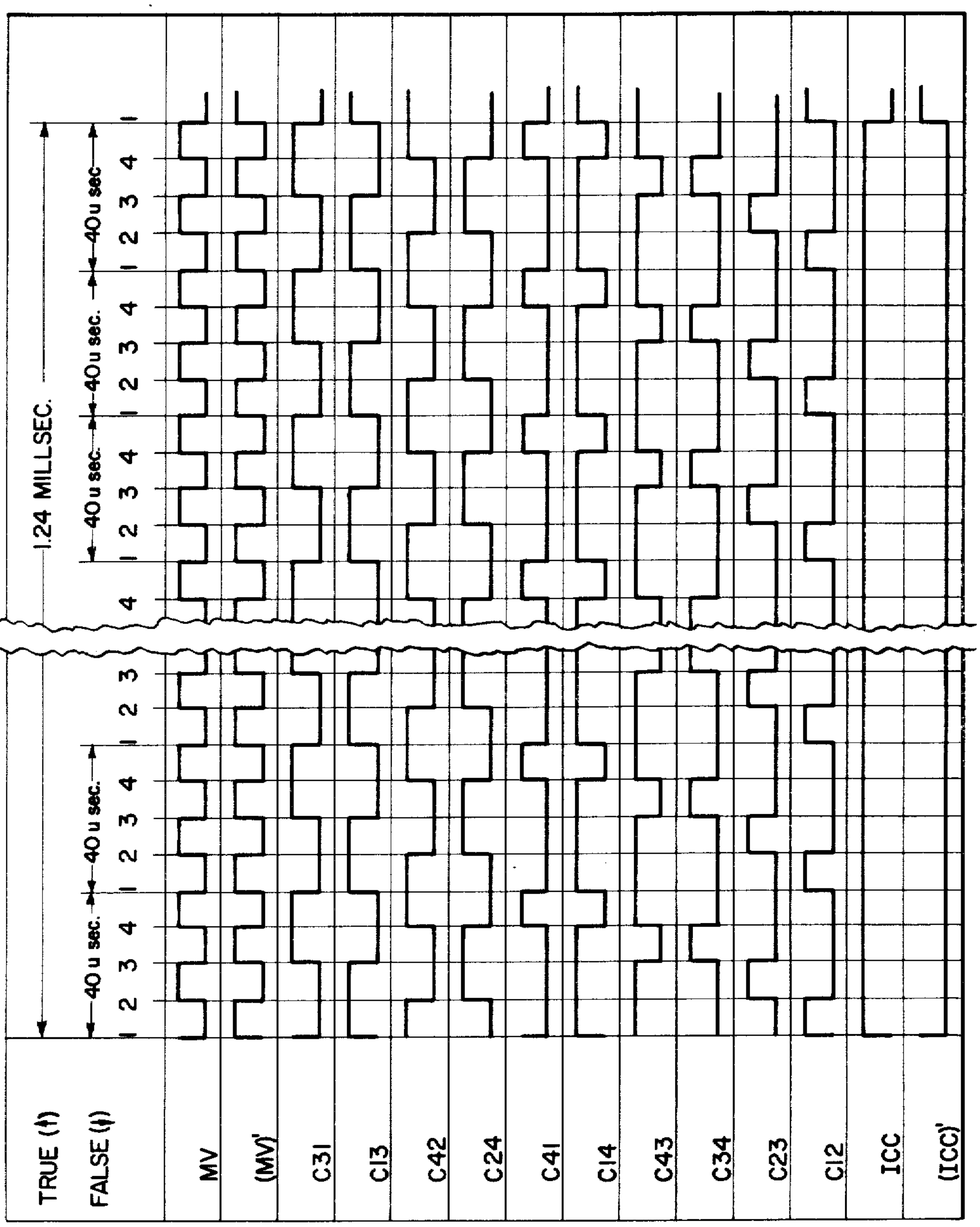
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. F. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Cavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 66

FIG. 87B



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

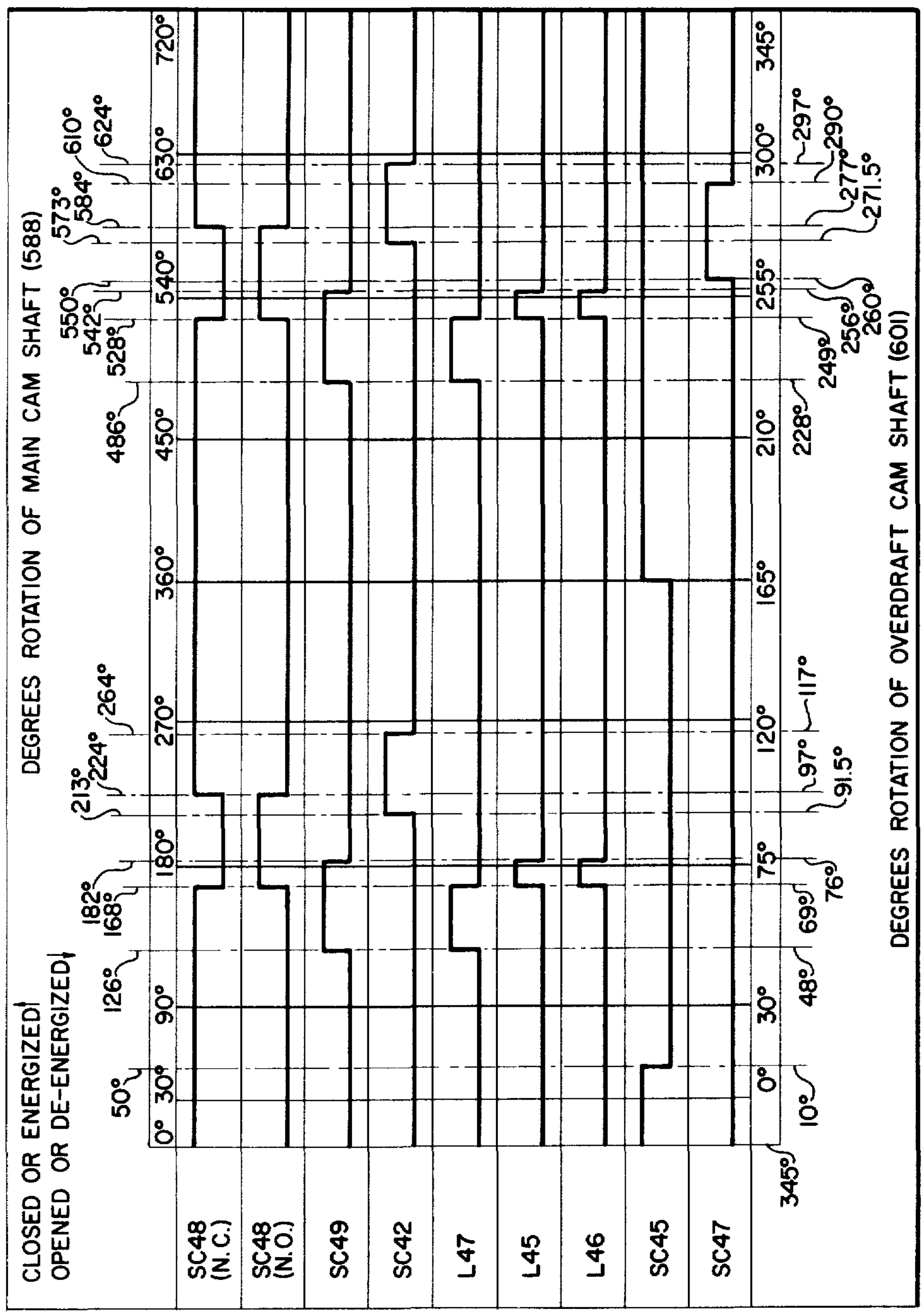
BY Louis A. Kline
J. Cavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 67

FIG. 87C

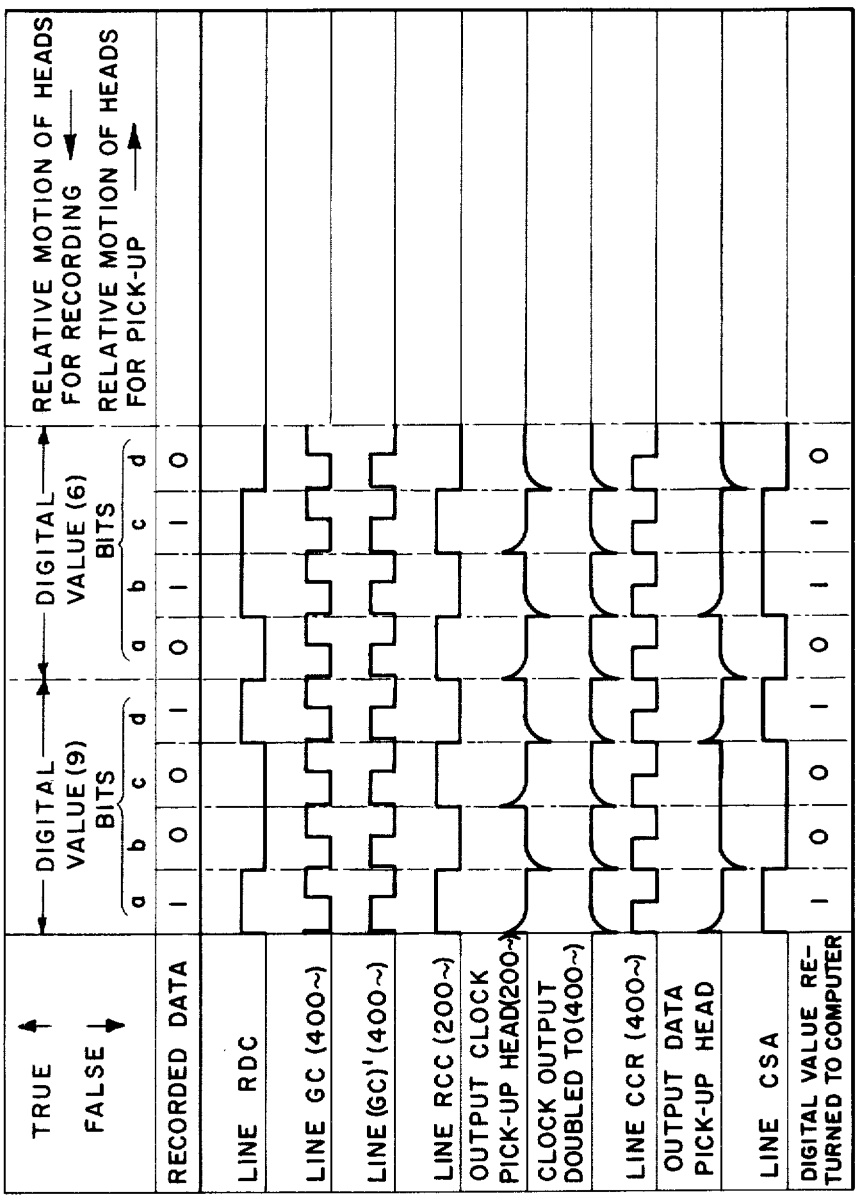


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Cavender

THEIR ATTORNEYS

FIG. 87 D



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
[signature]
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

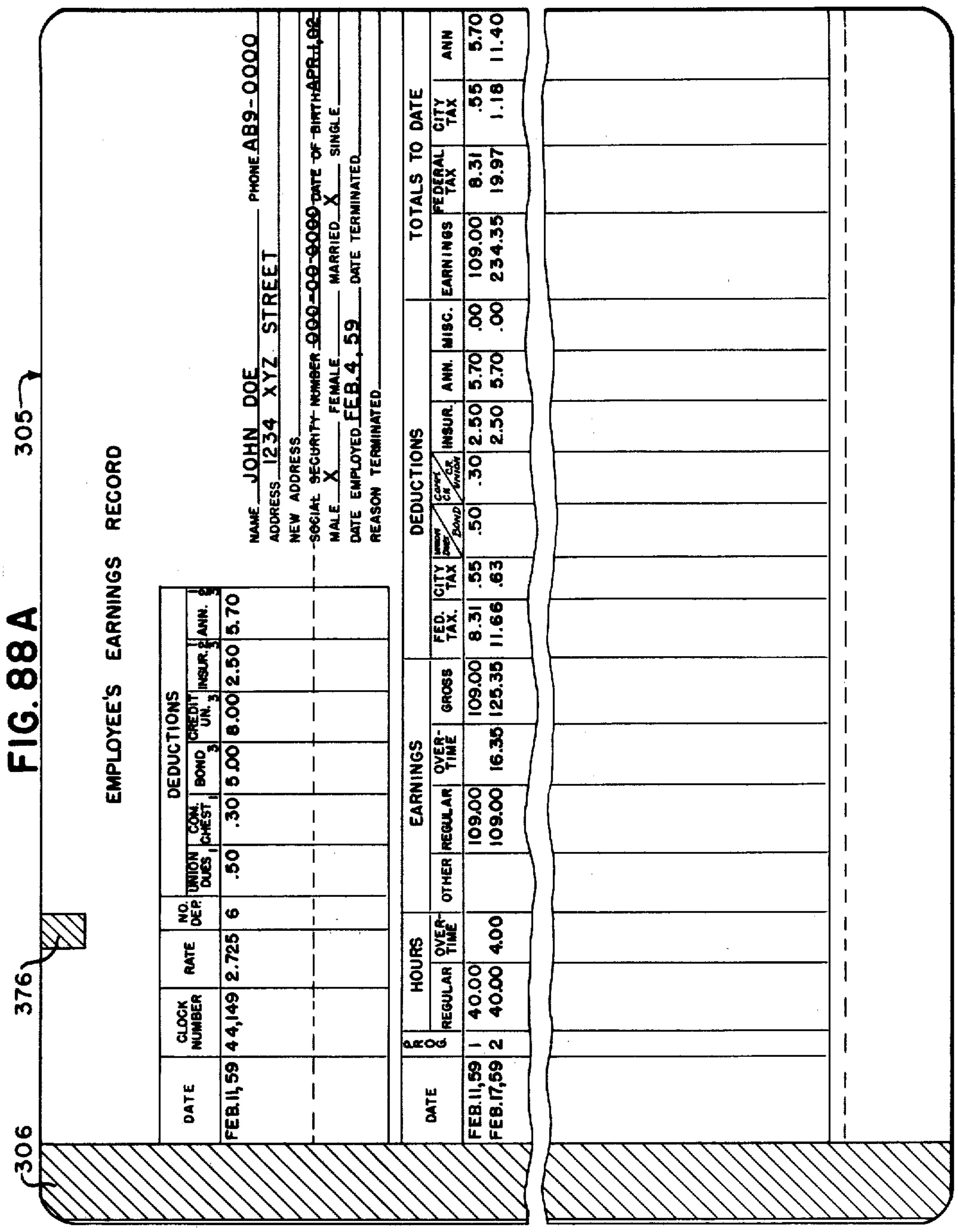


FIG. 88A

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J Lavender
THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 70

FIG. 88 B

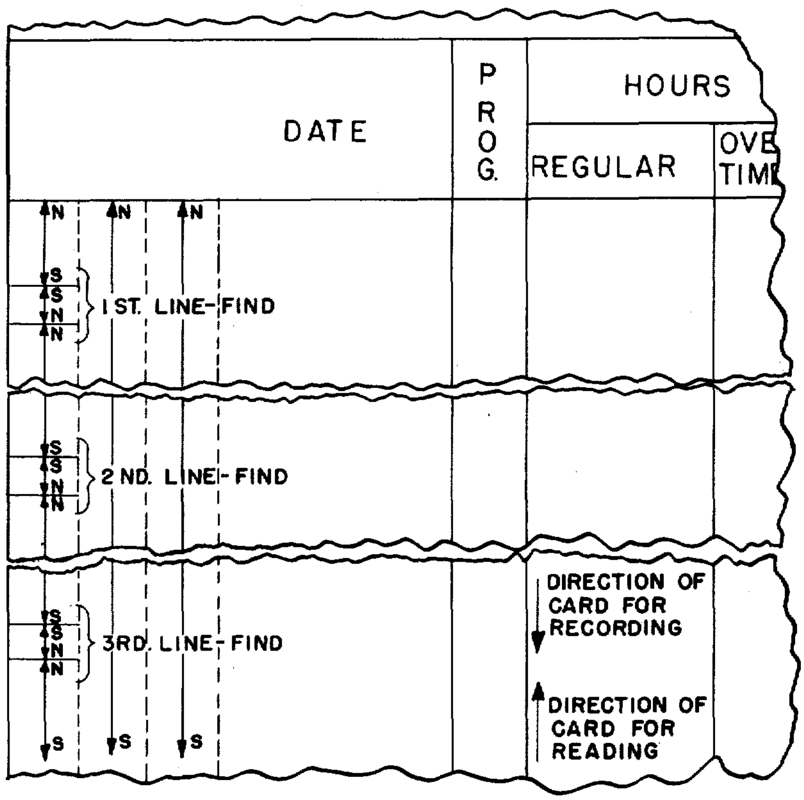
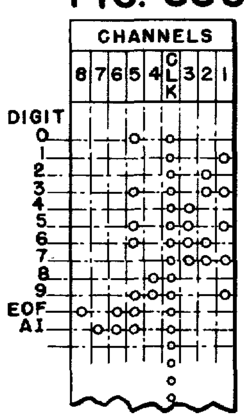


FIG. 88C



INVENTORS
P. B. CLOSE, J. A. COULTER,
L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK,
E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER,
K. A. KINKER, J. D. LYONS,
N. D. MANOR, R. P. MARVIN,
W. L. MILLER, P. L. O'BRIAN,
J. H. RANDALL, E. M. GARNER,
R. O. SATHER, R. L. YOST &
A. B. BRADEN, JR.

BY Louis A. Kline
Lavender

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 71

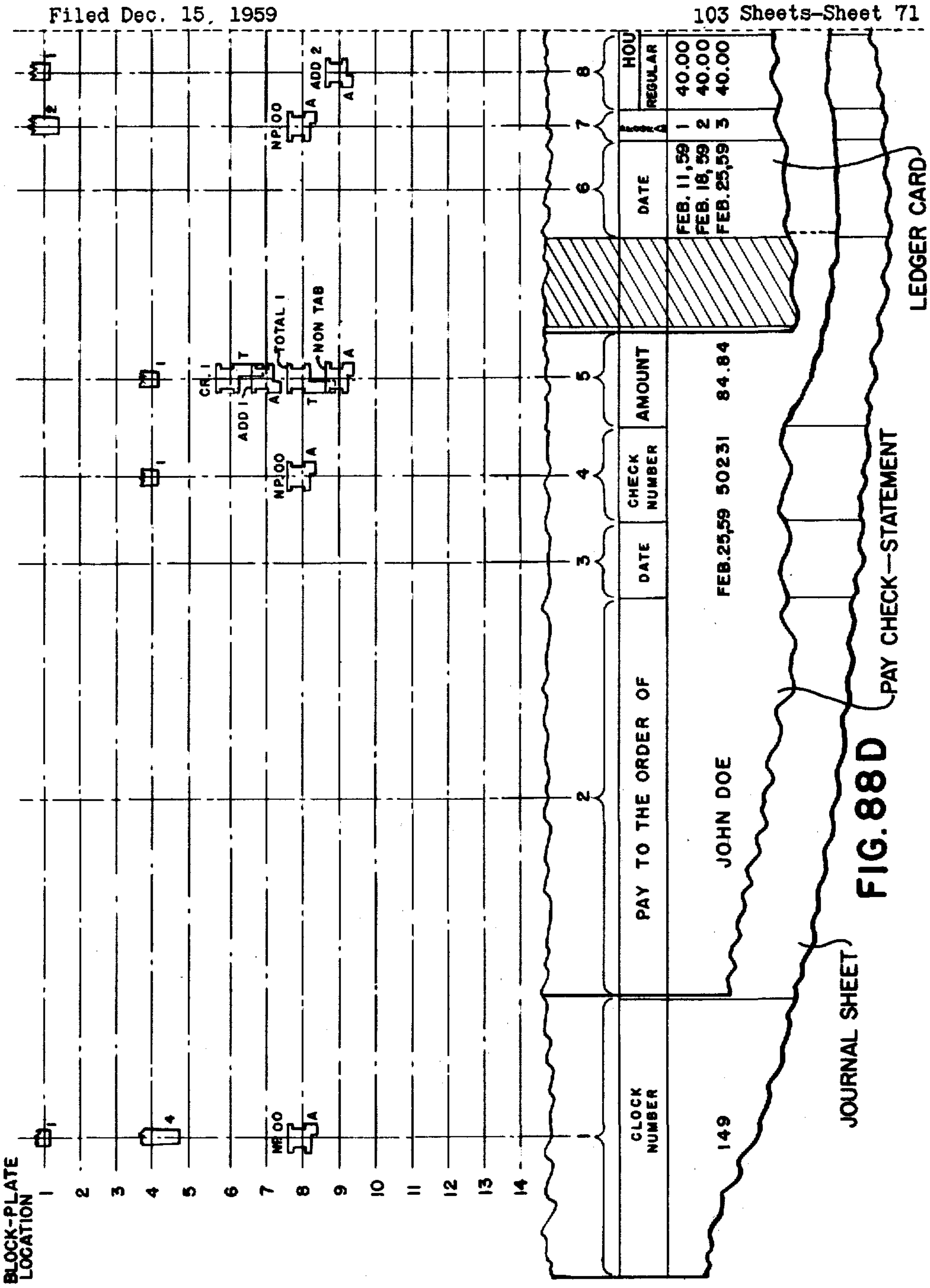


FIG. 88D

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
[signature]
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 72

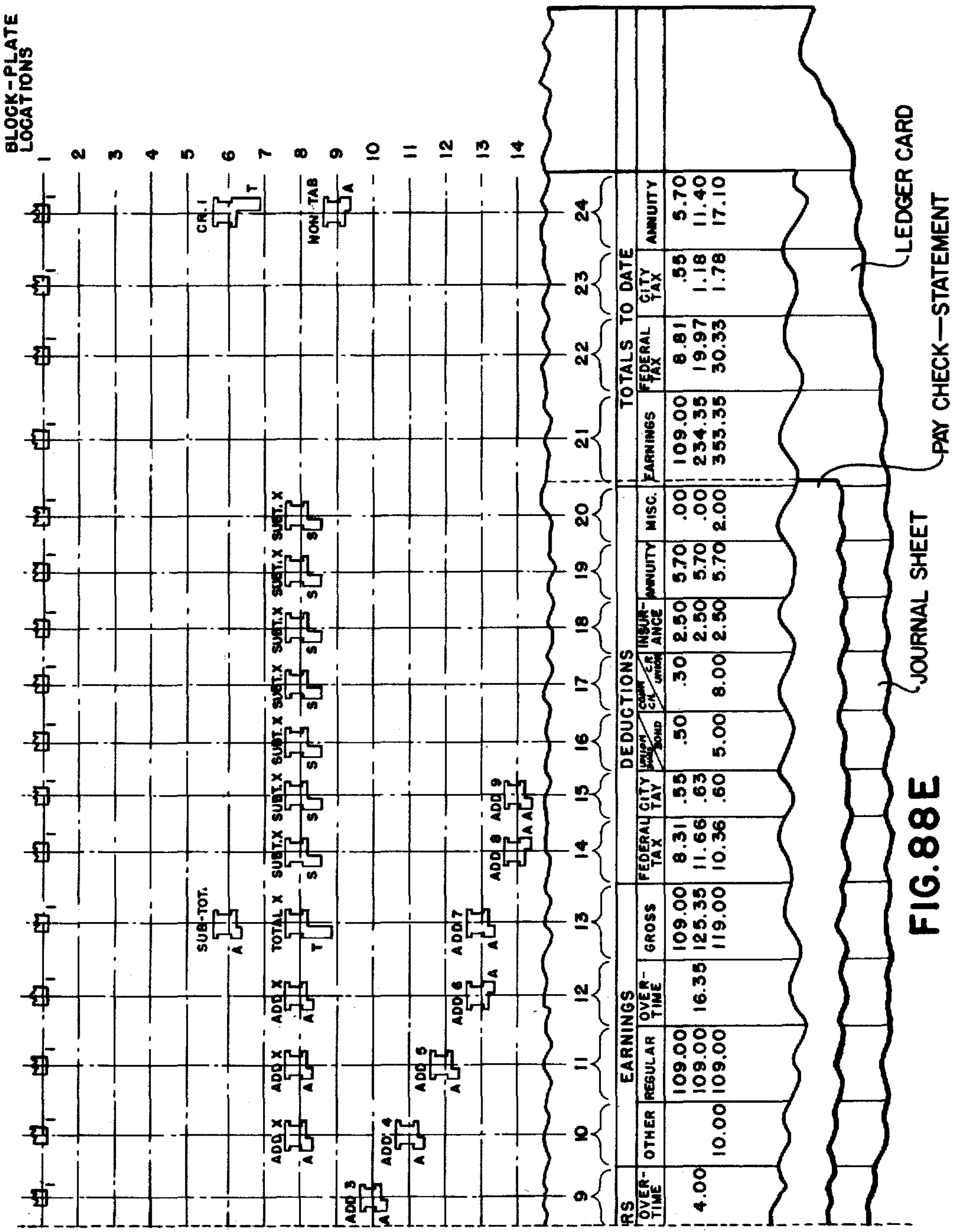


FIG. 88E

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
H Lavender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 73

FIG. 88 F

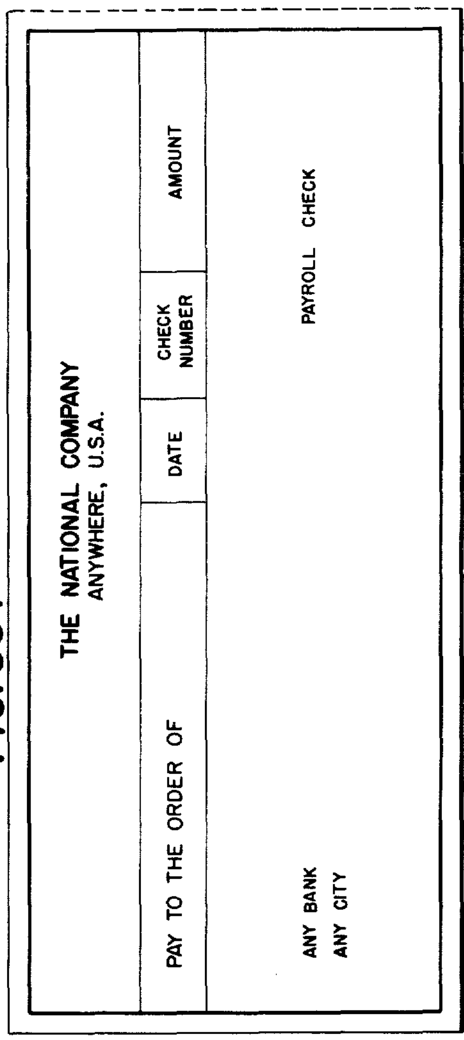
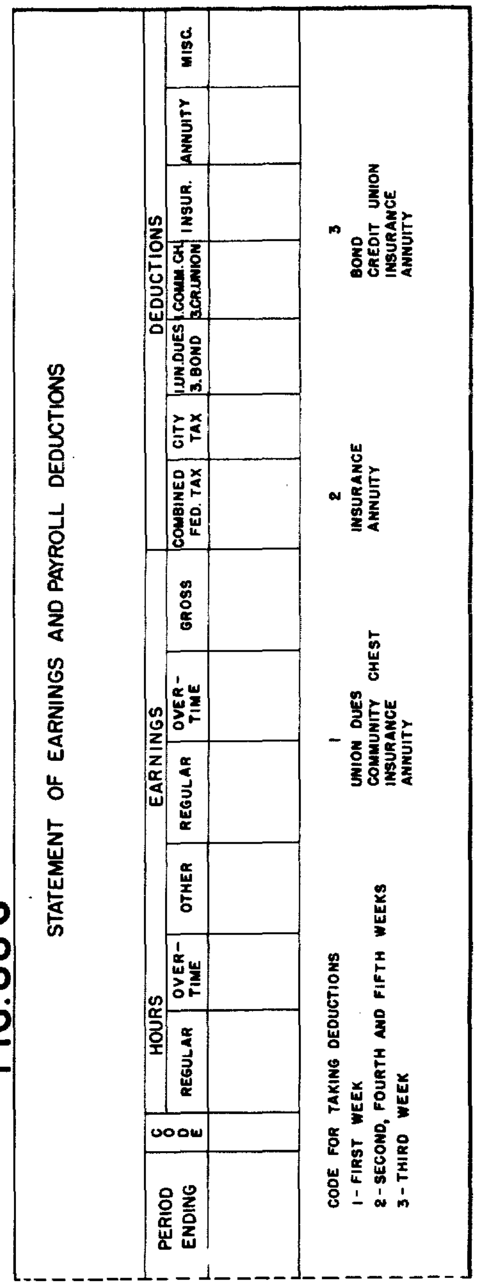


FIG. 88 G

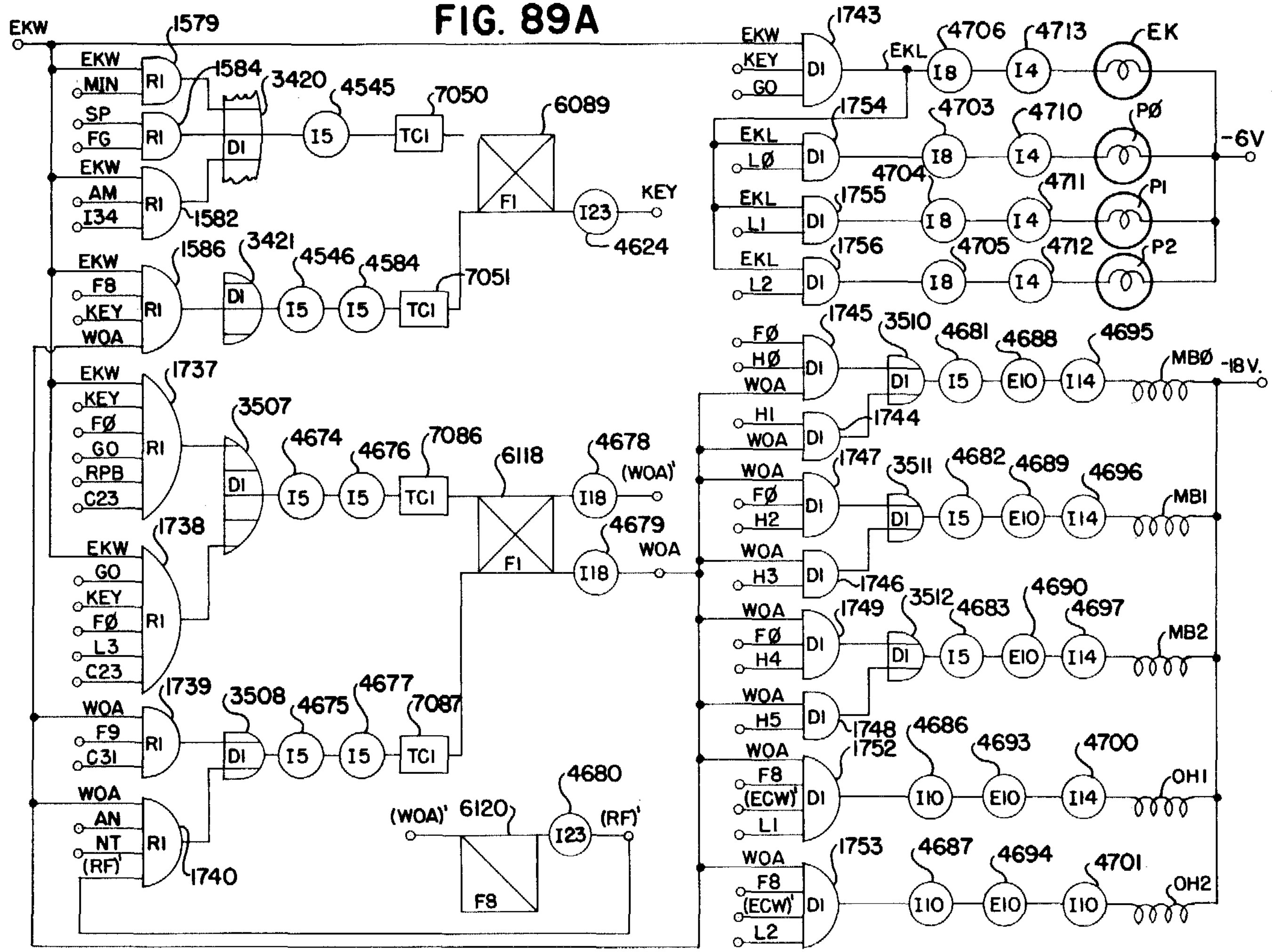


INVENTORS——P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Cavender
THEIR ATTORNEYS

Filed Dec. 15, 1959
ELECTRONIC COMPUTING MACHINE

FIG. 89A

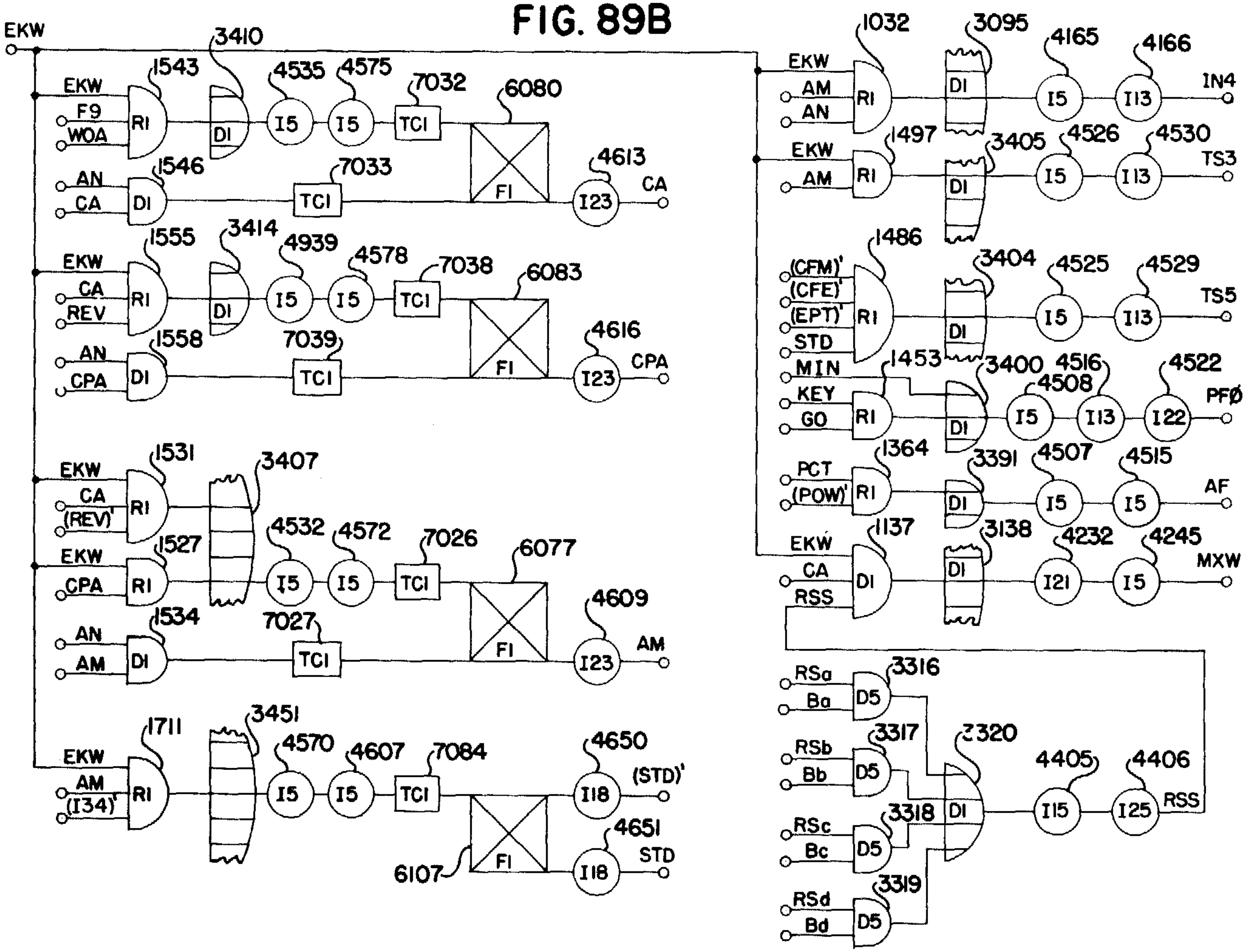


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

Filed Dec. 15, 1959 ELECTRONIC COMPUTING MACHINE 103 Sheets-Sheet 75

FIG. 89B



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 76

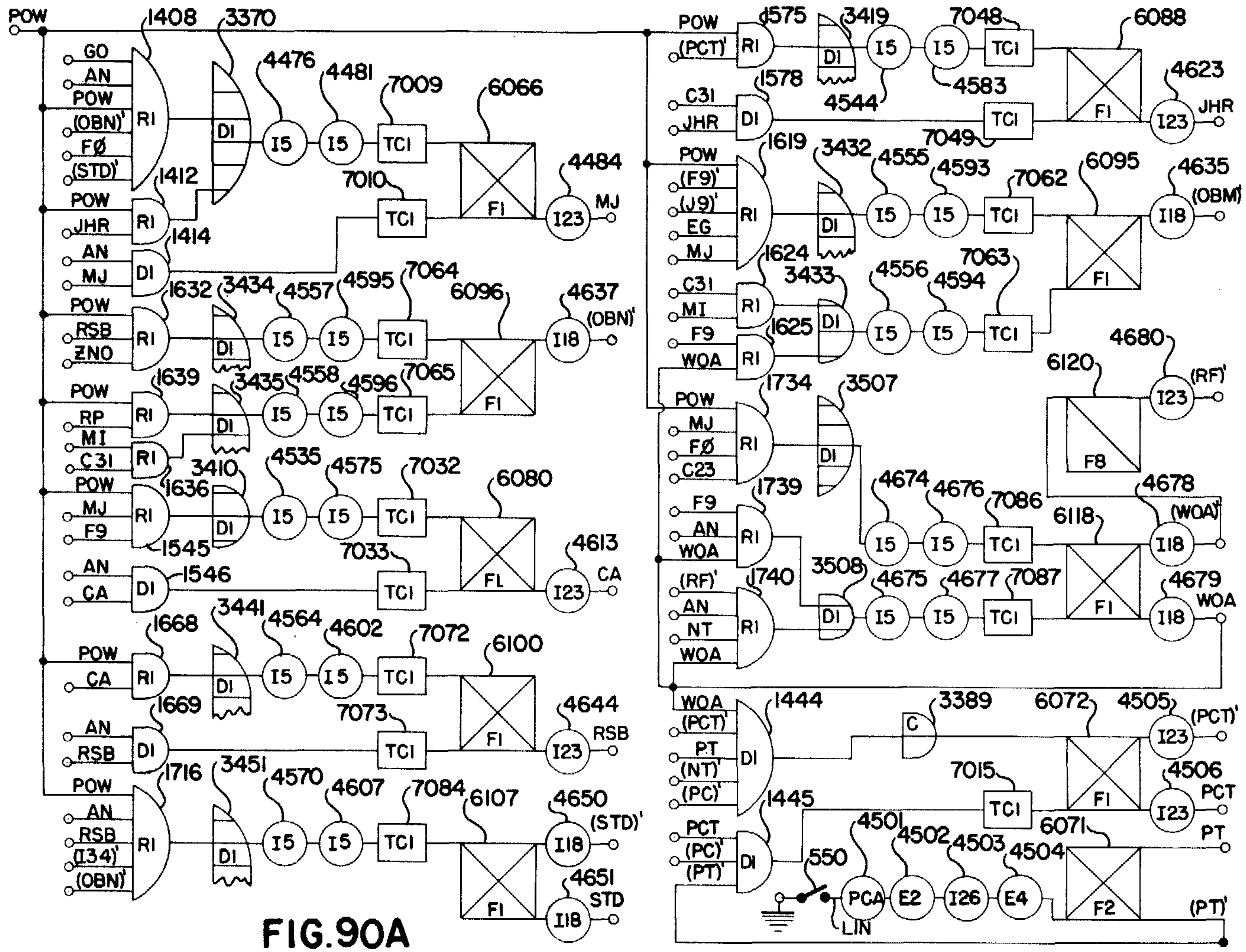


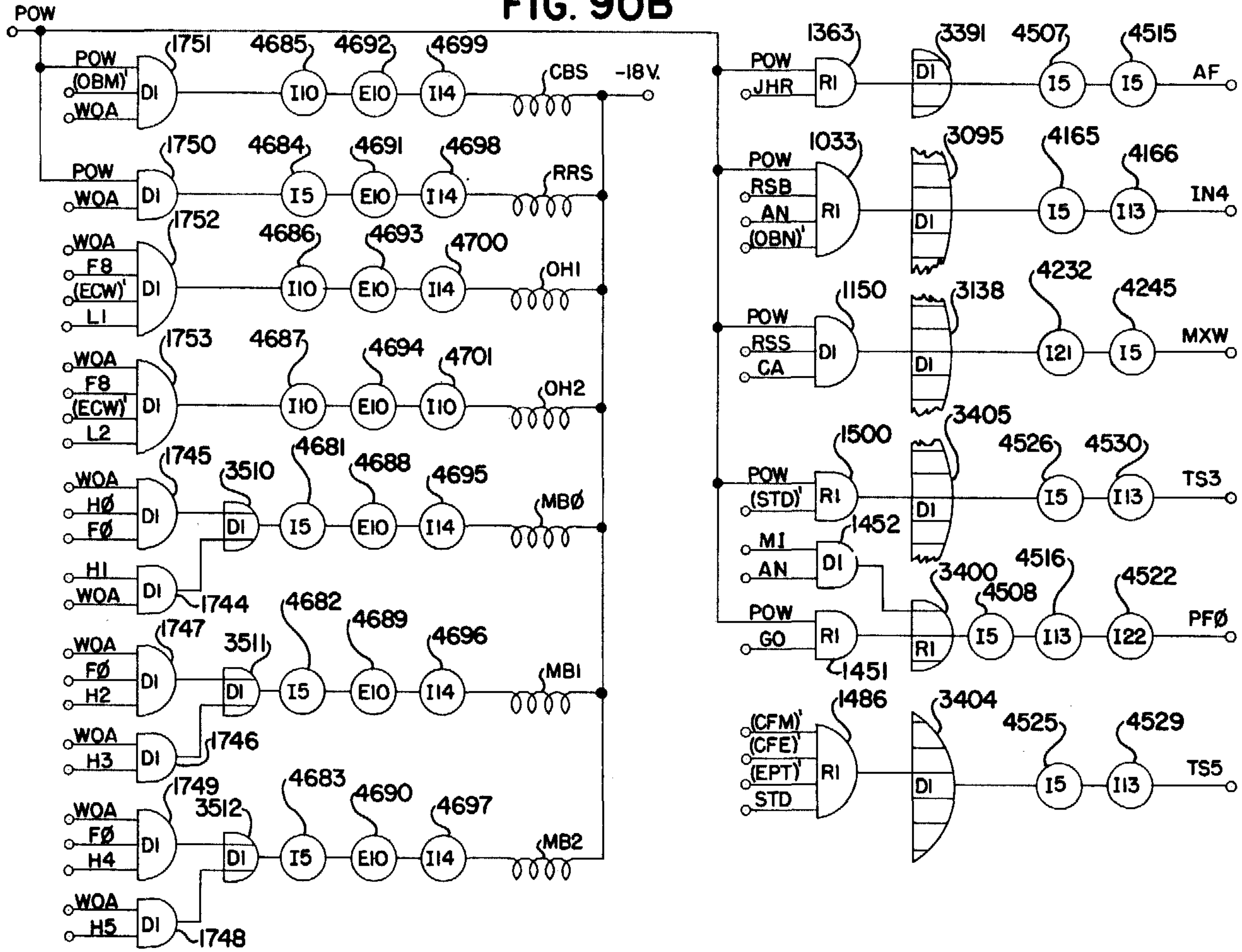
FIG.90A

INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK, F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN, C.S. JENKINS, W.L. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR, R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL, E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 90B

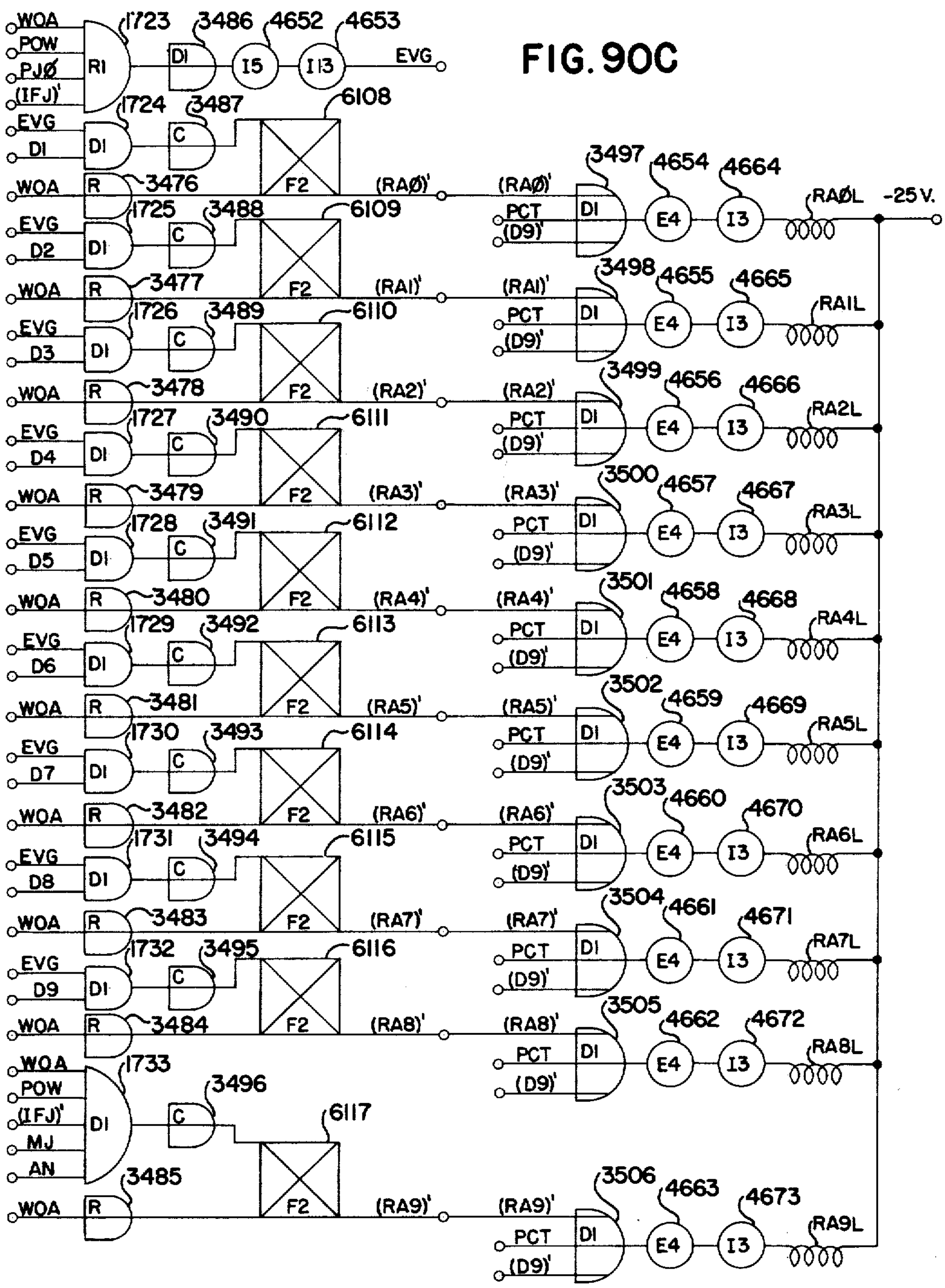


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 78

FIG. 90C

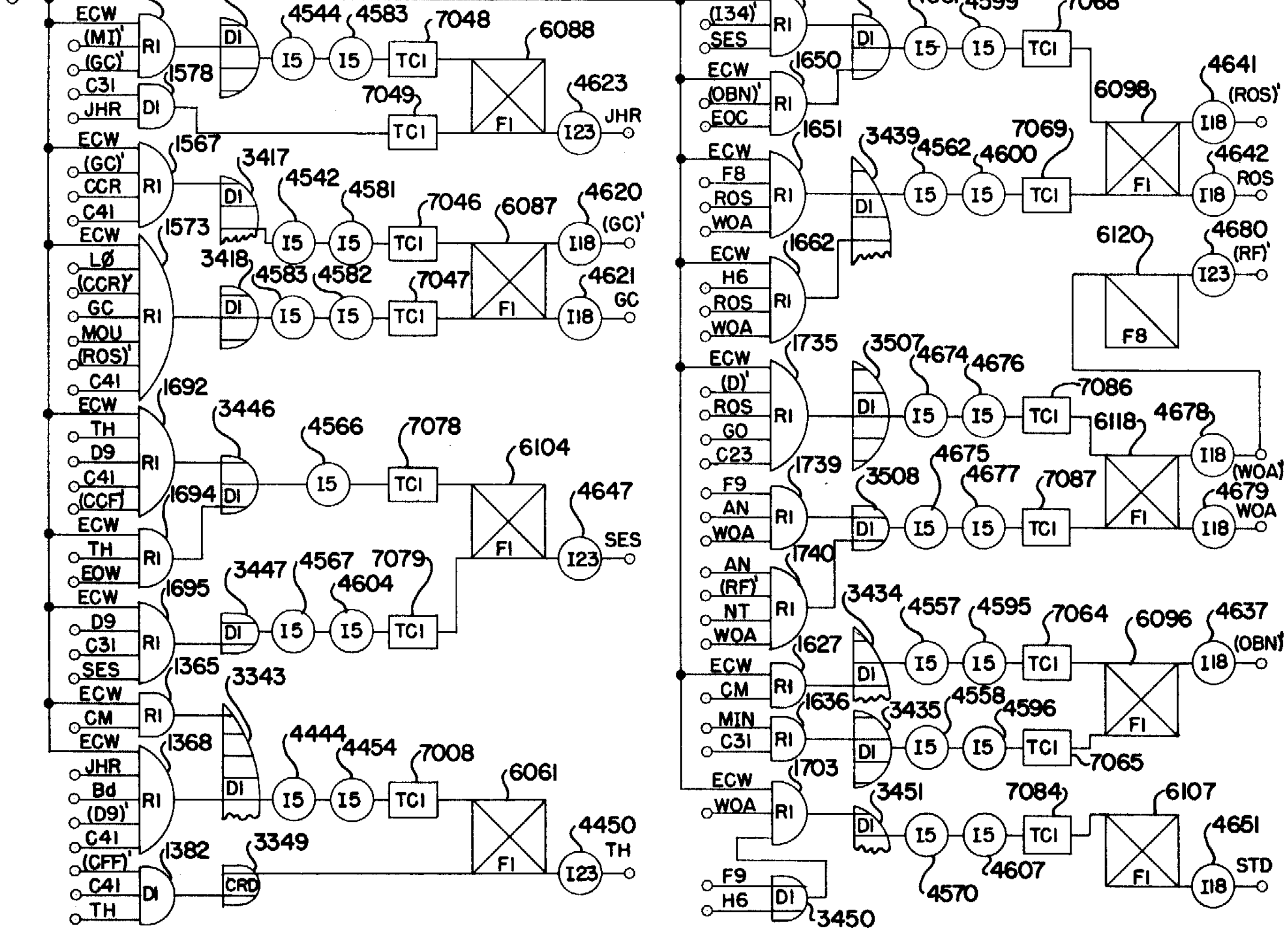


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Cavender

THEIR ATTORNEYS

FIG. 91A

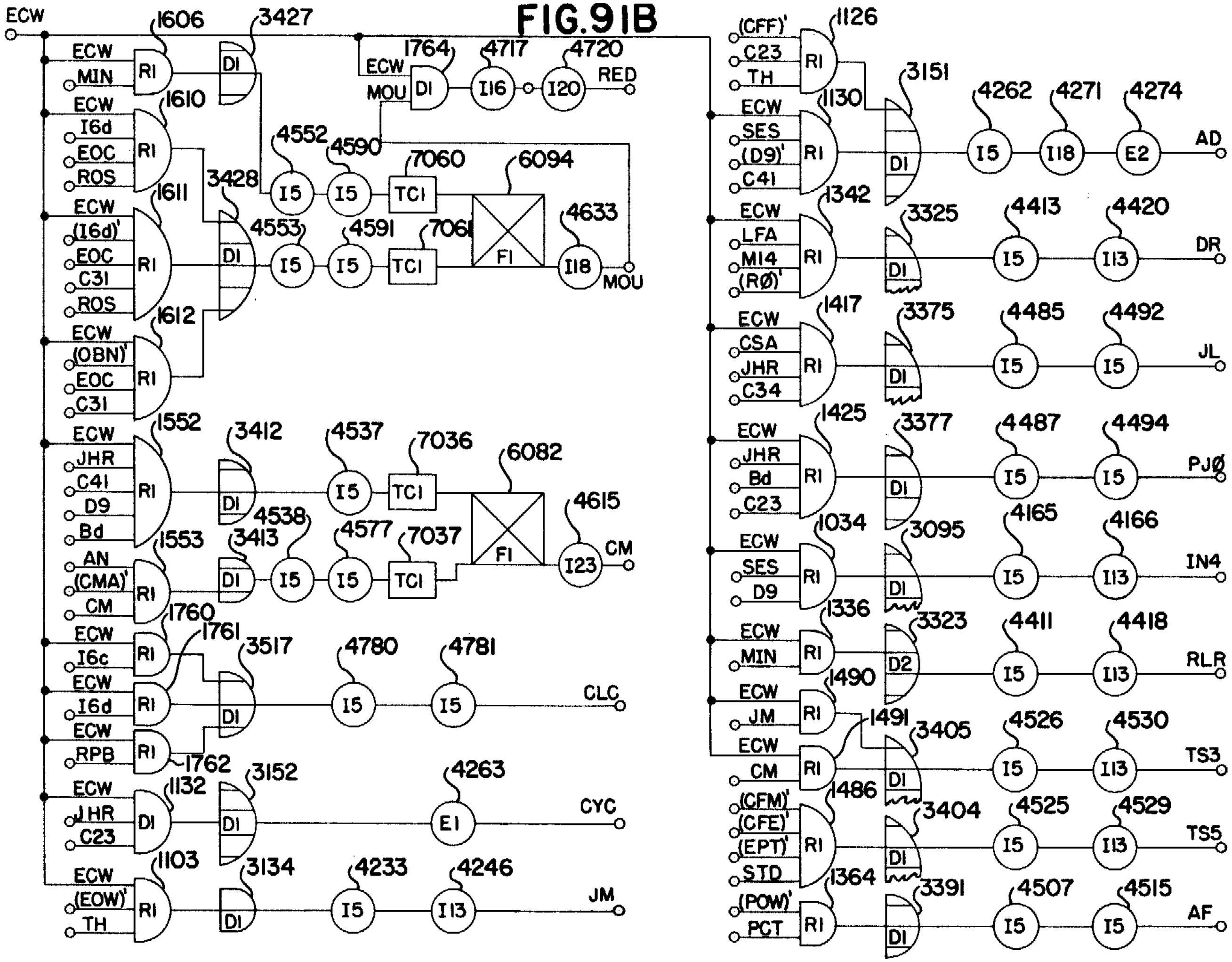


INVENTORS— P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 91B



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

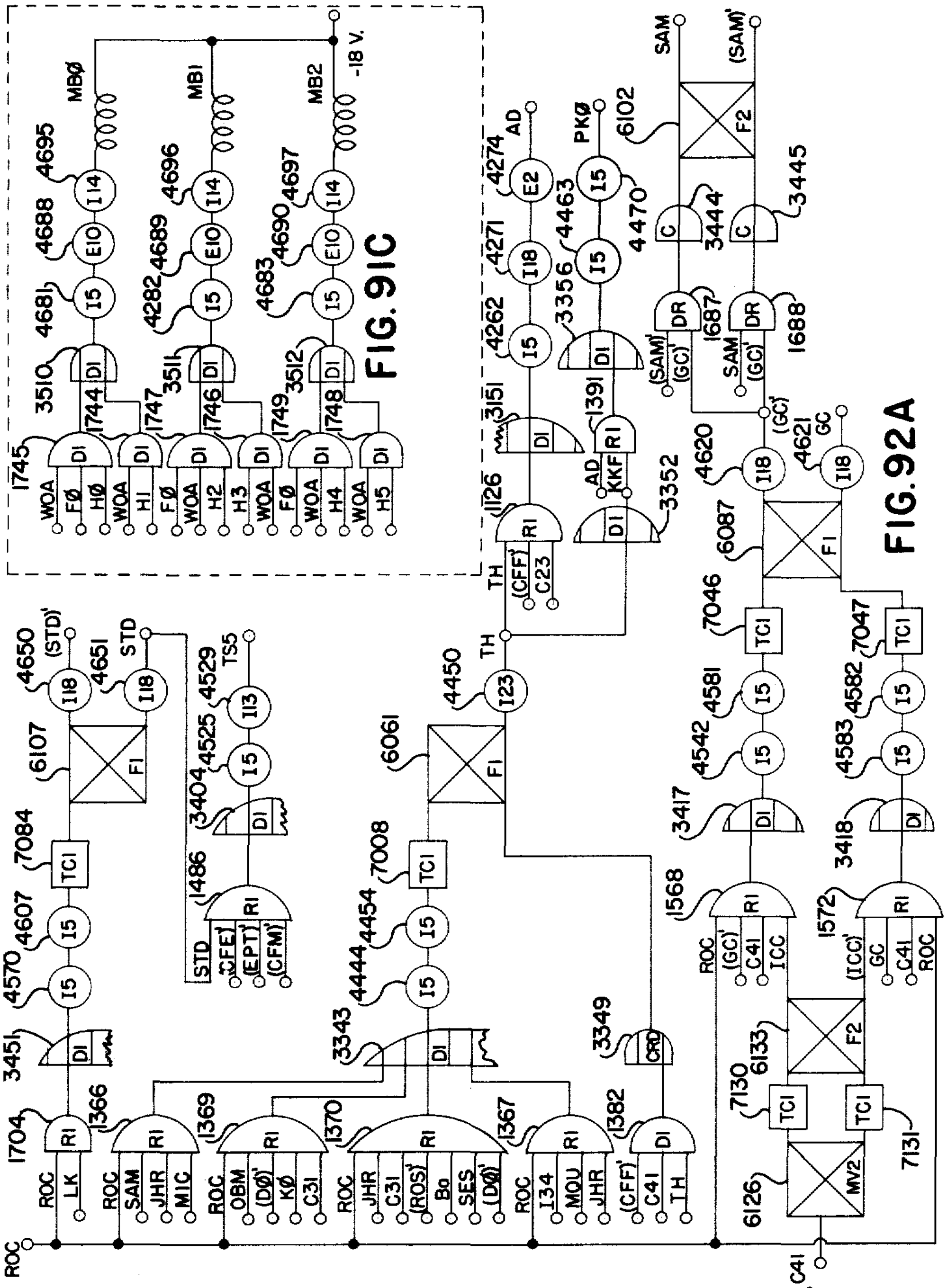


FIG. 91C

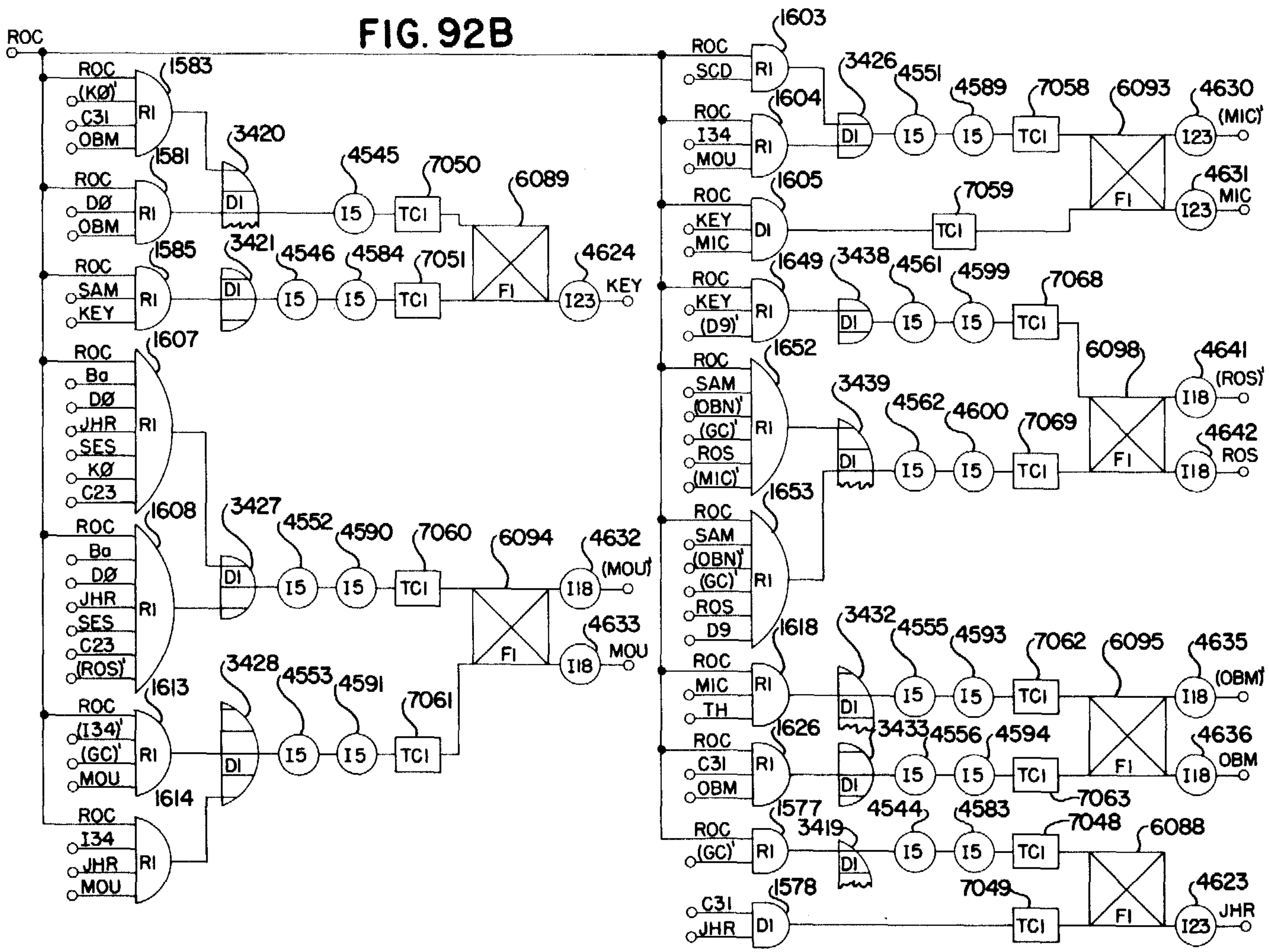
FIG. 92A

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Cavender

THEIR ATTORNEYS

FIG. 92B

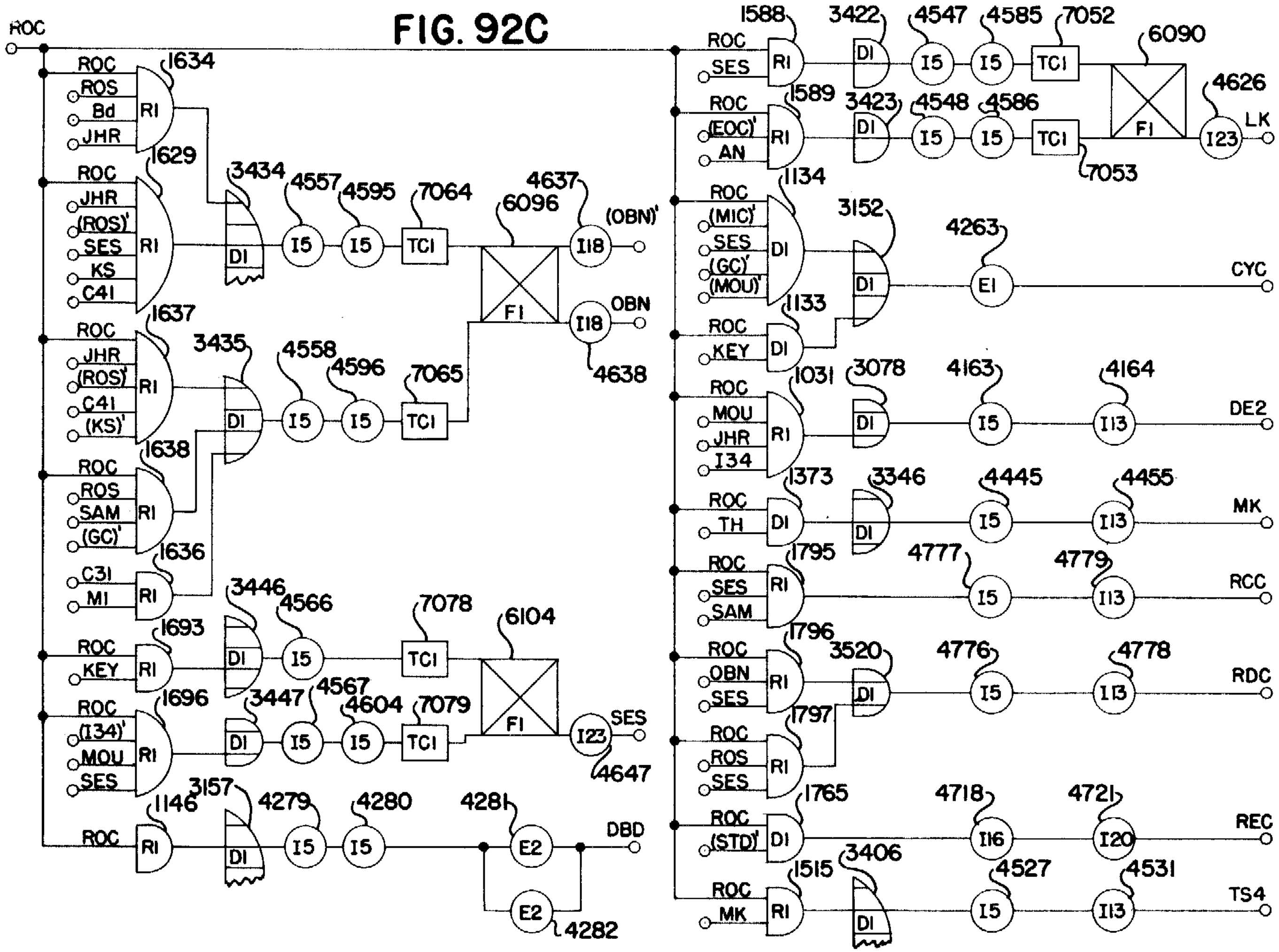


INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN,
C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR,
R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL,
E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.

BY

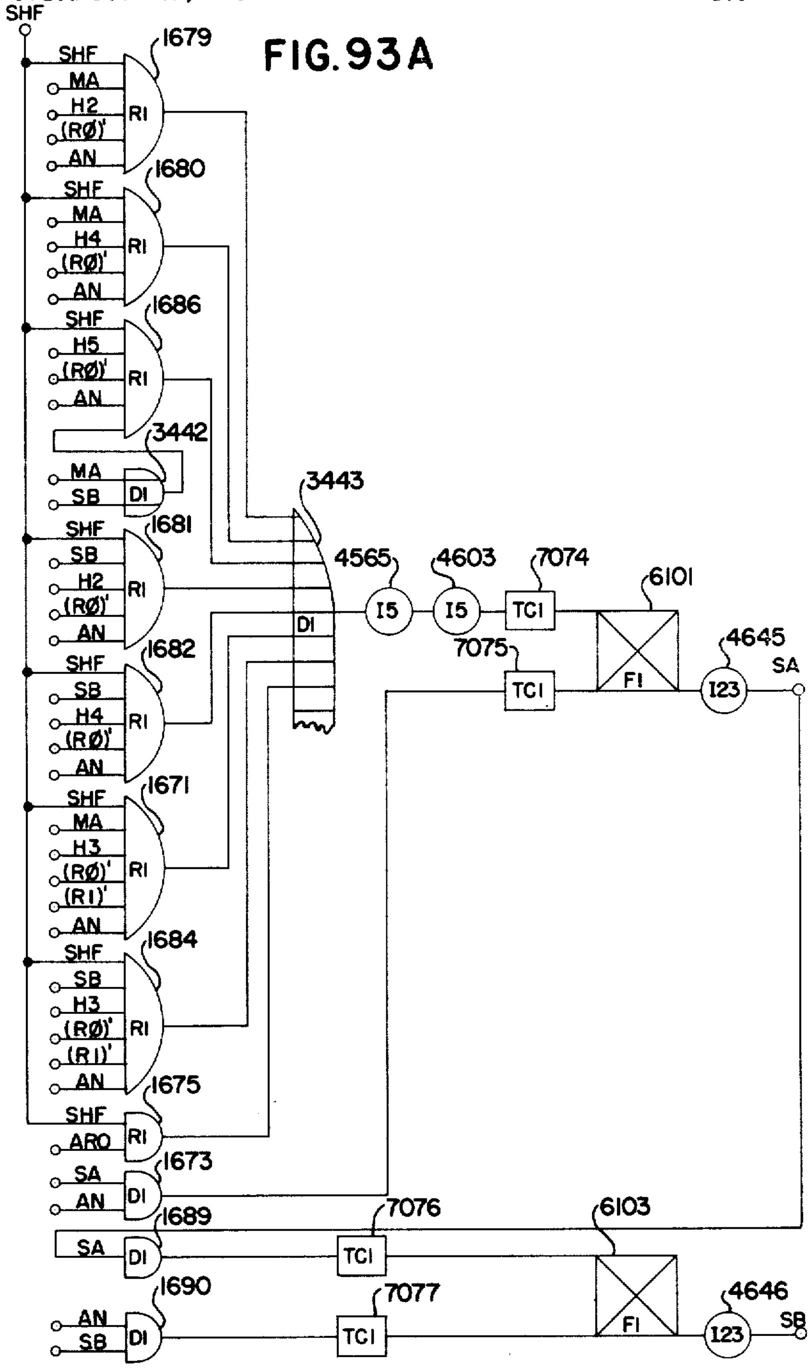
THEIR ATTORNEYS

FIG. 92C



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

FIG. 93A



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
J. Cavender
THEIR ATTORNEYS

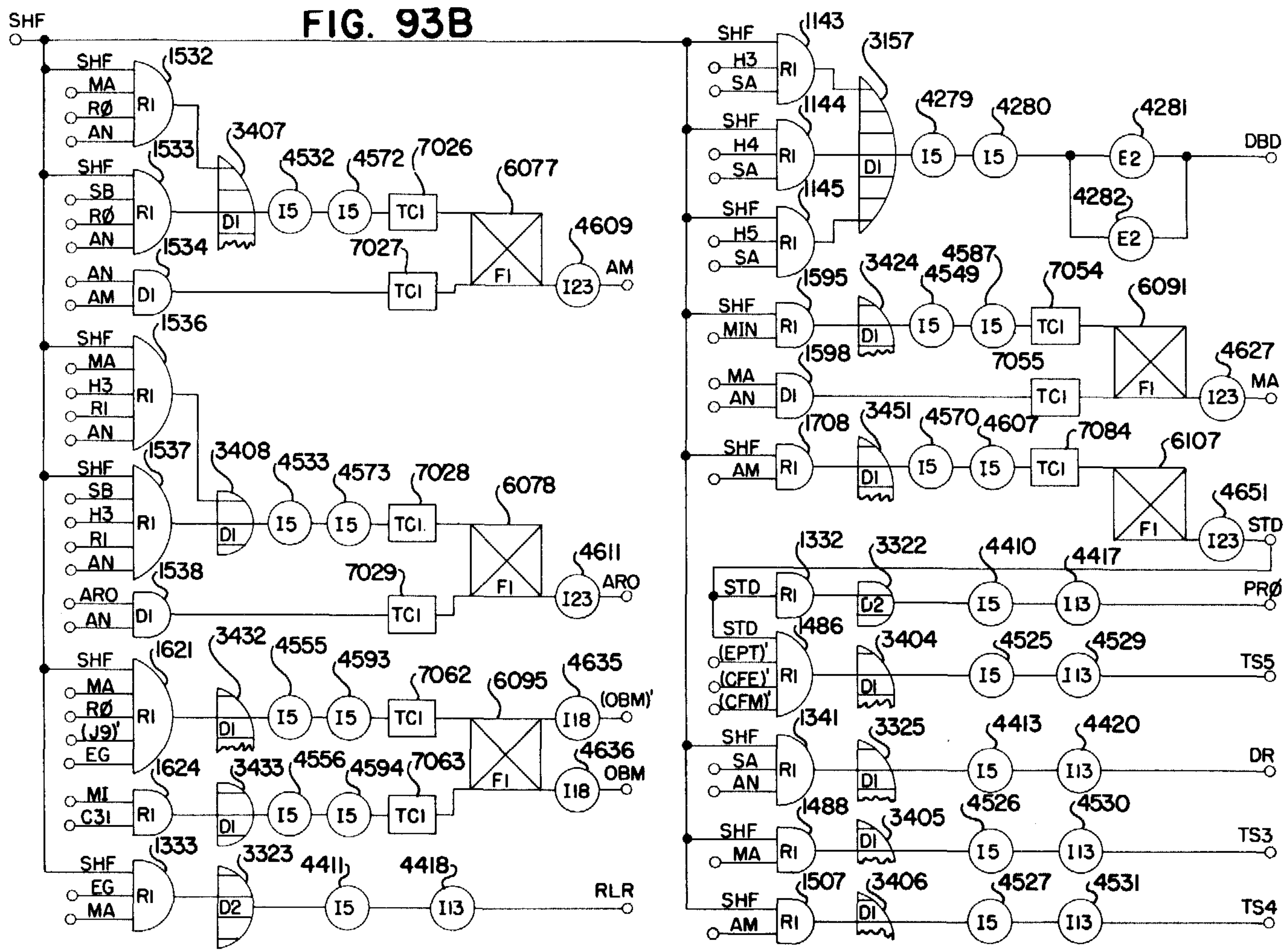
Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

Filed Dec. 15, 1959

ELECTRONIC COMPUTING MACHINE

103 Sheets-Sheet 85

FIG. 93B



INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK, F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN, C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR, R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL, E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.

BY

THEIR ATTORNEYS

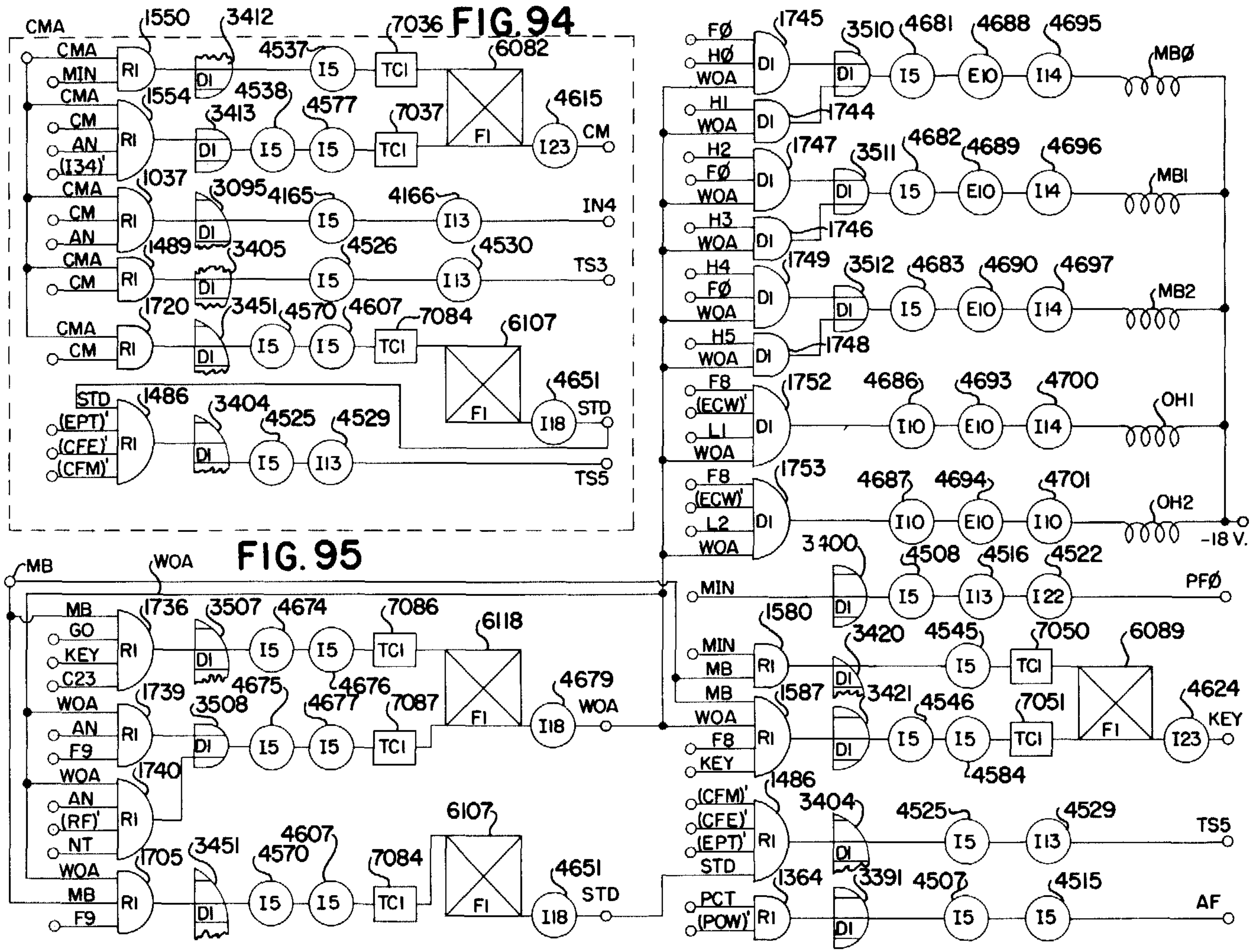
Nov. 26, 1963 P. B. CLOSE ETAL 3,112,394

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 86

FIG. 94

FIG. 95



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 96

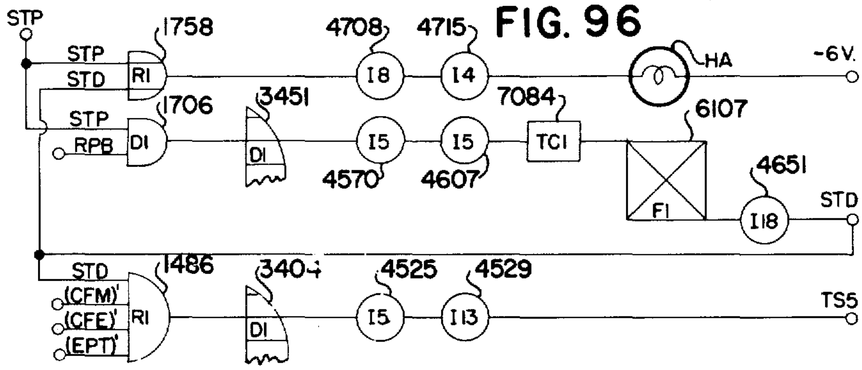
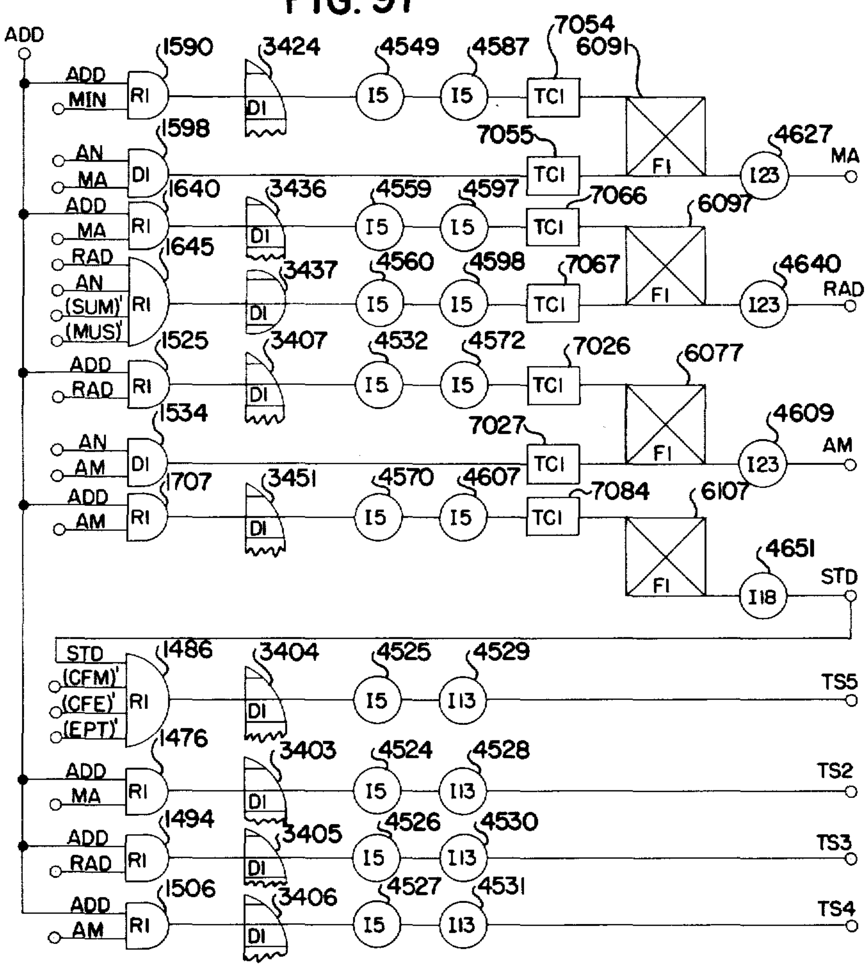


FIG. 97



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
J. Cavender
THEIR ATTORNEYS

FIG. 98

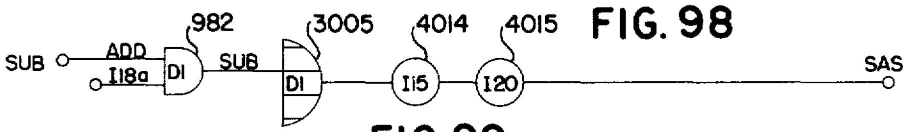
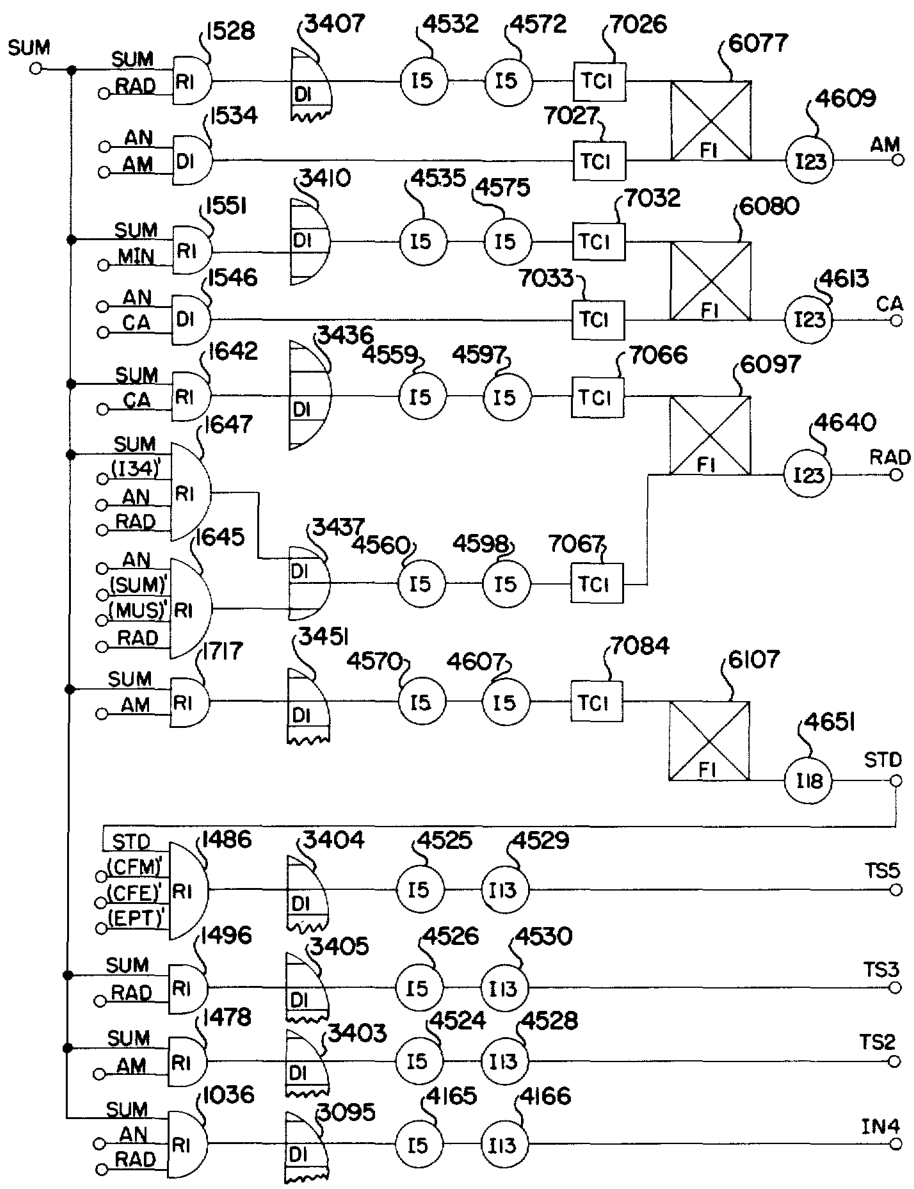


FIG. 99



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Hauender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 89

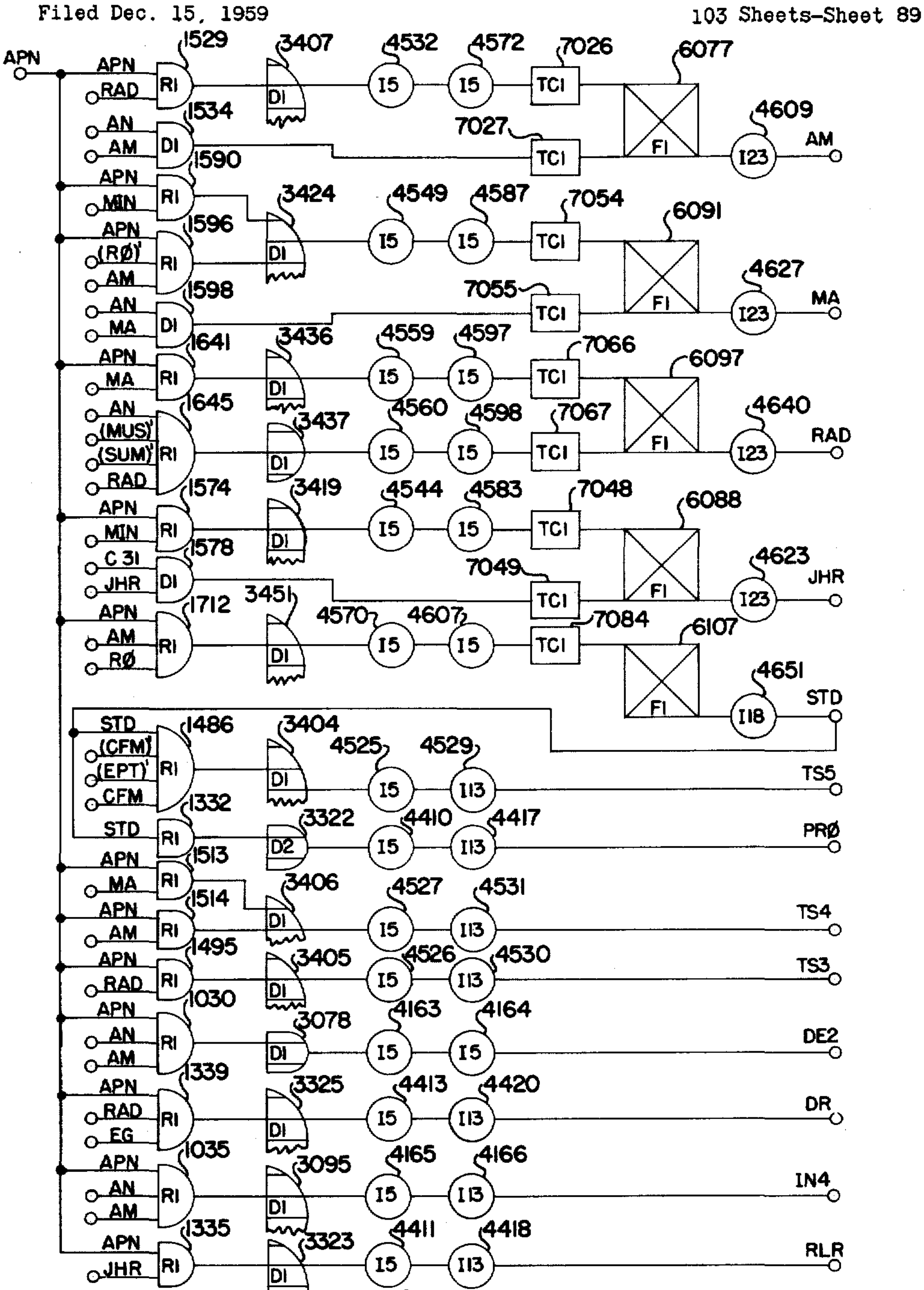
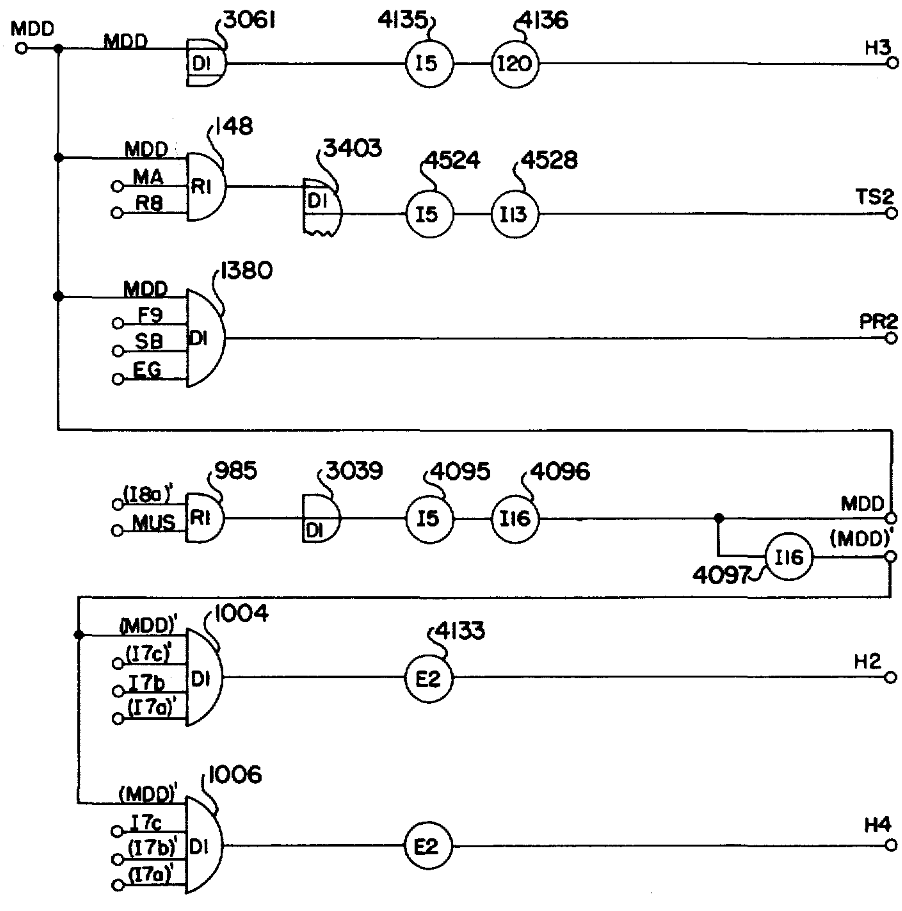


FIG. 100

INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Hauender
THEIR ATTORNEYS

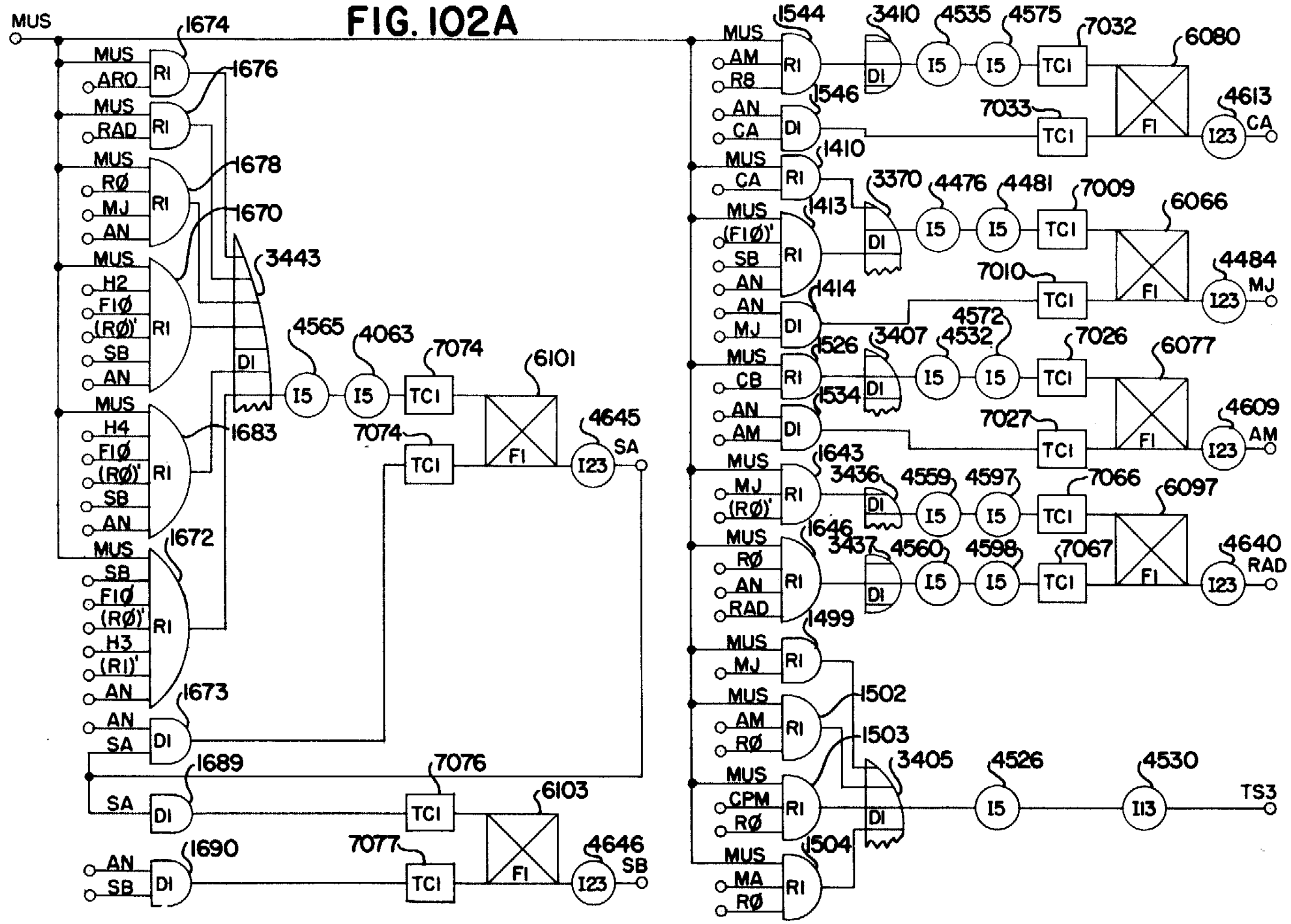
FIG. 101



INVENTORS— P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
THEIR ATTORNEYS

FIG. 102A

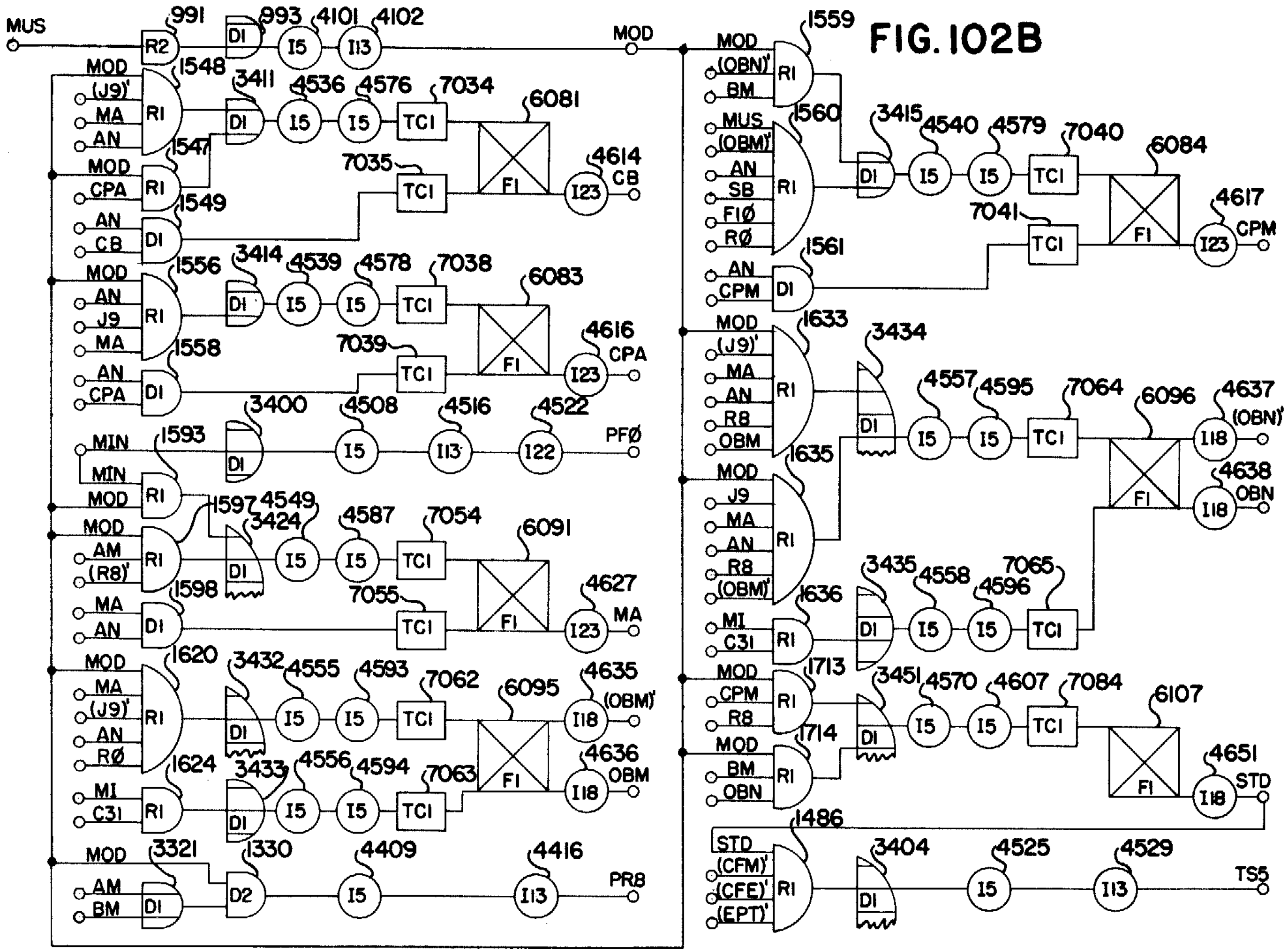


INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN,
C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR,
R.P. MARVIN, W. L. MILLER, P.L. O'BRIAN, J.H. PANDALL,
E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.
BY
THEIR ATTORNEYS

Filed Dec. 15, 1959

ELECTRONIC COMPUTING MACHINE

FIG. 102B

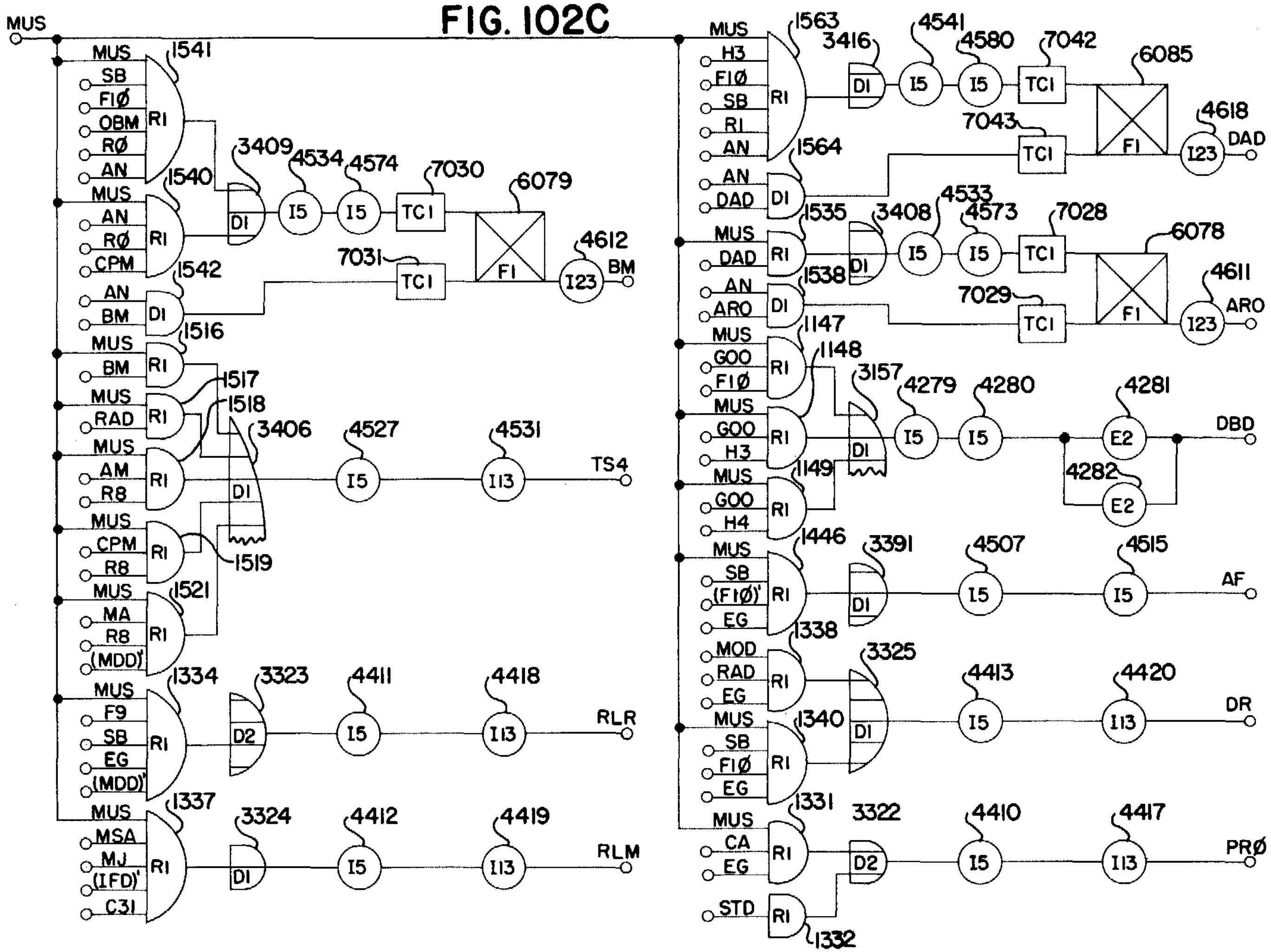


INVENTORS—P.B. CLOSE, J.A. COULTER, L.R. DITMER, R.P. FAHRENBRUCK,
F.R. GOLDAMMER, L.J. GOODRICK, E.V. GULDEN, T.P. HOLLORAN,
C.S. JENKINS, L.D. KILHEFFER, K.A. KINKER, J.D. LYONS, N.D. MANOR,
R.P. MARVIN, W.L. MILLER, P.L. O'BRIAN, J.H. RANDALL,
E.M. GARNER, R.O. SATHER, R.L. YOST & A.B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 102C



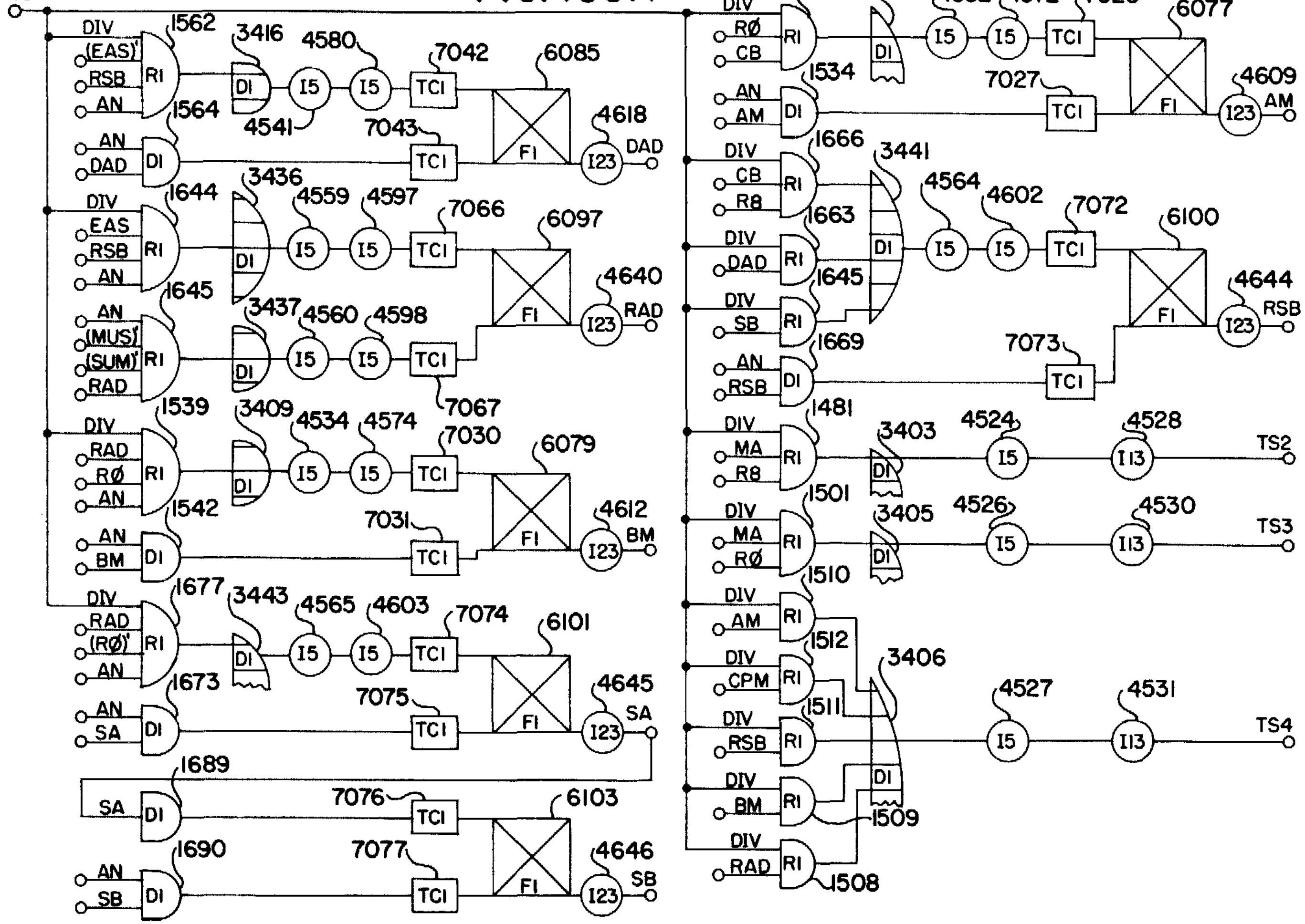
INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, G. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

Filed Dec. 15, 1959

ELECTRONIC COMPUTING MACHINE

103 Sheets-Sheet 94

FIG. 103A

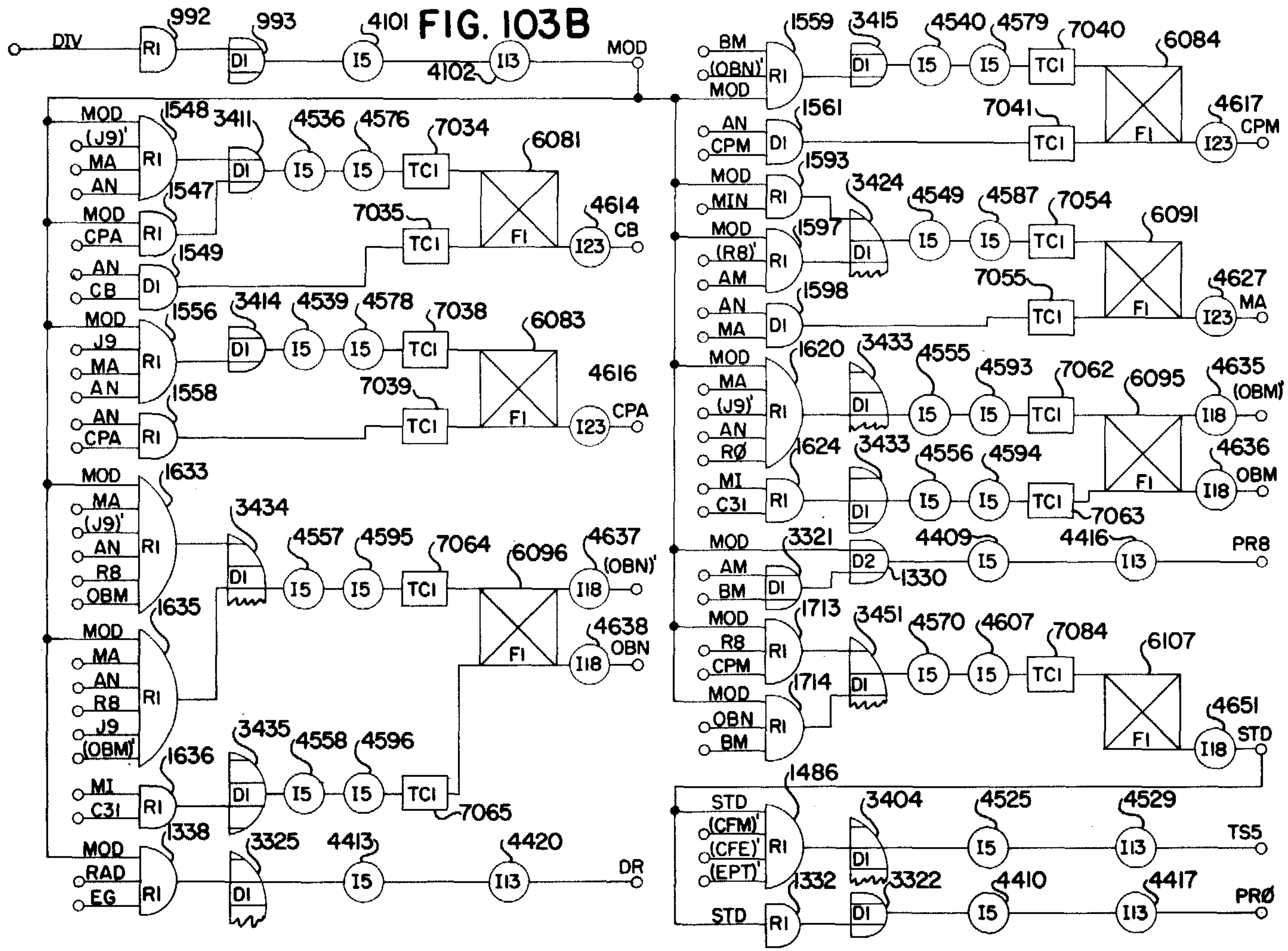


INVENTORS—P.B.CLOSE, J.A.COULTER, L.R.DITMER, R.P.FAHRENBRUCK, F.R.GOLDAMMER, L.J.GOODRICK, E.V.GULDEN, T.P.HOLLORAN, G.S.JENKINS, L.D.KILHEFFER, K.A.KINKER, J.D.LYONS, N.D.MANOR, R.P.MARVIN, W.L.MILLER, P.L.O'BRIAN, J.H.RANDALL, E.M.GARNER, R.O.SATHER, R.L.YOST & A.B.BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 103B

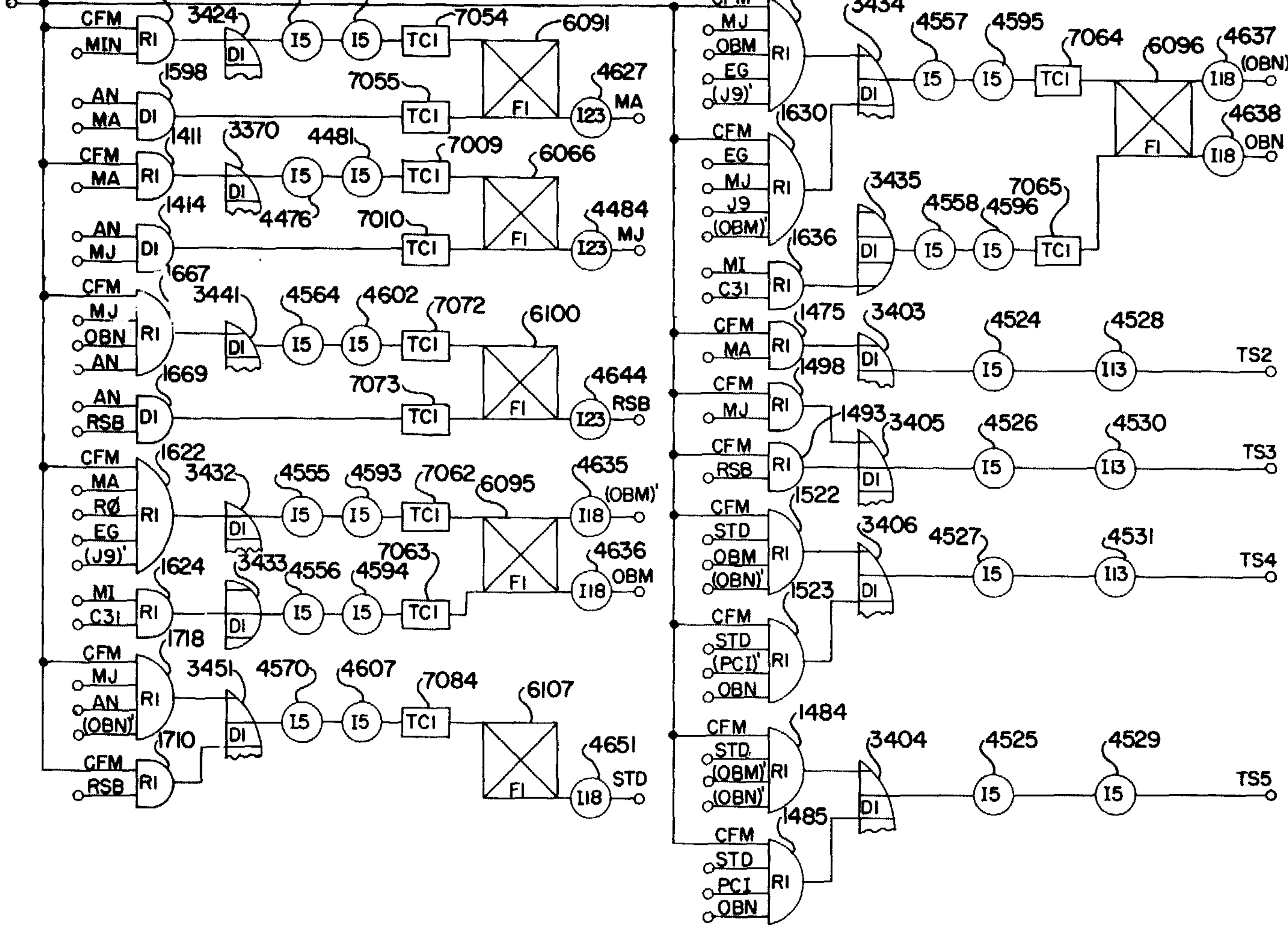


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, G. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 104

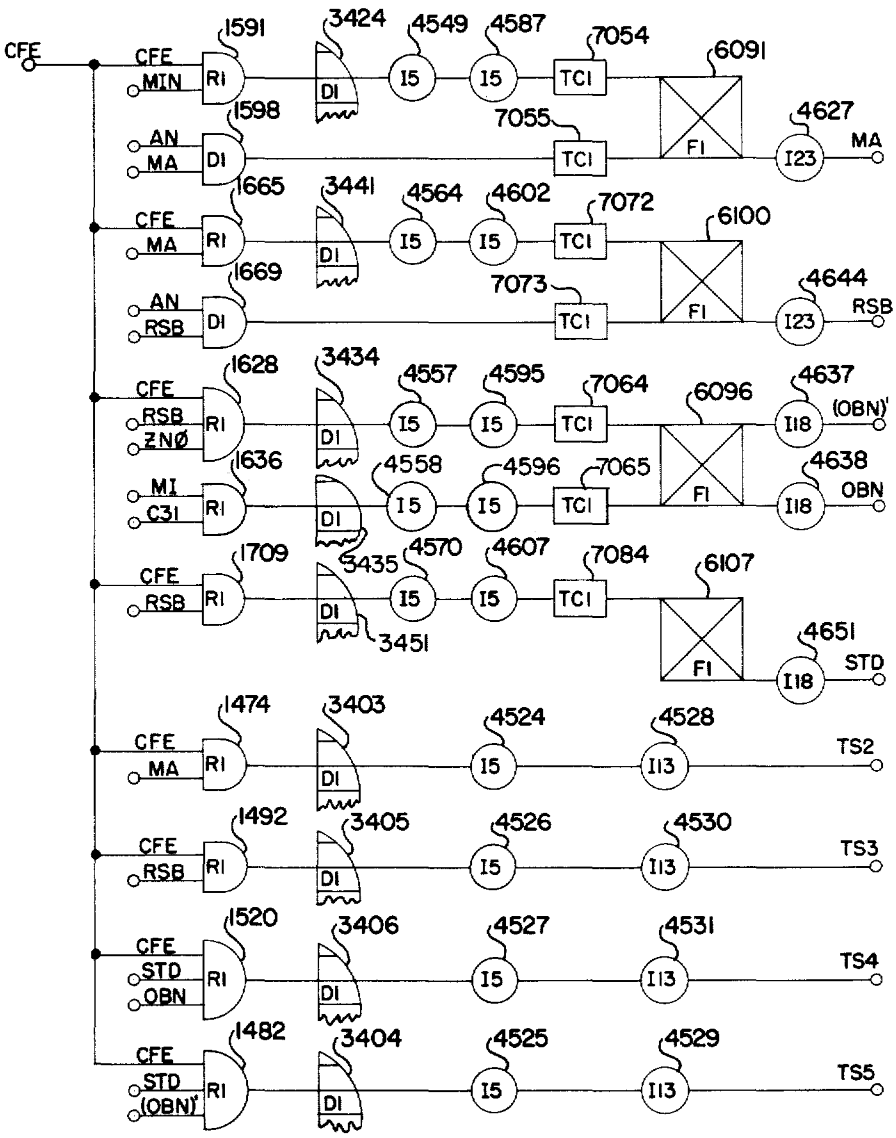


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK,
F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN,
G. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR,
R. P. MARVIN, W. I. MILLER, P. L. O'BRIAN, J. H. RANDALL,
E. M. GARNER, I. O. SATHER, R. L. YOST & A. B. BRADEN, JR.
BY
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 97

FIG. 105



INVENTORS — P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A Kline
Havender
THEIR ATTORNEYS

Nov. 26, 1963

P. B. CLOSE ETAL

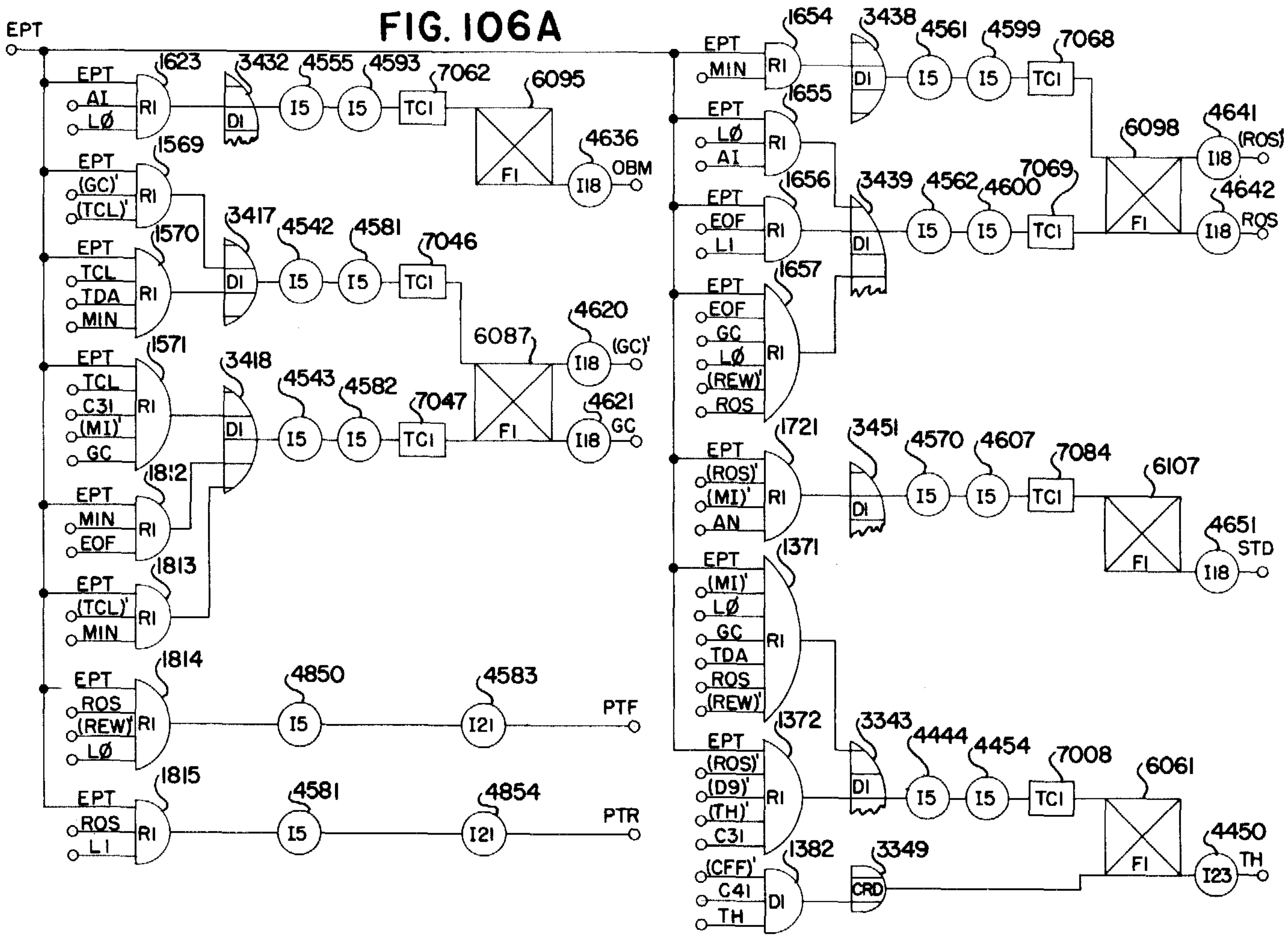
3,112,394

Filed Dec. 15, 1959

ELECTRONIC COMPUTING MACHINE

103 Sheets-Sheet 98

FIG. 106A

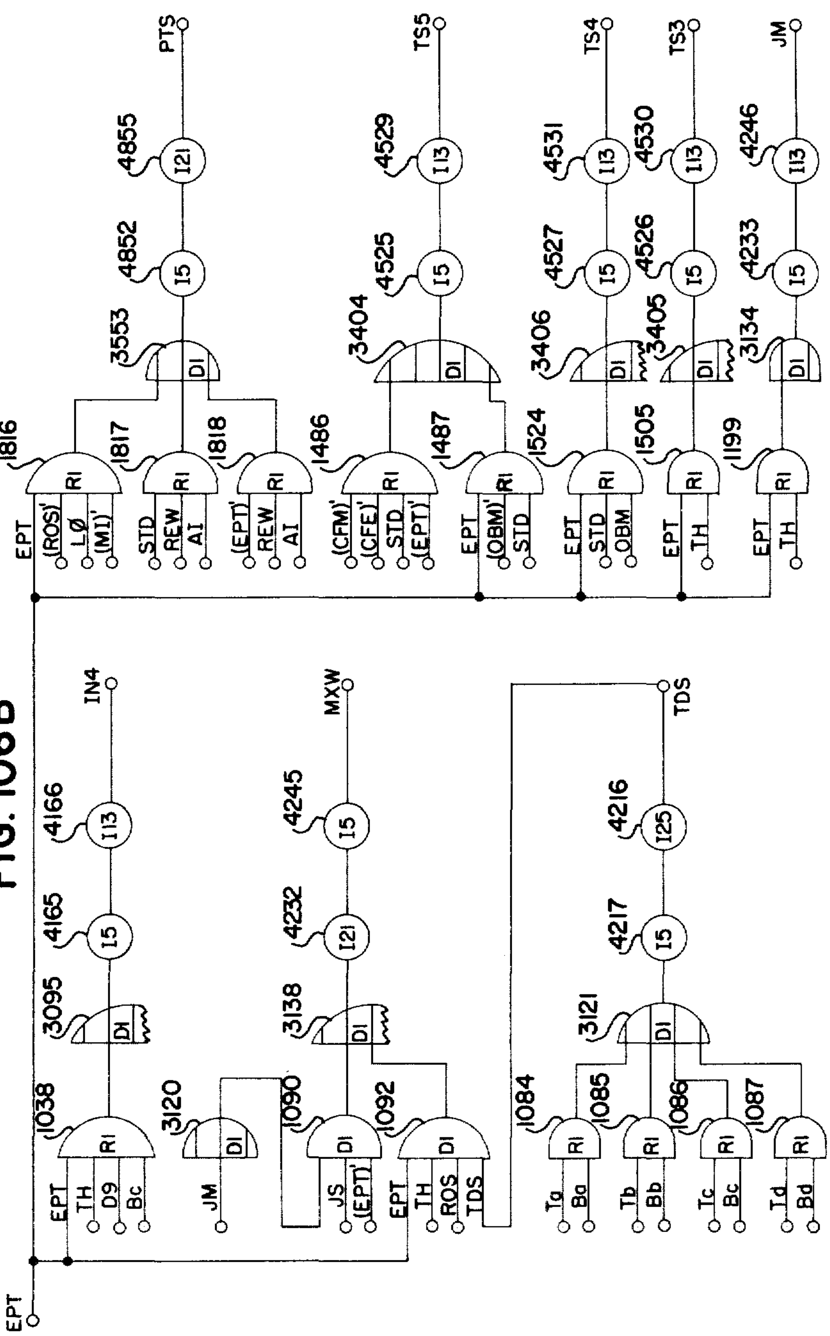


INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY

THEIR ATTORNEYS

FIG. 106B



INVENTORS — P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN

BY Louis A Kline
Stravender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 100

FIG. 107A

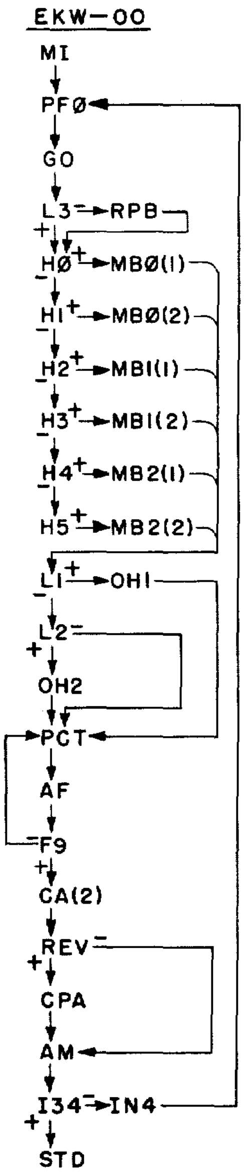
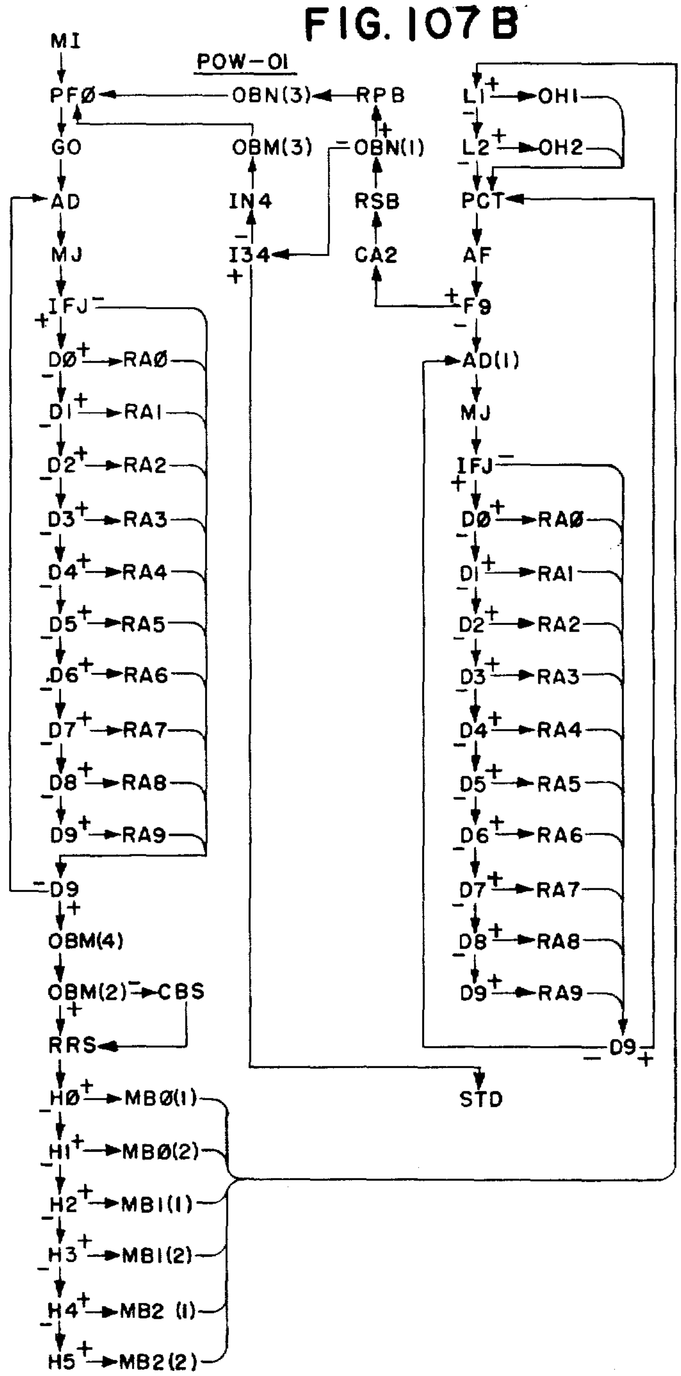


FIG. 107B



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

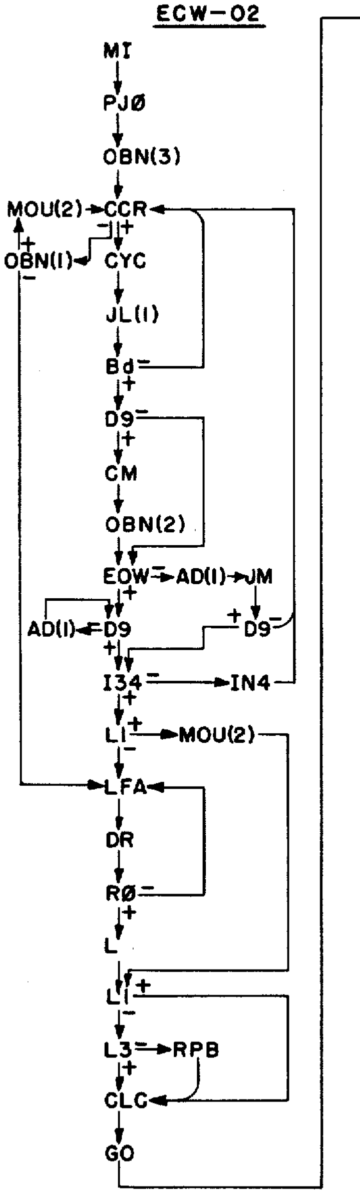
BY Louis A Kline
Hravender
THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 101

FIG. 107C

ECW-02



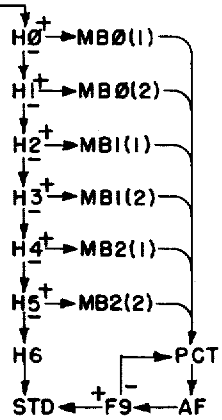


FIG. 107D

ROC-03

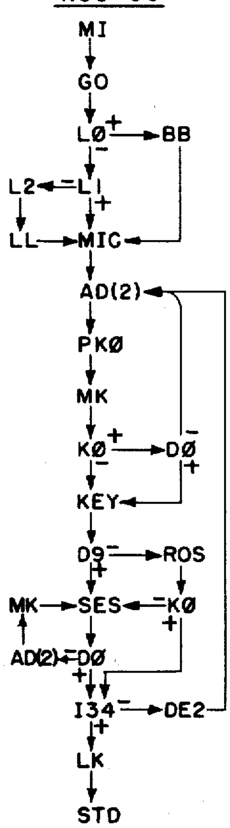


FIG. 107E

SHF-04

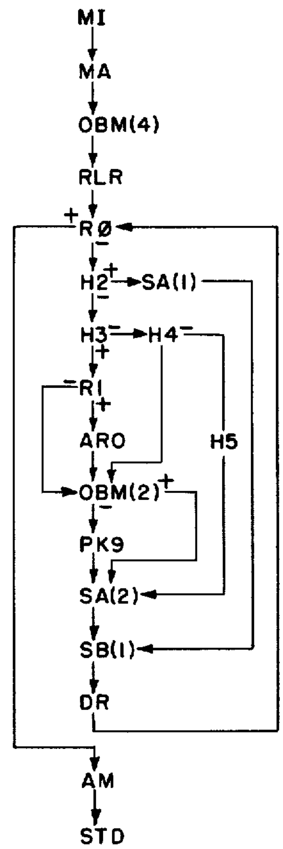
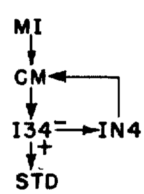


FIG. 107 F

CMA-05



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline
Saunders

THEIR ATTORNEYS

ELECTRONIC COMPUTING MACHINE

Filed Dec. 15, 1959 103 Sheets-Sheet 102

FIG. 107G

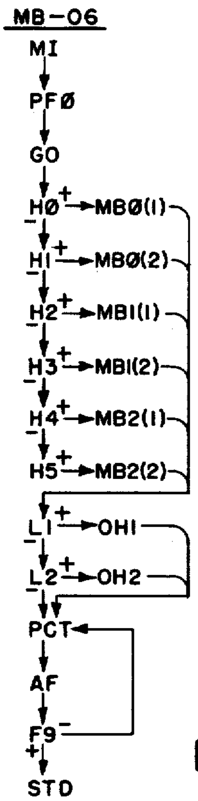


FIG. 107H

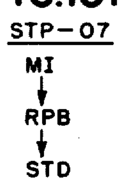


FIG. 107I

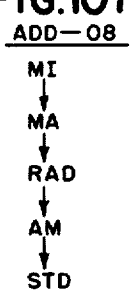


FIG. 107J

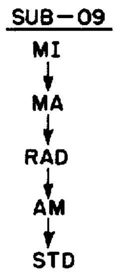


FIG. 107K

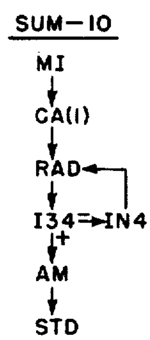


FIG. 107L

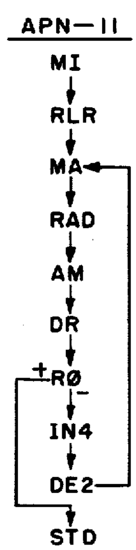


FIG. 107M

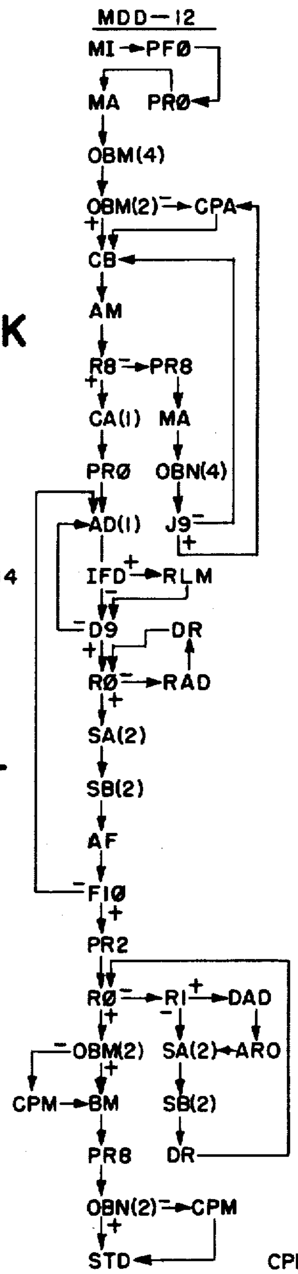
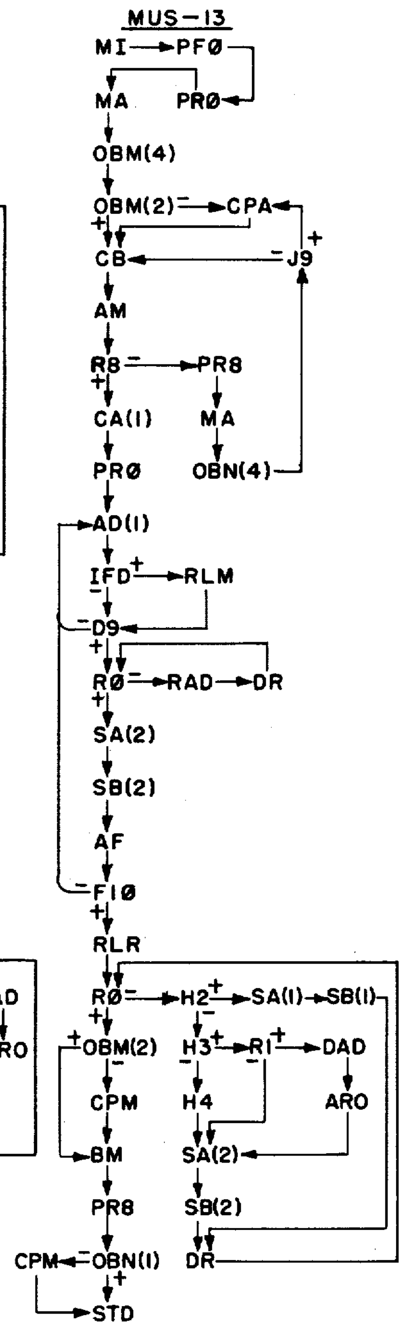


FIG. 107N



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline / Havender

THEIR ATTORNEYS

FIG. 107P

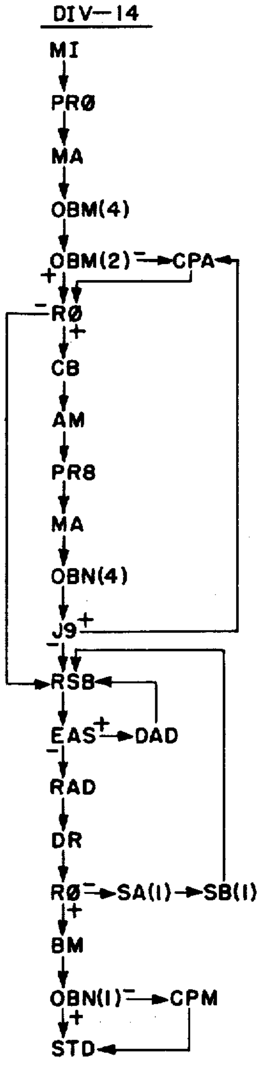


FIG. 107Q

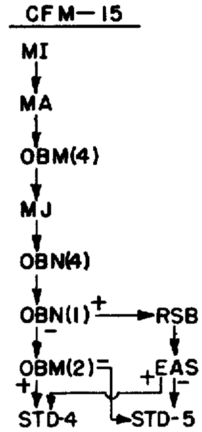


FIG. 107R

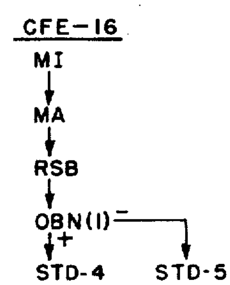
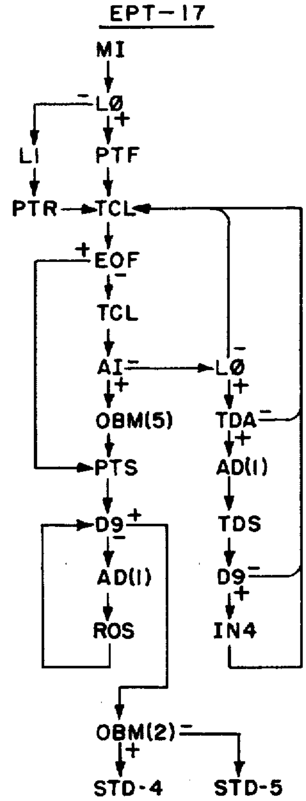


FIG. 107S



INVENTORS—P. B. CLOSE, J. A. COULTER, L. R. DITMER, R. P. FAHRENBRUCK, F. R. GOLDAMMER, L. J. GOODRICK, E. V. GULDEN, T. P. HOLLORAN, C. S. JENKINS, L. D. KILHEFFER, K. A. KINKER, J. D. LYONS, N. D. MANOR, R. P. MARVIN, W. L. MILLER, P. L. O'BRIAN, J. H. RANDALL, E. M. GARNER, R. O. SATHER, R. L. YOST & A. B. BRADEN, JR.

BY Louis A. Kline / Slavender
THEIR ATTORNEYS

3,112,394
ELECTRONIC COMPUTING MACHINE

Patrick B. Close and James A. Coulter, Dayton, Lowell R. Ditmer, Laura, Roy P. Fahrenbruck, Madeira, Freeland R. Goldammer and Le Roy J. Goodrick, Dayton, Ernest V. Gulden, Centerville, and Thomas P. Holloran, Dayton, Ohio, Charles S. Jenkins, Lexington, Ky., Lawrence D. Kilheffer, Karl A. Kinker, and James D. Lyons, Dayton, Neil D. Manor, Xenia, Richard P. Marvin, William L. Miller, Paul L. O'Brian, and James H. Randall, Dayton, and Edward M. Garner, Centerville, Ohio, Roy O. Sather, Dearborn, Mich., and Robert L. Yost and Arthur B. Braden, Jr., Dayton, Ohio, assignors to The National Cash Register Company, Dayton, Ohio, a corporation of Maryland

Filed Dec. 15, 1959, Ser. No. 859,598
10 Claims. (Cl. 235—61.9)

The present invention relates generally to electronic computing devices and more particularly relates to a unique, low cost, general purpose type of electronic digital computing device.

With the ever-increasing enormity and complexity of record keeping and accounting systems in the business world of today, there is accordingly an ever-increasing demand, by necessity, for the adaptation and utilization of high-speed mechanization techniques compatible with such systems. Even though the installation of large scale data processors and electronic computing equipment has partially satisfied the needs of the larger business establishments, unfortunately they are not economically within the realm of practical usage by the smaller business establishments. Consequently, the long-felt need by smaller businesses is yet to be satisfied.

Therefore, one of the principal objects of the present invention is to devise a new and improved low-cost electronic digital computer, which is of the general purpose type in that it possesses substantially unlimited arithmetic and programming capabilities and yet is characterized by extreme simplicity of operation and compatibility with present-day record keeping and accounting systems.

Another object of the present invention is to devise a unique electronic computer having the characteristics aforesaid and which utilizes a commercially available accounting machine as one of its several input-output means in a manner such that the printing format is controlled almost entirely by the accounting machine and thereby minimizing memory space necessary for printing format-control.

Still another object of the present invention is to devise a unique electronic computer having the characteristics aforesaid and which utilizes a serial-digit data-transfer technique and an addressable, random access, coincident-current magnetic core memory synchronized to its several input-output devices in a novel manner to alleviate the necessity of extensive buffering to effect communication with the devices.

A further object of the present invention is to devise an ambiguous-word type computer having the characteristics aforesaid and which is capable of utilizing, in an extremely simple manner, an address incrementing and decrementing format for an internally stored program whereby a single instruction of the program is capable of initiating a sequence of like operations on a multitude of word significations.

Another object of the present invention is to devise such a new and improved low-cost computer which is adapted to utilize input media in the form of data-carrying ledger cards, each having recorded thereon a combination of historical, current, and fixed data which is not only human-readable, but additionally is machine-readable and readily adaptable for computational purposes.

Still another object of the present invention is to devise a unique computer of the above-mentioned characteristics

which not only utilizes the data carried by the ledger card media and other input media in addition to key-board entry data to perform all calculations involved in a computing operation, but additionally, as an incident to the computing operation, updates the ledger card data in both its printed human-readable and its magnetically-recorded machine-readable sections and is thereby capable of producing a printed journal sheet and other output documents, if desired.

A further object of the present invention is to devise a novel low-cost computer which is capable of suppressing all insignificant digits while updating the ledger card data in both its printed human-readable and magnetically-recorded machine-readable sections.

And another object of the present invention is to devise a new and improved low-cost transistorized computer which does not require air conditioning and which utilizes inexpensive resistor logic and transistor flipflop control circuitry which perform multiple functions and thus maintain the cost of the computer at a minimum.

The features of the present invention which are believed to be novel are set forth with particularity in appended claims. The organization and manner of operation of the invention, together with further objects and advantages thereof, may best be understood by reference to the following description taken in connection with the accompanying drawings, in the several figures of which like reference characters identify like elements, and in which:

FIG. 1 is a perspective view of the novel computer constructed in accordance with the present invention;

FIG. 2 is a plan view of the keyboard of the accounting machine portion of the computer;

FIGS. 3A and 3B, when joined together at the dashed lines, form a longitudinal cross-sectional view of the accounting machine portion of the computer;

FIG. 4 is a plan view, partly in section, of the mechanism utilized in accordance with the present invention for stopping the amount racks of the accounting machine portion;

FIG. 5A is a plan view showing the conductor pattern disposed on the top surface of the printed-circuit board utilized for determining the stopped positions of the amount racks;

FIG. 5B is a plan view showing the conductor pattern disposed on the bottom surface of the printed-circuit board utilized for determining the stopped positions of the amount racks;

FIG. 5B is a plan view showing the conductor pattern disposed on the bottom surface of the printed-circuit board shown in FIG. 5A.

FIG. 6 is an exploded view of one of the various solenoid actuated mechanisms utilized for selectively stopping an amount rack;

FIG. 7 is a fragmentary view of the photoelectric means utilized for indicating the instantaneous point-by-point digital positions of the various amount racks;

FIG. 8 is a plan view of the timing rack comb utilized by the photoelectric means of FIG. 7;

FIG. 9 is a plan view of the solenoid-actuated switch basket means utilized for determining the stopped positions of the various amount racks;

FIG. 10 is a partial end view of the mechanism shown in FIG. 9;

FIG. 11 is a partial front view of the mechanism shown in FIG. 9;

FIG. 12 is a fragmentary cross-sectional view of the basket portion of FIG. 9;

FIG. 13 is a plan view of the solenoid-actuated rear form-bar readout mechanism utilized by the card and paper tape punching mechanisms for sensing the position of the accounting machine carriage;

FIG. 14 is a front view of the ledger card handling mechanism shown in FIG. 1 with the cover and table portions thereof removed;

FIG. 15 is a partial front view of the solenoid-actuated mechanism shown in FIG. 16 which is utilized for selectively preventing the lower compression rolls from being disengaged from the accounting machine platen;

FIG. 16 is a partly cross-sectional view taken along lines **16—16** of FIG. 14;

FIG. 17 is a partly cross-sectional view showing the relationship between the ledger card handling equipment and the accounting machine carriage;

FIG. 18 is a partial view of FIG. 17 showing the relationship between the compression rollers, the card-handling platen and the accounting machine platen when the carriage of the accounting machine is open;

FIG. 19 is a partial view of FIG. 17 showing the compression rollers latched against the card-handling platen, and, additionally showing the relative position of the magnetic recording and reproducing heads;

FIG. 20 is a partial view of FIG. 17 showing the construction of the slotted side-plate portion of the accounting machine carriage;

FIG. 21 is a fragmentary cross-sectional view of a portion of the card-handling platen driving mechanism shown in FIG. 14;

FIG. 22 is a plan view, partly in section, of a portion of the card-handling mechanism shown in FIG. 1;

FIG. 23 is a fragmentary cross-sectional view of the ledger card guide assembly shown in FIG. 22;

FIG. 24 is a partial plan view of the rear rack driving means shown in FIG. 17 which is utilized for effecting linear translation of the accounting machine carriage;

FIG. 25 is a fragmentary view, partly in section, of FIG. 24 showing the switching means utilized for detecting home position of the accounting machine carriage;

FIG. 26 is a fragmentary view, partly in section, of the means utilized for preventing a typewriter key on the accounting machine keyboard from being depressed while the accounting machine is carrying-out a cycle of operation;

FIG. 27 is a fragmentary view of the means utilized for manually locking the typewriter carriage-return keys;

FIG. 28A illustrates the rotatable position of the commutator switching means, initially shown in FIG. 14, when the compression rollers are latched with respect to the ledger card drive platen;

FIG. 28B illustrates the rotatable position of the commutator switching means when the compression rollers are unlatched from the ledger card drive platen;

FIG. 29 is a fragmentary cross-sectional view longitudinally taken through the ledger card handling mechanism;

FIG. 30 is a fragmentary cross-sectional view of the solenoid-actuated means shown in FIG. 29 which is utilized for selectively locking the accounting machine carriage in home position;

FIG. 31 is a fragmentary view of the card-drive platen-braking means shown in FIG. 15;

FIG. 32 is a fragmentary view of the solenoid means utilized for automatically unlatching selected low-order ones of the order-hooks of the accounting machine from their respective type-sectors;

FIG. 33 is a side view, partly in section, of one of the magnetic recording and reproducing heads utilized by the computer in accordance with the present invention;

FIG. 34 is a partial side view, partly in section, of the solenoid-actuated type-sector latching and unlatching mechanism;

FIG. 35 is a partial plan view of the solenoid-actuated type-sector latching and unlatching mechanism shown in FIG. 34;

FIG. 36 is a fragmentary cross-sectional view of the composite magnetic recording and reproducing head assembly utilized by the computer;

FIG. 37 is a fragmentary view, partly in section, of a portion of the linkage of the overdraft control mechanism of the accounting machine;

FIG. 38 is a side elevational view of a portion of the cycle-initiating motor-bar mechanism of the accounting machine and the solenoid-actuated means utilized for automatically effecting selective operation thereof;

FIG. 39 is a fragmentary view of the solenoid-actuated means which is utilized to selectively prevent an early release of a depressed amount key during an addition or subtraction cycle of operation of the accounting machine;

FIG. 40 is a fragmentary view of the portion of the accounting machine mechanism which is utilized in initiating a total-taking cycle of operation thereof;

FIG. 41 is a fragmentary view of the reverse tabulation mechanism of the accounting machine;

FIG. 42 is a fragmentary view of the switching means shown in FIG. 41 which is utilized in detecting whether or not the accounting machine is in the process of carrying out a cycle of operation thereof;

FIG. 43 is a fragmentary view of the accounting machine forward-tabulating lever and the switching means associated therewith utilized in detecting whether or not the accounting machine carriage is tabulating over a carriage stop;

FIG. 44A is a fragmentary view of the totalizer reverse key and the mechanism associated therewith;

FIGS. 44B through 44G are fragmentary views of various cam actuated switching means which are utilized to effect selective energization of the paper tape punching mechanism shown in FIG. 44H;

FIG. 44H is a partial side elevational view of the paper tape punching mechanism utilized by the computer;

FIGS. 45A and 45B, when joined together at the dashed lines, form a simplified block diagram of the computer circuitry;

FIG. 45C is a simplified block diagram of the adder-subtracter portion of the computer circuitry;

FIG. 45D is a simplified block diagram of the paper tape punch control circuitry;

FIGS. 46 through 51 disclose the schematic circuit diagrams of all of the various building blocks utilized in constructing the circuitry portion of the computer;

FIGS. 52A and 52B, when joined at the dashed lines, form a schematic circuit diagram of the coincident-current magnetic core memory;

FIG. 53 is a portion of the logical diagram of the adder-subtracter circuitry;

FIG. 54 is the remaining portion of the logical diagram of the adder-subtracter circuitry;

FIG. 55 is a logical diagram of section **1** of the instruction register;

FIG. 56 is a logical diagram of the decoder portion of section **1** of the instruction register;

FIG. 57 is a logical diagram of section **2** of the instruction register;

FIG. 58 is a logical diagram of sections **3** and **4** of the instruction register;

FIG. 59 is a logical diagram of section **5** of the instruction register and the compare circuitry relating to sections **3** and **4** of the instruction register;

FIG. 60 is a portion of the logical diagram of the read-write, digit-cycle, word-cycle and subcommand initiating circuitry;

FIG. 61 is an additional portion of the logical diagram of the read-write, digit-cycle, word-cycle and subcommand initiating circuitry;

FIG. 62 is a logical diagram of the bit-counter;

FIG. 63 is a logical diagram of the digit-counter;

FIG. 64 is a logical diagram of the high-order section of the word-selecting register;

FIG. 65 is a logical diagram of the low-order section of the word-selecting register and a logical diagram of the Y-drivers;

FIG. 66 is a logical diagram of the X and Y grounders;

FIG. 67A is a logical diagram of the memory sense amplifiers;

FIG. 67B is a logical diagram of the rack-readout circuitry;

FIG. 68 is a logical diagram of the R-counter;

FIG. 69 is a logical diagram of a portion of the digit-cycle initiating circuitry;

FIG. 70 is a logical diagram of the "K" digit-register and associated circuitry;

FIG. 71 is a logical diagram of the "J" digit-register and associated circuitry;

FIG. 72 is a logical diagram of the F-counter and compare circuitry;

FIG. 73 is a logical diagram of the instruction register transfer circuitry;

FIGS. 74 thru 79 are logical diagrams of the various word-cycle, time-delay and subcommand initiating flip-flops;

FIG. 80 is a logical diagram of the rack-stopping solenoids and flip-flops;

FIG. 81 is a logical diagram of various solenoid control and light indicating circuitry;

FIG. 82 is a portion of the logical diagram of the ledger card handling circuitry;

FIG. 83 is an additional portion of the logical diagram of the ledger card handling circuitry;

FIG. 84 is a logical diagram of the various timing clock and push-button control circuitry;

FIG. 85 is a logical diagram of the magnetic recording reproducing and control circuitry;

FIG. 86 is a schematic diagram of a portion of the tape punch control circuitry;

FIG. 87A is a timing chart illustrating the instantaneous logical states of the various control lines utilized during a memory read-write cycle of operation;

FIG. 87B is a timing chart illustrating the instantaneous logical states of the various synchronizing clock lines;

FIG. 87C is a timing chart illustrating the instantaneous conditions and states of energization of the various switching and solenoid means utilized by the tape punch mechanism;

FIG. 87D is a timing chart illustrating the instantaneous logical states of the various control lines utilized during a magnetic recording and reproduction cycle of operation;

FIGS. 88A and 88B are diagrammatic illustrations of a portion of the magnetic ledger card utilized by the computer;

FIG. 88C is a diagrammatic illustration of a portion of the punched paper tape utilized by the computer;

FIGS. 88D and 88E together form a diagrammatic illustration of a portion of the journal sheet and pay-check-statement combination properly placed in the accounting machine carriage, ready for a posting operation thereon;

FIG. 88F is an illustration of a pay check;

FIG. 88G is an illustration of a statement form;

FIGS. 89A and 89B logically illustrate the various portions of the circuitry utilized by the computer in carrying out an enter-keyboard-words (EKW) instruction;

FIGS. 90A, 90B and 90C logically illustrate the various portions of the circuitry utilized by the computer in carrying out a print-out-words (POW) instruction;

FIGS. 91A, 91B and 91C logically illustrate the various portions of the circuitry utilized by the computer in carrying out an enter-card-words (ECW) instruction;

FIGS. 92A, 92B and 92C logically illustrate the various portions of the circuitry utilized by the computer in carrying out a record-on-card (ROC) instruction;

FIGS. 93A and 93B logically illustrate the various portions of the circuitry utilized by the computer in carrying out a shift (SHF) instruction;

FIG. 94 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a clear-memory-addresses (CMA) instruction;

FIG. 95 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a motor-bar (MB) instruction;

FIG. 96 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a stop (STP) instruction;

FIG. 97 logically illustrates the various portions of the circuitry utilized by the computer in carrying out an add (ADD) instruction;

FIG. 98 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a subtract (SUB) instruction;

FIG. 99 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a sum (SUM) instruction;

FIG. 100 logically illustrates the various portions of the circuitry utilized by the computer in carrying out an add-pairs-of-numbers (APN) instruction;

FIG. 101 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a multiply-dollar-decimal (MDD) instruction;

FIGS. 102A, 102B and 102C logically illustrate the various portions of the circuitry utilized by the computer in carrying out a multiply-and-shift (MUS) instruction;

FIGS. 103A and 103B logically illustrate the various portions of the circuitry utilized by the computer in carrying out a divide (DIV) instruction;

FIG. 104 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a take-alternate-instruction (CFM) instruction;

FIG. 105 logically illustrates the various portions of the circuitry utilized by the computer in carrying out a take-alternate-instruction (CFE) instruction;

FIGS. 106A and 106B logically illustrate the various portions of the circuitry utilized by the computeer in carrying out an enter-punched-tape (EPT) instruction;

FIG. 107A is a flow diagram of the sequence of operations executed in carrying out an EKW instruction;

FIG. 107B is a flow diagram of the sequence of operations executed in carrying out a POW instruction;

FIG. 107C is a flow diagram of the sequence of operations executed in carrying out an ECW instruction;

FIG. 107D is a flow diagram of the sequence of operations executed in carrying out an ROC instruction;

FIG. 107E is a flow diagram of the sequence of operations executed in carrying out an SHF instruction;

FIG. 107F is a flow diagram of the sequence of operations executed in carrying out a CMA instruction;

FIG. 107G is a flow diagram of the sequence of operations executed in carrying out an MB instruction;

FIG. 107H is a flow diagram of the sequence of operations executed in carrying out an STP instruction;

FIG. 107I is a flow diagram of the sequence of operations executed in carrying out an ADD instruction;

FIG. 107J is a flow diagram of the sequence of operations executed in carrying out an SUB instruction;

FIG. 107K is a flow diagram of the sequence of operations executed in carrying out an SUM instruction;

FIG. 107L is a flow diagram of the sequence of operations executed in carrying out an APN instruction;

FIG. 107M is a flow diagram of the sequence of operations executed in carrying out an MDD instruction;

FIG. 107N is a flow diagram of the sequence of operations executed in carrying out an MUS instruction;

FIG. 107P is a flow diagram of the sequence of operations executed in carrying out a DIV instruction;

FIG. 107Q is a flow diagram of the sequence of operations executed in carrying out a CFM instruction;

FIG. 107R is a flow diagram of the sequence of operations executed in carrying out a CFE instruction; and

FIG. 107S is a flow diagram of the sequence of operations executed in carrying out an EPT instruction.

Inasmuch as the following description of a physical embodiment of the present invention is of considerable length and complexity, and is essentially divided into a multiplicity of separate sections, the various section headings are herein serially numbered and listed below in order to facilitate immediate reference to the various portions of the specification.

1. *General Description of Computer*

With reference to FIG. 1 of the drawings, the computer constructed in accordance with the present invention is completely transistorized and is adapted to be contained in an ordinary metal office desk, indicated generally as **12**. Mounted in the center of the desk is one of the input-output devices for the computer. This particular input-output device is in the form of an accounting machine, indicated generally as **13**, which is of the same general type as that shown and described in United States Letters Patent 2,626,749, issued January 27, 1953, to R. A. Christian et al. The electronic computing equipment is contained in both the left-hand and the right-hand sections of the desk while the two bottom drawers of the right-hand section thereof have been retained. One of the drawers is especially adapted for storage of additional "form-bars" for the accounting machine while the remaining drawer is adapted for storage of magnetic "ledger cards" which, together with the form bars, are utilized to program the computer in a manner as will hereinafter be more fully described. The bridge, located in back of the accounting machine and connecting the right-hand and left-hand sections thereof, provides the necessary support for the electrical power supplies for the computer, and which are energized from a normal 110-volt power source. Mounted on the right-hand side of the accounting machine and disposed in front of the carriage portion thereof, is a card handling mechanism, indicated generally as **14**, which is utilized in reading and recording magnetic information on the ledger cards. And, disposed on the front of the right-hand section of the desk, is a control panel **15** which supports a portion of the various manually operable electrical controls and indicating lights for the computer.

2. *Accounting Machine Portion of Computer (General)*

As the accounting machine portion of the computer is of the well-known type substantially as that shown and described in the aforesaid Patent 2,626,749, a detailed description of each of the multitude of variously interconnected mechanisms contained therein is not deemed necessary in order to obtain a full and complete understanding and appreciation of the present invention. However, in order to insure such a full and complete understanding, there will hereinafter be given a brief description of the various keyboard controls, as well as the salient features and functions of the major component parts thereof, along with the minor modifications and additions thereto which have been found necessary in order to adapt the accounting machine as an input-output device for the computer.

3. *Electric Typewriter Portion of Accounting Machine*

With reference to FIG. 2 of the drawings, the accounting machine contains an electric typewriter in addition to mechanisms for adding and subtracting amounts, and, by means of its form bar, independently functions in many ways as a general purpose computer. All the mechanisms contained therein are aimed at producing three results, namely: the ability to select, accumulate and print totals; the ability to print amounts and dates which are indexed in the key-board; and the ability to type descriptive information on the various forms and ledger cards used by the machine. The electric typewriter is an independent power-driven unit having a keyboard located directly in front of the "amount" keyboard. "ON" and "OFF" switches located on the left side of the keyboard, control the supply of operating power to the machine, and, a red light indicator **17**, located on the front of the machine near the right side, indicates whether or not the electric power is "ON."

Three selective carriage-return keys "RETURN-1," "RETURN-2" and "RETURN-3" are located on the right side of the typewriter keyboard to effect return of the carriage to a predetermined stop position, with or without vertical spacing of the platen. The position at which the carriage returns, depends upon the length of inserts provided in each of the carriage "stops," illustrated as reference numeral **73** in FIG. 3A. Depression of each of the typewriter carriage RETURN keys, raises a "return stop lever" to a given height so that it contacts the desired "carriage return insert" in the carriage stop, to halt the travel of the carriage. Vertical spacing of the platen, under control of the carriage RETURN keys, is effected by paper feed slides **18**, **19**, and **20**, each located at the side of a different one of the RETURN keys. When one of slides **18** thru **20** is moved toward the front of the machine, it couples the corresponding RETURN key with vertical space key "VERT" in a manner such that depression of key "VERT" causes the platen to be spaced vertically in addition to being returned to a predetermined position. However, the paper feed key "VERT" may be used to vertically space the platen one or more positions for each depression thereof. Located directly below the carriage return keys is a typewriter tabulation key "TAB" which, when depressed, causes the carriage to tabulate to a particular column as selected by a corresponding typewriter stop. It is to be appreciated that all carriage motion to hereinafter be described is dependent upon the fact that the carriage is closed.

At the left of the amount keyboard is a typewriter ribbon control lever **21**. With lever **21** in the center or normal position, the typewriter prints in black; when in the lower position, the typewriter prints in red until the lever is manually restored to normal; and, when in the upper position, the typewriter prints in red until completion of the next operation of the accounting machine, at which time the lever is automatically restored to its central position. Another ribbon control lever, not shown, is located on the top of the machine under the platen. When this lever is in its forward position, normal black printing of amounts is effected except during a subtract operation in the #1 totalizer and during a credit balance operation which are automatically printed-out in red. When in its rearward position, typewriter ribbon control lever **21** is rendered effective to control the amount printing operation, the same as on typewriter printing.

4. *Amount Keyboard of Accounting Machine*

The amount keyboard consists of ten rows of keys for entering amounts into the accounting machine totalizers and/or into the computer memory. Each amount row consists of nine keys numbered from "1" thru "9," zeros and punctuation being printed automatically. Any depressed amount key is restored to its original position when another amount key in the same row is depressed, and, all depressed amount keys are simultaneously restored upon depression of the release key "REL" which is located immediately to the right of the "BALANCE-1" key in the lower left corner of the keyboard. This release key is utilized to restore those keys depressed in error, and especially those keys located in the control section on the left side of the keyboard, which are not released by another key in the same row. Any depressed amount key, or keys, is automatically restored upon completion of each machine operation, except when "repeat entry" is called for, either by a stop control or by depression of repeat key "REP" which is located immediately to the right of the number "5" key in the control keyboard section. Those keys located on the left side of the keyboard, such as date keys 1 thru 9 and 1 thru 3 respectively located in the eleventh and twelfth rows, are "staydown" keys which are not released during each machine operation but are released only by depression of another key in that particular row, or by depression of a special release key "REL," located above the number "3" in the twelfth row. A non-repeat date key "NON REP" is located directly above the special release key "REL" and, when depressed, effects non-printing of the year date when depressed and is restored only by depression of the date release key.

Depression of tabulating key "TAB," located in the lower right-hand corner of the amount keyboard, causes the carriage to be tabulated to the left until the movement thereof is arrested by a settable stop on the carriage. The "TAB" key provides a convenient means for tabulating the carriage to a desired columnar position without the necessity of using a "motor-bar" operation, but is normally not used as a part of a posting sequence. A carriage-opening key "CARR. OPEN" is located at the extreme upper right-hand corner of the keyboard and, upon depression thereof, selectively initiates opening and closing operations of the carriage when the carriage is located in "home" position. A single depression of the carriage-opening key causes the carriage to open if it is closed, or to close if it is open. The key does not remain down when depressed, but instead is restored to its original position after each depression.

A carriage-return key "CAR. RET" is located immediately below the carriage opening key and, when depressed, effects return of the carriage to a given stop position which is determined by selective placement of a suitable carriage-return insert in the particular stop that controls the position to be selected if the carriage is not open. The carriage-return key is "locked-out" while the machine is operating and, conversely, machine operation is locked-out if the carriage-return key is depressed and held down.

A carriage-release key "CARR REL," located on the right side of the keyboard directly above upper-motor-bar **23**, functions in three different ways: if fully depressed and held down while the carriage driving means is de-energized, the carriage is unlocked and thus permitted to be manually moved in either direction to any desired position; when partially depressed, the carriage escapement mechanism is released to permit the carriage to be translated from right-to-left by the carriage driving means; and when fully depressed and held down while the carriage-drive is energized, the carriage is translated from left-to-right. The carriage-release key thus permits manual or driven movement of the carriage to any desired position without interruption by forward or reverse-tabulating stops. The carriage-release key is self restoring and functions only while depressed, and when allowed to resume its initial position, effects engagement of the escapement mechanism to hold the carriage at that position.

Non-automatic key "NON-AUTO," located at the right of upper-motor-bar **23**, is a stay-down key which, when depressed, remains down until released by a non-automatic release key "REL," located directly below it. The non-automatic key disables the automatic "machine-release" mechanism to prevent automatic machine operations called for by stop-control. When the non-automatic key is depressed, the control keys at the left side of the keyboard must be operated manually in order to initiate selection of a particular totalizer. Disabling of the automatic machine-release mechanism alters the normal posting procedure in order to allow amounts to be recorded in a column in which entries are infrequently made during a posting operation. The non-automatic release key "REL," in addition to releasing the "NON-AUTO" key, when depressed and held down, unlocks item-counter reset knobs **24**, **25** and **26** so that the corresponding item counters 1 through 3 may be reset to zero, if desired.

Three motor-bars **23**, **27** and **28**, located on the right of the keyboard, one above the other, are utilized primarily to release the machine for operation. By selective use of the three motor-bars, along with motor-bar control lever **29**, the carriage is controlled for normal tabulation, skip tabulation, return tabulation and paper feed, in addition to the disabling of the carriage opening mechanism. There are two methods of motor-bar depression, each causing different results in carriage functions: the first is a normal "touch" depression which requires less than ten ounces of pressure for a distance of one-eighth of an inch, after which a greater resistance to further travel is met; and the second is a "hold" depression which requires more than two pounds of pressure for a total distance of seven thirty-seconds of an inch during more than one-half of the cycle of the machine. As the various carriage functions are also controlled automatically from carriage stops, it would be necessary to study in detail all of these functions and controls to obtain a complete understanding of just what takes place under a different motor-bar condition as modified by control lever 29. As such a detailed understanding is not deemed necessary for a full and complete understanding of the present invention, only a few of the major motor-bar operations will be briefly discussed. However, a more detailed description thereof may be had by reference to previously-mentioned Patent 2,626,749.

A touch depression of upper-motor bar 23 disables carriage return but does not disable stop-controlled skip tabulation or paper feed. Even though paper feed is not provided by the stop over which a touch operation occurs, the upper-motor bar causes the platen to space according to the setting of the platen-spacing control lever.

A hold depression of upper-motor-bar 23 disables all tabulation and carriage return. The upper-motor-bar is designed primarily to permit certain variations in carriage travel from the right or left, depending upon the requirements of the applications. For example, the upper-motor-bar disables carriage tabulation and thus permits vertical posting, and also disables stop-controlled carriage return and thus provides additional flexibility to meet the requirements of the application.

A touch or hold depression of middle-motor-bar 27 provides varying results according to the position of motor-bar control lever 29 and the stops which control paper feed, tabulation, and carriage return. With the motor-bar control lever in the "TAB" position as shown, a touch depression of the middle-motor-bar causes normal tabulation. However, any stop-controlled tabulation takes precedence over a touch depression function. A hold depression takes precedence over carriage tabulation and thus causes the carriage to return to the nearest carriage-return insert. If a hold depression takes place while the carriage is over a stop containing a skip-control insert, the carriage then returns to a carriage-return insert and then tabulates to another predetermined stop depending upon the established skip-control. With motor-bar control lever 29 in a central position midway between "TAB" and "VERT," a touch depression of the middle-motor-bar disables paper feed, in addition to normal carriage tabulation and return. On a hold depression, normal carriage tabulation, stop-controlled non-tabulation, and carriage return are all disabled. However, in this instance, paper feed and stop-controlled carriage skip-tabulation function as usual. With the motor-bar control lever in its lowermost position "VERT," on both a touch and hold depression of the middle-motor-bar, normal carriage tabulation and return are disabled. Stop-controlled skip-tabulation and paper feed are not disabled on either a touch or a hold depression.

The middle-motor-bar thus provides variable control of carriage travel and platen spacing in order to meet the requirements of the application. In addition to operating the machine, it initiates carriage travel to the next predetermined posting position. For such posting procedures as "Accounts Receivable," "Accounts Payable," "General Ledger," etc., the touch and hold depression operations of the middle-motor-bar provide the ability to make multiple postings where the carriage must alternately tabulate and return between two posting columns or between the reference column and the posting column.

A touch or hold depression of lower-motor-bar 28 takes precedence over and thus disables paper feed, carriage return and certain types of stop-controlled carriage tabulation, regardless of the position of the motor-bar control lever. This disabling ability permits the lower-motor-bar to effect a "skip-tabulate" machine operation simply by a touch or hold operation thereof. It also provides the means for skipping columns in which no postings are to be made and to skip operations which are not required on every posting run; it permits carriage tabulation when a stop has established non-tabulation control; and it takes precedence over many functions of the stop, thus providing variable control over paper feed and carriage tabulation and return.

As previously stated, motor-bar control lever 29 affects the function of the middle-motor-bar only. When the lever is in its upper position as shown, the middle-motor-bar functions in a normal manner, whereas, when in its middle and lower positions, normal operation of only the middle-motor-bar is modified.

The control portion of the accounting machine keyboard is located to the left of the amount section and includes the usual totalizer selecting and control keys. For example, "ADD 1" and "SUBT 1" keys respectively select totalizer #1 and conditions the totalizer to receive an additive or subtractive entry. The "BALANCE 1" and "SUB-BAL 1" keys respectively select totalizer #1 and cause it to be operated in a "total" or "sub-total" operation. Located in the same row as keys "BALANCE 1" and "SUB-BAL 1" are four keys numbered 2 thru 5 which select and condition the correspondingly-numbered totalizer to receive additive entries. However, when any one of the totalizer-selecting keys is depressed simultaneously with "SUBT 2–5" key, the related totalizer is selected and conditioned to receive subtractive entries. Similarly, when any one of the totalizer-selecting keys is simultaneously depressed with "TOTAL 2–5" key, the related totalizer is selected and conditioned to be operated in a total taking operation. However, when the "SUB-TOT" key is operated in addition to the totalizer-selecting and the "TOTAL 2–5" keys, the selected totalizer is conditioned to operate in a sub-total operation.

Similarly, located in the same row as keys "ADD 1" and "SUBT 1" are four keys designated 6 thru 9, which, when operated, select and condition a correspondingly-numbered totalizer to receive additive entries. When any one of these four keys is simultaneously used with "SUBT 6–9" key, the selected totalizer is conditioned to receive subtractive entries. However, when one of these totalizer-selecting keys is simultaneously used with the "TOTAL 6–9" and the "SUB-TOT" keys, the selected totalizer is conditioned to operate in either a total or a sub-total operation in the same manner as described above for total and sub-total operations of totalizers #2 thru #5.

Only one of the totalizer-selecting keys in a particular row can be depressed at one time even though two totalizers may be selected simultaneously by depressing a desired totalizer-selecting key in each of the two control rows. However, if a selecting key is depressed in each of the two rows, only the numerical symbol having the highest number designation is printed for each row, each of the selecting keys being restored to its undepressed position at the end of each machine operation. Each of the totalizer-selecting keys may also be manually restored by depressing release key "REL" located to the right of the first row. Thereafter, the selecting keys will not cause a repeat of the machine operation even though repeat key "REP" (located to the right of the totalizer #5 selecting key) is depressed or even though repeat printing of amounts is called for by a stop-control.

If the stop over which the machine is operating has subtract, sub-total, or total-taking control over the same totalizer as does the corresponding selecting key, the stop-control takes precedence. However, in such instances, if non-select key "NON SELECT" (located in the upper left corner of the control keyboard) is depressed, the totalizer-selecting key takes precedence over the stop control.

It should also be noted that it is possible to utilize the keyboard control keys in various combinations to enable amounts to be transferred from one totalizer to another. For example, a total may be taken from any selected totalizer of the group containing totalizers #2 thru #5 or the group containing the totalizers #6 thru #9 and be transferred either additively or subtractively into any one of the totalizers of the opposite group. It is also possible, through the use of the totalizer #1 control keys, to enter such totals either additively or subtractively into totalizer #1. Similarly, a total taken from totalizer #1 may be entered either additively or subtractively into any one of totalizers #2 thru #9 by depression of the proper control keys.

Thus, it is seen that the totalizer-selecting keys are used whenever it is desired to manually select a total or totals for addition, subtraction, sub-totaling, or totaling when they are not selected and controlled by a suitable carriage-stop insert. Totalizer-selecting keys may also be used at random to produce totals for analysis or distribution purposes. For example, sales analysis figures and/or accounting data is produced as a by-product of "Billing" and "Accounts Receivable" posting operations by using totalizer-selecting keys to distribute amounts into totalizers other than those used in the regular posting sequence. In the same manner, major account control totals are established as a by-product of an "Accounts Payable" posting operation.

If the total being cleared is negative, credit balance key "CR. BAL" must also be depressed in order to cause the printing in red of the true credit balance followed by a "CR" symbol. A non-select key "NON-SELECT" is located in the upper left corner of the keyboard and, upon depression thereof, disables all selection and control of the totalizers from a carriage stop but has no effect on printer or carriage functions. The non-select key also permits the machine to be operated when the carriage is not positioned over a stop. Thus, when the non-select key is depressed, the machine is permitted to be operated while either over or off a stop while maintaining complete manual control over all totalizers.

A reverse key "REV." is located immediately below the non-select key and, upon depression thereof, causes a reversal of all addition and subtraction control from a carriage stop. The reverse key, however, has no effect on total or sub-total operations from a stop-control, or any normal function of the control keys. For example, if the reverse key is depressed when a "total #1, add #2, and then subtract #6" operation is called for by a carriage stop, the operation functions normally by first clearing totalizer #1. Thereafter, the amount cleared from totalizer #1 is subtracted from totalizer #2 and simultaneously added to the contents of totalizer #6. If any of the totalizer-selecting keys are depressed during the same operation, the functions normally produced thereby respectively proceed to cause an addition operation in their corresponding totalizers, or a subtraction operation if the subtract control keys are depressed. Simultaneous depression of the reverse key and the credit balance key permits the "X" totalizer to be cleared while the machine is operating over a carriage stop, with subtract control of the "X" totalizer being maintained while being changed to add control.

The reverse key effects printing of a lateral diamond symbol to the left of the amount, which symbol takes precedence over all other symbols with the exception of the "CR" symbol. The reverse key may also be used in a variety of instances such as: correcting errors made while positing; posting "returns" while invoicing; posting "debits" and "credits" in the same column; securing an old-balance-pick-up of credit balances if the carriage is operated while positioned over an "add #1" stop; and securing a second old-balance-pick-up of credit balances if the carriage is being operated while positioned over a "subtract #1" stop.

A non-symbol key, "NON SYM," is located in the extreme left-hand row directly opposite totalizer-selecting key #9 and functions to disable certain symbol printing.

The non-symbol key disables all symbol printing controlled by a carriage stop, except the "CR" symbol on credit balance and sub-credit balance operations. However, the non-symbol key has no effect on the printing of symbols which are initiated by the keyboard control keys on manually-controlled operations. The non-symbol key is locked-down for continuous non-print of symbols by latch **33** which is rendered effective when moved upward as viewed. However, when latch **33** is in its ineffective position, as shown, the non-symbol key is restored at the end of each machine operation. The non-symbol key may also be manually restored by the release key, reverse key or non-select key, and, consequently, may be used to non-print symbols when it is necessary to confine machine printing to a more narrow column than that required for the printing of an amount followed by a corresponding symbol.

Nine overdraft indicators, indicated generally as **34** and numbered 1 through 9, are located directly above the control section of the keyboard and correspond to the correspondingly-numbered totalizer, with the exception of the "X" totalizer. The overdraft indicators are utilized as visual "flag" signals which indicate a minus balance occurring in a corresponding totalizer. For example, when the contents of the totalizers are positive, the corresponding indicators display a gray color, however, when a minus condition occurs in a related totalizer, the correspondingly-numbered indicator displays a white color. However, if a totalizer is cleared after the contents thereof was previously negative, or, if various positive amounts were added back into the totalizer initially containing a negative amount, thus causing it to contain a net positive amount, the corresponding overdraft indicator again displays a gray color. Consequently, the overdraft indicators provide a means of determining when a particular totalizer contains a negative amount so that the credit balance key may be depressed by the operator to control the printing of the true credit balance. In some applications, such as the automatic machine release being disabled by a particular stop to provide optional procedures, the overdraft indicators enable the operator to determine the existence of a credit balance and in which totalizer it exists.

In accordance with the present invention, the accounting machine just described is slightly modified to facilitate its adaptability as an input-output device for the computer. Accordingly, manual operation of motor-bars **23**, **27** and **28** are prevented by a cover **35** disposed thereover and affixed to the keyboard so that selective operation of each of the motor-bars is controlled solely by the computer. However, a "resume-program" bar *rpb* is located just below the three normal motor-bars and is adapted for manual operation by the operator to initiate a computing operation, all of which is to be more fully described hereinafter.

Keyboard decimal-point lights Pφ, **P1** and **P2** are located at the lower section of the amount keyboard between rows **2** and **3**, **5** and **6** and **8** and **9**, respectively, and provide a visual indication as to the decimal-point location on the keyboard for each of the keyboard entries. Operation indicating lights "EK," "EC" and "HA," located on the left front of the machine just below the control section of the keyboard, respectively give visual instructions to the operator as to "Enter-Keyboard-Word," "Enter-Card-Word" and "Halt" instructions, whose functions, again, will be core fully described hereinafter.

5. *Construction and Mode of Operation of Accounting Machine Portion of Computer*

With reference to FIGURES 3A and 3B, there is shown a greatly simplified cross-sectional view of the accounting machine which is essentially the same as that shown in detail in FIGURES 3A and 3B of said aforementioned Patent No. 2,626,749 and fully described therein. As a repetition herein of such massive detailed description and illustrations is not deemed essential for a full and complete understanding of the present invention, but a portion of its salient mechanical features are illustrated in FIG. 3 for the sake of simplicity. However, enough of its mechanical structure is illustrated herein and a brief description thereof is given in order that the various modifications of the machine made in accordance with the present invention may be fully understood.

Inasmuch as each denominational row of amount keys is identical with the other, a description of the #9 denominational row, shown in FIG. 3A, is believed to suffice for the remaining ones. As shown, each of the nine amount keys of a denominational row is slidably supported for vertical movement by top-plate **40**. Each key supports a square stud **41** which is normally disposed on the side of the downwardly projecting stem portion thereof. Located near the center of each key stem is a stud **42** projecting therefrom and which cooperates with a diagonal-walled cam slot **43** formed in control or latching slide **44**. Latching slide **44** is supported for lateral movement by rollers **45** and is spring-urged to the right, as viewed, into engagement with stud **42** by suitable spring means not shown. Thus, when one of the amount keys is depressed, slide **44** is forced to the left by stud **42** and into engagement with zero-stop pawl **46**. As a result of the engagement, pawl **46** is rotated in a counter-clockwise direction about it pivot **47**, thereby releasing block **48** integrally connected to differential actuator or rack **49**. With block **48** released, actuator **49** is freed to move to the right toward the rear of the machine until the corresponding shoulder **50**, formed thereon, comes into engagement with stud **41** of the depressed key, thus stopping the rack at a lateral position commensurate with the digital value of the key depressed. For example, when the #8 key is depressed, the square stud on the lower end of the stem thereof engages the lowermost shoulder of rack **49** and thereby stops the rack in its #8 digital position, the square stud on the #7 key engages the next higher shoulder, and so on.

The #9 key, however, does not function exactly as do the remaining keys. When the #9 key is depressed, slide **44** is forced to the left, pawl **46** is rotated counter-clockwise, block **48** is released, and rack **49** is permitted to travel the entire distance to the right, during the machine operation, until stopped by projecting bar **51**. This extreme position of the rack corresponds to its #9 digital position. If none of the keys of a particular row are depressed, the corresponding rack is not permitted to move and thus remains in its #0 digital position.

All of the racks are guided at the rear ends thereof for straight-line movement during their travel by means of a slotted bar, not shown, which is supported between the side-frames of the machine. Each of the racks associated with a particular amount row of the keyboard is connected by link **54** to an auxiliary rack **55** which is also guided for straight-line sliding movement. It is thus evident that the exact linear translational movement of amount rack **49** is immediately transmitted to its corresponding auxiliary rack **55**. With respect to the racks associated with the date and symbol key-banks, no auxiliary rack is provided inasmuch as these banks are used merely for printing purposes and consequently do not have any corresponding totalizer wheels associated therewith. Each of the amount racks is provided with a slot **56** formed on the lower side thereof, for receiving stud **57** mounted on reducer-arm segment **58**, which, in turn, is rotatably mounted on reducer arm shaft **59** journaled at either end in the side-frames of the machine. The rightmost lower edge of reducer-arm segment **58** is spring-urged by spring **64** into engagement with leading-frame bar **60** which is supported between a pair of identical leading-frame arms **61**, only one of which is shown. Arms **61** are fixedly secured to either ends of reducer-arm shaft **59** and are located just inside the side-frames of the machine. Each of arms **61** is provided with a roller **62** on the lower end thereof which are adapted to engage with bifurcations provided on a cam follower arm, not shown.

When the main cam shaft of the machine is rotated through one revolution by motor **63** mounted on the lower side of the machine base **65**, the cam follower arms are first rocked in one direction and then are rocked in the opposite direction so as to effect movement of leading frame **61** first toward the rear of the machine and then back to its original position as shown. As each of reducer-arm segments **58** is spring-urged into engagement with leading-frame bar **60**, by means of pin and slot connection **56—57** segment **58** is capable of translating its related amount rack first to the right and then to the left, as viewed. Hence, all of the amount racks are simultaneously urged by spring **64** toward the rear of the machine during the first half of the machine cycle until each is stopped by its respectively depressed key. Thereafter, all of the amount racks are positively restored to their home position, as shown, by means of leading-frame bar **60** coming into direct engagement with arm **58** during the second half of the cycle.

The resulting differential positioning of each amount rack is transmitted to its corresponding type sector **67**, by means of link **68** pivotally connected between type sector **67** and segment **58** to rotatably position the type-sector to correspond to the digital position of the amount rack, type-sector **67** being pivotally connected to the upper end of printer arm **69** rotatably mounted on printer shaft **70**. Printer arm **69** is normally locked in the position shown by printer release trigger **66**, which, when rocked counter-clockwise, releases the printer and permits it to be spring-urged clockwise by spring **75**, thereby driving type-sector **67** against the record material placed around platen **71**. After type-sector **67** has been substantially positioned during the first half of the machine cycle and leading frame bar **60** has completel its rearward movement, aligner bar **72** is rocked into engagement with the aligning notches formed on the lower end of reducer arm segment **58** so as to precisely align all of the printing sectors in their differentially-set positions. After the printing operation is completed, aligner bar **72** is disengaged from the reducer arm aligning notches thus permitting the racks and type-sectors to return to home position.

In some types of accounting operations, it is desirable that a "zero" is not to be printed when an amount key in a particular amount row is not depressed. Thus, as more clearly shown in FIG. 34, a "zero-elimination" order-hook **76** is pivotally mounted on rod **77** supported in printer framework **52** and is provided with a tooth which cooperates with a corresponding tooth on control plate **78** so as to retain plate **78**, arm **69**, and hence sector **67** against printing movement even though trigger **66** is rocked counter-clockwise as previously described. There are other times, however, when it is desired that the "zero" be printed when an amount key is not depressed. This is accomplished as follows: As previously described, during the first half of a machine operation the amount racks are moved rearwardly, thereby causing reducer arm segment **58** to be moved out of its home position. This rearward movement causes stud **53**, mounted on an extension of segment **58**, to cam down the tail of order-hook **76** and thus move the tooth thereof out of engagement with the corresponding tooth on plate **78** and thus permits type-sector **67** to make a printing stroke.

With reference to FIG. 35, there are shown sixteen zero-elimination order-hooks **76** disposed on shaft **77** in a side-by-side relationship with respect to one another, each being associated with a particular type-sector to control the operation thereof as previously described. The first and second ones of order-hooks **76**, counting from right to left as viewed, respectively control operation of the first and second ones of type-sectors **67** to allow the printing of symbol designations only. The third order-hook controls operation of the third type-sector which normally prints the first-order or "penny" decimal digit of the amount word having a value from "0" through "9." The fourth order-hook controls operation of the fourth type-sector which normally prints the second-order or "dime" decimal digit of the amount word with the decimal digit "0" thru "9" being preceded by a decimal point. Thus, by selective control of the third and fourth type-sectors, printing of amounts from .00 to .99 cents are effected. The eighth order-hook controls the printing operation of the eighth type-sector which normally prints the sixth-order or "thousand dollar" decimal digit of the amount word, with each decimal digit "0" thru "9" preceded by a comma. Likewise, the eleventh order-hook controls printing operation of the eleventh type-sector which normally prints the ninth-order or "million dollar" decimal digit of the amount word, with each decimal digit thereof preceded by a comma.

Therefore, it is evident that for those zero digits which are to be suppressed, the corresponding order-hook is permitted to lock its corresponding type-sector against a printing movement. Accordingly, for those zero digits which are to be printed, the corresponding order-hooks are rocked out of engagement with their respective type-sectors to permits those sectors to complete their printing operation. Interlocking studs are provided on the upper ends of order-hooks **76** and cooperate with one another in a conventional manner so that a disengaged order-hook effects the simultaneous disengagement of all lower-order order-hooks with respect thereto.

The just described accounting machine is provided with ten conventional add-subtract totalizers, each of which has a capacity of 99,999,999.99. Only six of the ten totalizers are diagrammatically illustrated in FIGS. 3A and 3B and are numbered **1** through **5**, and one designated as the "X" totalizer which is used in overdraft operations in a manner to be later described herein. Each of the totalizers includes one gear wheel for each of the denominational orders of the number, with each gear wheel having twenty teeth formed along the periphery thereof. All of the gear wheels of the totalizers are located in a co-planar relationship with respect to amount racks **49** and auxiliary racks **55** and are adapted to be selectively engaged with and disengaged from the rack teeth in order to permit keyboard entries to be transferred therein and to perform the various calculating functions of the machine.

Located beneath the control keys of the keyboard are a plurality of totalizer control slides, not shown, which selectively effect the engagement and disengagement of totalizers #2 through #9 with respect to their respective amount racks. The type of operation to be performed in any of the totalizers, such as add, subtract, total, or non-total, is determined by the distance through which its associated totalizer control slide is permitted to move. The movements of these slides are controlled either by carriage stops, indicated generally as **73** in FIG. 3A, which are mounted on the traveling carriage, indicated generally as **74** in FIG. 3B, or by the control keys located on the left-hand side of the keyboard, as previously described. As a complete understanding of the totalizer control apparatus, necessary to carry out the above-mentioned cycles of operation, is not believed necessary for a full and complete understanding of the present invention, a detailed description thereof is not herein given. However, such a detailed description may be found in the before referred-to Patent 2,626,749, and, consequently, will not be again repeated herein.

In order to provide the computer with the capability of either "sensing," and thereby determining the respective longitudinal positions of the various amount racks, or to selectively arrest movement of the racks to position each in a predetermined one of its ten positions indicative of numerals "0" through "9," a "read-in read-out" device has been devised and attached to the rear of the accounting machine. With additional reference to the plan view of FIGURE 4, the device consists essentially of a pair of side plates **80** vertically disposed on either side and to the front thereof. Side plates **80** are connected together by means of a horizontally disposed mounting-bar **81**, affixed by suitable means (not shown) to plate-attaching blocks **79** attached to plates **80**. Plates **80** are additionally connected together by rack-solenoid cross-bar **82**, affixed thereto by means of screws **83** threaded in either end of cross-bar **82**. Mounting-bar **81** is also fixedly secured to the base **65** of the machine (FIG. 3B) by means of screws threaded therein, whereas, cross-bar **82** is additionally fixedly secured to header-bar members **84** of the machine by strap braces **85**, respectively affixed thereto by bolts **86** and **87** threaded therein. A second pair of side plates **88** are each secured to one of side plates **80** by screws **89** threaded in plates **80**, and are additionally secured to one another by means of horizontally disposed switch-bars **90** affixed normally thereto by means of screws **91** threaded in either end of bars **90**.

A printed-switch support member **94** (FIG. 3B), affixed to bars **90** suitable means not shown, is provided with a pair of vertical projections **95** located on either end thereof, and onto which is secured a horizontally disposed plunger-support member **96** by means of screws **97** (FIG. 4) threaded in projections **95**. As shown in FIG. 4, a plurality of identical rack extensions **98** are individually connected by means of pins **99** and spring clips **100** to a different one of the ten auxiliary racks **55** which, when viewed from right to left, respectively represent the low-order digit "0" through the high-order digit "9" of the amount word entered in the keyboard of the accounting machine as previously described. As shown in FIG. 3B, on the uppermost edge of each of rack extensions **98** are formed ten "saw-tooth" shaped rack teeth, indicated generally as **101**, each of which, from right to left as viewed, respectively represents one of the ten digital amount positions "0" thru "9" of a different one of the amount racks.

In order to obtain "straight-line" travel of the amount racks during movement thereof, each of rack extensions **98** has riveted thereto a switch plunger **102** which is in sliding contact with plunger support member **96** and is guided between guide washers **103** which, in turn, are rotatably supported by screws **104** threaded in support member **96**. Vertical movements of the amount racks are prevented by means of a horizontally disposed cross-bar **92**, which is affixed to side plates **80** by screws **93** threaded in either end of the bar, cross-bar **92** being fitted in a rectangular slot **108** formed in each of the amount racks. Rack "wobble" is prevented by means of a double-edged comb **109** affixed to the front face of cross-bar **92** by screws **110** threaded therein, comb **109** being inserted in slot **108** of each of the racks in such a manner that each rack is effectively slideably fitted in the space between oppositely disposed teeth of the comb, thus preventing any bending movement of the racks during travel thereof.

When all of the amount racks are traveling from home position toward the rear of the machine in a "setting" direction, it is necessary that all travel in unison with neither lagging nor leading the other. Consequently, a "helper" spring **105** is connected between each of rack extensions **98** and spring bar **106** attached at either end thereof to side plates **88** by suitable means not shown. Each of the springs possesses a "spring-constant" characteristic sufficient to maintain all of reducer arm segments **58** (FIG. 3A) in contact with their respective leading-frame bar **60** during the cycle of operation of the machine. Thus, during the rack-setting operation, all racks simultaneously move in exact alignment during their setting operation.

For reasons to become more apparent hereinafter, it is desired that the various digital positions at which the amount racks have been positioned, be easily determinable by the computer. Consequently, wiper arm assembly, indicated generally as **107**, is affixed to and carried by the lower side of each of rack extensions **98**. The assembly comprises an angle-bracket **111** welded to rack extension **98**, and which supports a laminated body of electrically non-conductive wafers **112** having sandwiched therebetween a pair of flexible, electrically conductive and oppositely disposed wiper blades **113** and **114** which are fixedly secured at one end by means of suitable bolts **115**. The opposite ends of wiper blades **113** and **114** are in sliding contact with the upper and lower faces, respectively, of a printed circuitry type of switch board **116** which is secured to support member **94** by means of screws **117** threaded in blocks **90**.

With reference to FIGS. 5A and 5B, printed-circuit board **116** consists essentially of an electrically non-conductive support having a plurality of electrical conductors formed on the opposite surfaces thereof by electrodeposition, etching, or other well-known techniques. FIG. 5B is a plan view of the conductor arrangement disposed on the bottom face of board **116**, which arrangement comprises ten conductors **119**, each positioned parallel to the movement of the amount racks, and thus wiper blades **114**, and terminating on the lower side of the board at terminals Dϕ through D9, respectively corresponding to a successively higher-order amount rack representation. FIG. 5A is a plan view of the conductor arrangement on the upper face of board **116**, which arrangement comprises ten conductors **118** positioned perpendicular to the movement of the amount racks, and thus wiper blades **113**, and terminating on the upper side of the board at terminals **120***a* thru **120***j*, respectively corresponding to the ten digital positions "0" thru "9" of the amount racks. As conductors **119** are arranged parallel to the movement of the amount racks, wiper blades **114** are, at all times, in electrical contact with corresponding ones of conductors **119**, whereas, wiper blades **113** are, at all times, in electrical contact with one of conductors **118**, depending upon, of course, the distance their respective racks have moved in order to get to their present digital location. For example, if the amount rack corresponding to the first-order digit is still at home position, the lowermost one of wipers **113** remain positioned on the first vertical conductor located on the left of board **116**, as viewed in FIG. 5A. However, if the amount rack has moved and is now located in its number "1" digital position, wiper blade **113** is then positioned on the conductor located second from the left, and so on, to the tenth vertical conductor upon which wiper blade **113** is positioned when the amount rack is located in its number "9" digital position.

Therefore, as wiper blades **113** and **114** are electrically connected together, in order to "interrogate" the first-order amount rack as to its digital position, for example, it is necessary only to supply a pulse of electrical energy at terminal Dϕ on the lower side of board **116**. Consequently, the interrogation pulse appears at one of terminals **120***a* thru **120***j* on the upper side of board **116** depending upon, and corresponding to, the digital position of the low-order rack. For example, if the first-order rack is in the number "5" digital position, the interrogation pulse appears at terminal **120***f* on the upper side of board **116**. It is now evident that in order to interrogate all of the amount racks as to their respective digital positions, it is necessary only to sequentially supply an electrical impulse to terminals Dϕ thru D9, and immediately thereafter a sequence of output pulses appear at output terminals **120***a* thru **120***j*, respectively corresponding to the digital position of each of the ten amount racks. That is, if the first pulse output is at output terminal **120***f*, if the first-order amount rack is located in the number "5" digital position; if the second output pulse appears at output terminal **120***b*, the second-order amount is located in the number "1" digital position, and so on, until the last or tenth output pulse appears at one of the output terminals indicative of the digital position of the tenth-order amount rack.

In order that the computer is capable of controlling the printing operation effected by the accounting machine, it is necessary that suitable means be provided whereby any selected one, or all, of the amount racks of the machine are capable of being stopped at predetermined digital positions under the control of the computer. With reference to FIG 3B, in addition to the "exploded" view of FIG. 6, such a computer-controlled rack-stopping mechanism consists essentially of a plate **125** attached to the front face of cross-bar **82** by screws **126** threaded therein and essentially extending the entire distance between side plates **80**.

It is to be appreciated that essentially there are a total of ten individual rack-stopping mechanisms, each one being identical to the one shown in FIG. 6 which corresponds to and controls positioning of the tenth-order amount rack.

An L-shaped bracket **127** is secured to plate **125** by means of screws **129** threaded in bracket **127** and a rack-stopping solenoid RA9L (corresponding to the tenth-order rack) is affixed to bracket **127** by suitable means not shown. A flat solenoid armature **130** has riveted thereto an L-shaped spring member **128**, the horizontally disposed leg portion of spring member **128** being affixed to bracket **127** by means of screws **132** threaded therein, and washer **131** being interposed between the heads of screws **132** and spring member **128**. The face of the vertical leg of armature **130** is disposed in close proximity to core **133** of solenoid RA9L and is magnetically attracted thereto each time the solenoid is energized by electrical current being passed through input leads **134** and **135** thereof. Also riveted to the vertical face of armature **130** is an horizontally disposed lever arm **136** having a pair of downwardly depending arms **137** supporting a cylindrical tempered-steel rack-stopping pin **138**.

Therefore, it is evident that each time any one of the ten rack-stopping solenoids is selectively energized, the vertical leg of armature **130**, corresponding to that particular rack, is attracted toward solenoid core **133**, the left end of lever arm **136** is deflected downwardly, and pin **138** is inserted in front of the tooth on rack extension **98** corresponding to the digital position that particular rack is desired to be stopped. Thus, by selectively energizing all ten of rack-stopping solenoids RAϕL–RA9L (FIG. 4) from computer controlled means at particular times during their movement in the setting direction, each of the amount racks is stopped and thus positioned at any digital position from "0" to "9" and each is then effective to control printing of the numerical digit corresponding to the digital position of that particular amount rack as previously described. With the exception of the rightmost one of teeth **101**, all of the teeth on rack extension **98** are equally spaced by a distance of approximately five thirty-seconds of an inch. However, in order to provide a suitable clearance of approximately twenty-five thousandths of an inch between rack-stopping pin **138** and the rightmost or "0" digit tooth during insertion of the pin, the spacing between the "0" digit tooth and the adjacent "1" digit tooth is five thirty-seconds less twenty-five thousandths, or approximately .132 inch.

A laminated spring buffer mechanism (FIG. 3B) comprising top, center, and bottom L-shaped leaves **142**, **143**, and **144**, respectively, is affixed to the upper surface of cross-bar **82** by means of screws **145** threaded therein. The lower end of each of the depending leg portions of leaves **142** and **143** has a vertical slot, not shown, centrally formed therein to accommodate lever arm **136** of the rack-stopping mechanism together with rack extension **98**, both of which are slidably positioned therein. The width of the slot in leaves **142** and **143** is slightly larger than the greatest thickness of arm **136** or rack **98**, but is much smaller than the width projection of depending arms **137**.

Additionally, the buffer mechanism is so positioned that each end of rack-stopping pin **138** is in close proximity to the front surface of leaf **142**. As a result, leaves **142** and **143** provide a support for the backside of pin **138** to absorb the force of the blow exerted by the rack teeth on pin **138** when the rack is suddenly stopped thereby. However, it has been found that when the rack is traveling at maximum velocity in the rearward setting direction and is brought to a sudden stop by pin **138** being inserted in front of one of teeth **101**, the lower ends of leaves **142** and **143** are deflected to the right as viewed in FIG. 3B. When the rack is suddenly brought to a standstill, the depending legs of leaves **142** and **143**, due to the energy stored therein, attempt to resume their original position. As a result, they tend to overshoot their mark by deflecting to the left and thus attempt to drive the rack in the opposite direction. To minimize this undesirable effect, a screw **146** is slidably mounted in aligned holes formed in leaves **142** and **143** and is threaded in leaf **144**. A pressure setting spring **147** is interposed between the head of screw **146** and washer **148** positioned against the front face of leaf **142**. Thus, by suitable adjustment of screw **146**, a suitable friction damping effect is provided the rack-stopping mechanism and rebounding of the racks is minimized.

As previously described, during a rack setting operation, all non-stopped amount racks travel in unison and in alignment with the other. Consequently, control means have been provided whereby the computer not only detects the corresponding digital positions of all stopped racks, but, in addition, detects the point-by-point digital position of all non-stopped racks. With reference to FIG. 4, an eleventh or timing rack **149** is provided having a construction and mode of operation exactly the same as the just-described amount and auxiliary racks **49** and **55**, respectively, with the exception that timing rack **149** is not stopped at any time by a depressed key on the accounting machine keyboard and is not provided with a solenoid-actuated rack-stopping mechanism. Actually, a conventional amount and auxiliary rack are used for this purpose and the row of keys, located on the accounting machine keyboard and which selectively stop that particular rack when depressed, along with the zero stops **46** (FIG. 3A) associated therewith, have been removed. Timing rack **149** is provided with an extension **150** attached thereto by pin **99** and spring clip **100**, and is essentially of the same general construction as rack extensions **98** with the exception that rack extension **150** does not have any teeth formed thereon.

With reference to FIG. 7, together with the partial plan view thereof in FIG. 8, a flat and horizontally disposed, timing rack comb **151** is welded to the uppermost face of rack extension **150** and has formed therein eight rectangular shaped slots **152** parallel disposed with respect to one another with their long axes perpendicularly disposed with respect to the direction of movement of rack extension **150**. An L-shaped bracket **156** is affixed to the upper surface of cross-bar **82** by means of screws **157** threaded therein, and a cubical block of insulating material **158** is attached to the lower front face of the depending leg of bracket **156** by means of screws **159** threaded in block **158**. Block **158** has embedded therein a photocell **160** whose terminals are connected to output terminals **161** and adapted to be energized by leads **162** connected to a suitable source of electrical energy, not shown. A further L-shaped bracket **163** is attached to mounting bar **81** by suitable means, not shown, and has mounted on the horizontal leg portion thereof, a light source **164** which is energized by leads **165** connected to a suitable source of electrical energy, not shown. A horizontally disposed flat plate **166** is fixedly secured to the lower end of block **158** by suitable means, not shown, and is positioned directly below and parallel with respect to timing rack comb **151** intermediate light source **164** and comb **151**. Plate **166** has a single rectangular-shaped slot **167** formed therein disposed substantially parallel with respect to, and of sufficient length so as to partially overlap the slots formed in comb **151**. It is evident, therefore, that as timing rack **150** is moved from left to right, as viewed, the slots formed in comb **151** sequentially coincide with the slot formed in fixed plate **166**, thus sequentially exposing light source **164** to photocell **160**.

For the accomplishment of the desired objectives in accordance with the present invention, it is desirable that the sequential alignment of slots **152** and **167**, from the rightmost to the leftmost of slots **152**, be synchronized with the time at which the non-stopped amount racks simultaneously pass through digital positions number "1" thru "8." For example, at the beginning of a rack-setting operation when all the amount racks are positioned in their number "0" digital positions, the relative positions of slots **152** with respect to slot **167** are as shown in FIG. 8. However, when the non-stopped amount racks are approaching their number "1" digital positions, slot **167** is also approaching alignment with the rightmost one of slots **152**; when all non-stopped amount racks are approaching their number "2" digital positions, slot **167** is approaching alignment with the second from the rightmost one of slots **152**; and so on, until all non-stopped amount racks are approaching their number "8" digital positions, at which time slot **167** is approaching the eighth or leftmost one of slots **152**.

However, as the just-described solenoid-actuated rack-stopping mechanism is not capable of inserting rack-stopping pin **138** (FIG. 6) in front of the desired one of rack teeth **101** in a zero amount of time even after the corresponding solenoid is energized, the computer, hence the solenoid, must be "warned" or notified slightly in advance in order for the rack-stopping mechanism to be effective in stopping its related rack at the desired digital position. For example, it has been empirically determined that, with a rack traveling at an average velocity of approximately twenty inches per second, approximately one-tenth of an inch is sufficient anticipation for the rack-stopping mechanism. Consequently, as the tooth pitch of the amount racks is approximately equal to five thirty-seconds of an inch, slots **167** and **152** are in substantial alignment approximately two-thirds of a digital position ahead of the non-stopped amount racks. However, as the amount racks start with an initial velocity of "zero" and are therefore traveling at a relatively low velocity when passing through digital positions number "1" and number "2," and also as the spacing between the first and the second ones of rack teeth **101** is approximately twenty-five thousandths of an inch less than the pitch of the remaining teeth, as previously mentioned, the distance between the slots formed in comb **151**, at the rightmost end thereof, are much less than the distance between the slots formed in the leftmost end thereof.

Ideally, the slot spacing should approximate the curve-plot of rack velocity vs. time. However, due to various loading conditions of the amount racks occurring during the setting operation thereof, the exact spacing between slots are empirically determined. Such an empirical determination is as follows: a satisfactory distance between slot **167** and the first one of slots **152**, counting from right to left and when in a "0" digit position, is .045 inch; a satisfactory distance between the second and the third ones of slots **152** is .162 inch; and the consecutive distances between the remaining adjacent ones of slots **152** are .164, .153, .157, .153 and .139 inch, respectively, corresponding to slot distances **3—4**, **4—5**, **5—6**, **6—7**, and **7—8**; a slot width of .030 inch has been found satisfactory.

With reference to FIG. 3B, there is shown a mechanical type of memory device which is utilized by the computer to temporarily store indications of the digital positions of each of the amount racks at the end of the previously-initiated rack-setting operation. The device then operates suitable output means in the form of a conventional card or paper tape punching mechanism which makes a permanent record of all of the sequential digital positions at which the amount racks were stopped during a programming or computing operation.

Such a memory device consists essentially of a horizontally disposed upper switch-basket, indicated generally as **170**, and a horizontally disposed lower switch basket located directly below switch-basket **170** and indicated generally as **171**. Upper switch-basket **170** consists essentially of a substantially flat, rectangular-shaped and horizontally disposed board **172** composed of a suitable electrically non-conductive material, such as fiberboard or the like, and affixed to the underside of cross-bars **173** by means of screws **174** threaded therein. Cross-bars **173**, in turn, are affixed to side plates **88** by means of screws **175** which are threaded in the ends of bars **173** as shown more clearly in FIG. 4. Board **172** is provided with ten equally spaced and rectangular shaped slots **176**, the long axis of each slot being in axial alinement with the direction of movement of a different one of the amount racks. A column of ten pairs of equally spaced and metallic spring-clips **177** are fixedly secured to an electrically non-conductive mounting block **178** and collectively form a unitary assembly of the upper switch-basket. As shown more clearly in FIG. 12, the lower end of each pair of spring-clips **177** making-up a column is "snapped in" and held by the same one of the rectangular slots formed in board **172**. With reference back to FIG. 3B, each column of spring-clips is disposed directly above and oriented parallel with respect to the direction of movement of a corresponding amount rack, and the spring-clips within the column are spaced in a manner such that each pair is positioned with respect to another pair to correspond to a different one of the ten digital positions of that particular amount rack. Each of the spring-clips has connected thereto an electrical conductor forming a part of input-output cable **179** which is electrically connected in a manner to be more fully described hereinafter.

Lower switch-basket **171** consists essentially of a substantially flat, rectangular-shaped and horizontally disposed upper shelf-plate **180** and lower shelf-plate **181** held in position by vertically disposed back and front side-plates **182** and **183**, respectively, in a parallel relationship with respect to one another. Shelf plates **180** and **181** are provided with a plurality of apertures individually through which is slidably disposed a lower electrically-conductive spring-clip member **184**. Spring-clip members **184** are arranged in equally-spaced columns and rows and are positioned so that each of the lower spring-clip members is disposed directly below one of upper spring-clip members **177**, as shown more clearly in the fragmentary view of FIG. 12. With reference to FIG. 9, the lower spring-clip members are arranged in ten equally-spaced columns which are designated, from right to left, #0 through #9 and are oriented in axial alignment with respect to the movement of and respectively corresponding to one the ten amount racks previously designated as the first-order digit rack thru the tenth-order digit rack. Each of the ten lower spring-clip members within a column is equally spaced with respect to one another and are positioned in rows, each of which corresponds to one of the ten digital positions of the amount racks. With reference to FIG. 12, two pair of upper and lower spring-clip members **177** and **184**, respectively, are taken from columns #8 and #9 and shown for illustrative purposes only. As shown, each of lower spring-clip members **184** within each column is guided for vertical movement only by means of common shaft **185** riding in an elongated slot **186** which is formed in the lower end of member **184**. Each of spring-clip members **184** within a row are spring-urged downwardly by means of spring **187** which is interlaced between rods **188**. Rods **188** are affixed at their ends to side-plates **182** and **183** (FIG. 9) and pins **189** are carried by spring-clip members **184**. An elongated bale **190** is rotatably mounted on shaft **191** affixed at its ends to side plates **182** and **183** (FIG. 9) and is spring-urged by spring **192** in a clockwise direction, as viewed, into engagement with each of lower spring-clip members **184** which are located in the same column.

Therefore, each time spring-clip member **184** is deflected upwardly, bale **190** is first rocked counter-clockwise by cam surface **169**, and then is rocked clockwise by spring **192** when cam surface **169** passes the end-most projection of bale **190**. Thereafter, bale **190** engages the lower-most portion of cam surface **169** and locks spring-clip member **184** in its upwardly deflected position, as shown by the dotted lines, and thus effects a short-circuit of upper spring-clips **177**. However, if a different one of the lower spring-clip members, located in the same column, is subsequently deflected upwardly, bale **190** is again rocked counter-clockwise and the lower clip member, which was previously held in a locked position, is returned to its lowermost position by means of spring **187**. However, the subsequently deflected lower clip member is locked in its upwardly deflected position by subsequent clockwise rotation of bale **190**, as before. It is therefore seen that only one of the lower spring-clip members in each column is permitted to be locked in an upwardly deflected position at any one time due to the fact that a subsequently deflected lower clip member releases all previously deflected lower clip members located in the same column.

With reference to FIGS. 9 and 10, lower switch basket **171** is provided with a shaft **193** on either end thereof which rides in an elongated slot **194** formed in arms **195** and **196**. Arms **195** and **196** are fixedly secured to either end of shaft **197** journaled in side-plates **88**. On the right-hand extension of shaft **197**, as viewed, is fixedly secured arm **198** having a slot formed in the end thereof, the slot being engaged by pin **199** carried by the rotatable armature **200** of an electrically-operated solenoid designated L46 and which is mounted on side-plate **88** by means of bolts **201**. Solenoid L**46** is a conventional commercially-available rotary type which, when energized, the armature thereof is caused to be rotated in a clockwise direction, and, when de-energized, the armature is spring urged in a counter-clockwise direction back to its initial starting position. Mounted on left side-plate **88** by means of bolts **202** is a second electrically-actuated solenoid designated L45 which is identical in construction as previously described solenoid L**46**. Rotatable armature **203** of solenoid L**45** carries a pin **204** which engages an elongated slot formed on the end of arm **195**.

With reference to FIG. 3B, lower switch-basket **171** is spring urged upwardly, as viewed, by suitable springs, not shown, and normally rests in the position as shown. However, when solenoids L45 and L**46** are simultaneously energized, shaft **197** is rocked clockwise and basket **171** is lowered to a position sufficient to engage the lowermost end of one of spring-clip members **184** in each of the ten columns with a corresponding raised cam surface **205** which is formed on the top side of the rightmost end of each of the ten switch plungers **102** which are individually affixed to the end of one of rack extensions **98**. Consequently, one of spring-clip members **184** in each of the ten columns engages cam **205** on its corresponding amount rack, by means of plunger **102**, in a manner such that all of the engaged clip members, one in each column, are latched in an upwardly deflected position as previously described. When solenoids L45 and L46 (FIG. 9) are both de-energized, switch basket **171** is spring urged upwardly and consequently all of the latched clip members **184** are thereby brought into engagement with, and thus short-circuits, corresponding ones of upper spring-clip members **177**. As shown, clip member **184**, corresponding to the number "0" digital position of its particular amount rack, is illustrated as being in engagement with clip member **177** which is located in the number "0" digit position in its particular column. However, had the amount rack been in its number "9" digital position, the rightmost one of clip members 184, that is, the one located in the number "9" digital position, would have been deflected upwardly instead of the lower clip member located in the number "0" digital position as shown.

With reference to FIGS. 9 and 11, a C-shaped bracket **206** is riveted or otherwise secured to either end of mounting board 180 and is provided with a slot 207 into which slidably fits a vertically oriented rod 208. Rod 208 is held fixedly secured by means of set-screw 209 threaded in bracket **210** which, in turn, is fixedly secured to side-plate **88** by means of screw **211** threaded in the end of bracket **210**. Consequently, it is seen that the desired vertical movement of lower basket **171** is insured at all times.

In some instances, operation of the card or paper tape punching mechanism is required only when the accounting machine carriage is in certain columnar positions corresponding to the columnar format of the ledger cards or business forms in the carriage thereof. Consequently, operation of the punching mechanism is additionally controlled by means of a plurality of switches located at the back of the traveling carriage which are actuated by a plurality of adjustable stops removably mounted on a rear form-bar located on the back of the carriage. Such a carriage-position switch mechanism is of the same general type as that shown and described in copending application Serial No. 567,411, filed February 23, 1956, by Edgar H. Sonnanstine, Jr., and assigned to the same assignee as the instant application. Consequently, a detailed description thereof is not deemed necessary to be again given herein in order to obtain a full understanding and appreciation of the present invention.

Briefly, however, with reference to FIG. 17, the mechanism includes a substantially C-shaped rear form-bar **212** removably mounted on suitable backets, not shown, which are secured to the lower rear framework of the carriage, the rear form-bar being essentially of the same general construction as the front form-bar which is similarly attached to the front of the carriage. Rear form-bar **212** is slidably supported between rollers **213** and extends substantially the full length of the carriage. Removably secured to the rearward face of form-bar **212** are a plurality of plates **214** which are disposed along the length thereof in predetermined positions corresponding to the columnar format of the business forms to be inserted into the carriage of the accounting machine.

Each of plates **214** is provided with two parallel and vertically disposed columns of holes formed therein, not shown, one predetermined hole or holes in each column having inserted therein one of switch operating studs **215** and **216**. With reference to the plan view of FIG. 13, stud **215** is adapted to coact with one of a plurality of switch plungers **217** arranged in a left-hand row, whereas, stud **216** is adapted to coact with one of a plurality of switch plungers **218** arranged in a right-hand row. The total number of switch plungers **217** and **218** are equal to the total number of columnar holes formed in plate **214**, and each plunger is disposed directly opposite a different one of the holes, and, consequently, directly opposite any stud inserted therein. As shown, plungers **217** and **218** are slidably mounted in a shiftable framework, which, together with plungers **217** and **218**, constitutes a switch basket similar to previously described switch basket **171** shown in FIGS. 3B and 9. Each of the plungers has an enlarged rearward end, or lower end as viewed, having parallel side surfaces which fits snugly between and is guided for movement by suitable slotted guide rods **219**. The forward, i.e. upper, end of each of the plungers is slotted to snugly embrace a suitably slotted guide rod **220**, guide rods **219** and **220** being fixedly secured at their ends to top plate **221** and a bottom plate, not visible, which are slotted to receive guide rod **222** to form a shiftable framework for supporting plungers **217** and **218**. A substantially V-shaped actuating lever **223** is pivotally mounted on stud **224** and has the end of the upper arm thereof pinned to upper plate **221** and also to the lower plate, not shown, by means of pin **225** which is fixedly secured between the upper and lower plates. The end of the remaining arm of lever **223** is slotted to receive stud **226** carried by the rotatable armature of an electrically-actuated, rotary solenoid **L47** which is of the same general construction as those previously described.

When solenoid L47 is energized, the armature thereof is rotated in a counter-clockwise direction, as viewed, thereby causing stud **226** to rock lever **223** in a clockwise direction to shift the switch basket upwardly until a selected one of each of the upper ends of plungers **217** and **218** respectively engage studs **215** and **216**. Consequently, as a result of such engagement, predetermined ones of plungers **217** in the left-hand row, and predetermined ones of plungers **218** in the right-hand row, are both depressed and latched in their downwardly deflected positions, and, additionally, simultaneously release selected ones of previously depressed and latched plungers. When solenoid L47 is de-energized, the switch basket is spring-urged back to its initial starting position. As a result of the basket return, the electrically non-conductive tip **228**, carried on the lower end of each of the plungers, engage and thus deflect downwardly the topmost ones of a plurality of switch blades **229** which are arranged in twenty rows with each row corresponding to a different one of plungers **217** and **218**. When the uppermost one of switch blades **229** is deflected downwardly, all of the switch blades in that particular column are electrically connected together until solenoid **227** is again energized to move the switch basket upwardly, as viewed, as previously described.

A substantially V-shaped arm **153** is pivotally mounted on stud **154** and supports a roller on the upper end thereof which is adapted to engage a projection of plate **214**, as shown in dotted lines, to rock arm **153** counter-clockwise, as viewed, and effect actuation of a switching mechanism, indicated generally as **SC41**, in a well known manner.

With reference to FIGS. 44B and 44D, cam actuated switching means have been provided to effect selective energization of solenoids L45 and L46 (FIG. 9) and L47 (FIG. 13) at predetermined times and in a predetermined sequence, in addition to effecting selective energization of a suitable paper tape punching mechanism in a manner to be described hereinafter. It is to be noted that FIG. 44D is essentially a fragmentary view of the main cam shaft mechanism of the accounting machine portion of the present computer which is illustrated in FIG. 44 of the previously referred-to Patent 2,626,749 and fully described therein.

Briefly, however, the main cam shaft **588** of the accounting machine extends transversely across the machine beneath base **65** and is journaled at either end in roller bearings **589**, only one of which is shown, mounted in brackets **590** secured to the under side of base **65** by means of screw **591** threaded therein. Even though not illustrated in FIG. 44D, cam shaft **588** has secured thereto a plurality of cams which are utilized for controlling various machine functions in a manner fully described in Patent 2,626,749. In accordance with the present invention, however, a pair of plate cams **592** and **593** are affixed to either side of a split hub **594** by means of screws **595** threaded therein. Hub **594**, in turn, is fixedly secured to shaft **588** by means of screws **596**. As shown more clearly in FIG. 44B, a flat plate **597** is secured to depending projections formed on the underneath side of base **65** by means of screws **598** threaded therein. A mounting frame **599**, in turn, is fixedly secured to the bottom side of plate **597** by means of screws **600** threaded therein and is adapted to have fixedly secured to the depending leg thereof, by suitable means not shown, a plurality of switches SC42, SC43, and SC49.

As diagrammatically illustrated in FIG. 44G, and, as shown by the timing chart of FIG. 87C, for reasons to become more apparent hereinafter each time main cam shaft 588 of the machine is rotated clockwise through an arc of 126 degrees from the starting position as shown in FIG. 46B, cam 592 engages the movable arm of switch SC49 such that the normally-opened contacts thereof are closed thereby; when the main cam shaft is rotated 168 degrees clockwise from the starting position as shown, cam 592 additionally engages the movable arm of switch SC48 such that the normally-opened contacts thereof are closed and the normally-closed contacts thereof are opened thereby; when the main cam shaft is rotated 182 degrees, the movable arm of switch SC49 is released by cam 592 and the normally-opened contacts thereof resume their initial opened condition; a 213 degree clockwise rotation of the main cam shaft causes cam 593 to engage the movable arm of switch SC42 such that the normally-opened contacts thereof are closed thereby; a 224 degree clockwise rotation of the main cam shaft causes the movable arm of switch SC48 to be released by cam 592 such that the normally-opened and the normally-closed contacts of switch SC48 are allowed to resume their initial conditions; and, a 264 degree clockwise rotation of the main cam shaft causes the movable arm of switch SC42 to be released by cam 593 such that the contacts thereof are allowed to resume their initial conditions.

As previously mentioned, due to the incorporation of an overdraft control mechanism in the accounting machine portion of the present computer, true negative totals are permitted to be selectively printed-out from any of the totalizers therein, all of which is fully described in the previously referred-to Patent 2,626,749. With reference to the fragmentary view of FIG. 44F, which essentially corresponds to FIG. 47 of the just referred-to patent, the overdraft control mechanism includes an overdraft cam shaft 601 extending transversely across the machine beneath base 65 and is journaled in brackets 602, only one of which is shown, which are secured to the underneath side of base 65 by means of screws 603 threaded therein. Secured to overdraft shaft 601 is a spur gear 604 which disconnectably engages a similar spur gear, not shown, mounted on main cam shaft 588 (FIG. 44D). As fully described in the just referred-to patent, the pitch diameter of overdraft gear 604 is twice that of the similar gear mounted on the main cam shaft of the machine, thereby causing overdraft cam shaft 601 to be rotated at one-half the speed of main cam shaft 588 when the latter is rotated during a machine cycle.

In accordance with the present invention, a pair of leaf switches, indicated generally as SC45 and SC47, are insulated from and fixedly secured to bracket 605 by means of screws 606 threaded therein, bracket 605 being fixedly secured to bracket 602 by suitable means not shown. Rotatably mounted on a stud 607 (FIGS. 44C and 44E) affixed at one end to bracket 605, are a pair of cam follower arms 608 and 609 which are respectively spring biased in a clockwise direction, as viewed in FIGS. 44C and 44E, by means of springs 610 and 611. Fixedly secured to overdraft cam shaft 601 are a pair of cams 612 and 613 which are properly positioned on shaft 601 so as to engage rollers 614 and 615 respectively carried by cam follower arms 608 and 609. Respectively affixed to the upper end of each of arms 608 and 609 is one of a pair of insulating tips 616 and 617 which are adapted to respectively engage leaf switches SC45 and SC47.

With additional reference to the timing chart of FIG. 87C, when main cam shaft 588 is rotated 30 degrees from the initial starting position thereof, overdraft cam shaft 601 is engaged thereto and is thereafter rotated at one-half the speed of the main cam shaft. Thereafter, each time overdraft shaft 601 is rotated 10 degrees counter-clockwise from its starting position as shown in FIG. 44C, cam follower arm 608 is permitted to be rocked clockwise by spring 610 due to the decreased diameter of cam 612 being presented to roller 614. Thus, when arm 608 is rocked clockwise, the normally-closed contacts of leaf switch SC45 are opened thereby. Each time overdraft shaft 601 is rotated 165 degrees counter-clockwise from its initial starting position, an increased diameter of cam 612 is presented to roller 614, and, consequently, cam follower arm 608 is rocked counter-clockwise and thereby closes the contacts of switch SC45.

With additional reference to FIG. 44E, each time overdraft shaft 601 is rotated 260 degrees counter-clockwise from its starting position as shown, the maximum diameter of cam 613 is presented to roller 615, and, consequently, cam follower arm 609 is deflected counter-clockwise such that the normally-opened contacts of leaf switch SC47 are closed thereby. However, each time the overdraft shaft is rotated 290 degrees counter-clockwise from its starting position, the minimum diameter of cam 613 is presented to roller 615 to allow arm 609 to be rocked clockwise by spring 611, which, in turn, allowed the contacts of switch SC47 to resume their undeflected positions as shown.

6. *Paper Tape Punch Portion of Computer*

It is to be appreciated that substantially any one of the multitude of commercially available paper tape or card recording mechanisms is quite easily adaptable to be utilized by the present computer to provide a permanent record of all of the sequential digital positions at which the amount racks of the accounting machine portion thereof were selectively stopped doing a programming or computing operation. However, FIG. 44H discloses a partially cross-sectional and fragmentary view of a preferred high speed paper tape recording mechanism so utilized by the present computer, which mechanism is fully described in copending application Serial No. 820,539, filed June 15, 1959, by Richard C. Simmerman et al. and assigned to the present assignee, FIG. 44H of the present application being substantially identical to FIG. 1 of the just-mentioned copending application.

It is to be noted at the outset that the recording mechanism partially shown in FIG. 44H essentially relates to that portion of the entire mechanism which is utilized, together with a common feeding means, for punching a single hole in the paper tape during each operation thereof. As it is desired to utilize a transversely oriented column of eight holes, plus one sprocket hole, in a recording tape with respect to the travel thereof, and in accordance with the particular code associated with the punching mechanism, it is to be appreciated that the complete paper tape recording mechanism, in fact, includes a combination of nine substantially identical ones of such punching mechanisms shown in FIG. 44H, plus the single paper tape feeding means.

As shown, side frames 618, only one of which is shown, form the main supporting structure for the recording mechanism, side frames 618 being held in a fixed spaced-apart relationship by means of a plurality of cross-bars including a punch guide block 619, a punched lever comb 620, and a punch interposer guide block 621. A rectangularly-shaped punch support block 622 is fixedly secured to one of side-plates 618, by suitable means not shown, and is provided with a single row of nine bores formed therethrough, each bore being adapted to have slidably disposed therein a punch member 623. Each of the punch members 623 is provided with a recess located midway its length, into which is disposed one end of a punch-restoring lever 624, punch-restoring lever 624 being rotatably supported on a shaft 625 fixedly secured to the side frames 618, and additionally being spring urged in a counter-clockwise direction, as viewed, by means of a spring 626 connected to the leftmost end thereof. Consequently, punch members 623 are spring-urged upwardly by means of corresponding ones of levers 624. A die block 627 is spaced from and parallel disposed below punch support block **622** and is provided with a corresponding row of nine bores formed therethrough, with each of the bores being in axial alignment with respect to a different one of punches **623**, and, the space between support block **622** and die block **627** being of sufficient magnitude to provide a proper clearance for the passage therethrough of paper tape **628**.

A plurality of nine punch levers **629** are each pivotally supported by a different one of punches **623** and are each rotatably mounted on an eccentric **630** formed on a shaft **631**, which, in turn, is journaled in the side frames **618**. A punch lever comb **620** is transversely disposed midway the length of punch levers **629** and is provided with a plurality of rectangularly-shaped slots for accommodating each of the punch levers. The comb **620** also houses a shaft **632**, which constitutes a stop for each of the punch levers **629** which are normally being urged thereagainst by means of the spring bias applied to punches **623**, as previously described.

Suitable power means, not shown, is operatively connected to the shaft **631** so as to continually rotate the eccentric **630** in a clockwise direction during operation of the punching mechanism. This clockwise rotation of the eccentric **630** causes the rightmost end of each of the punch levers **629** to be vertically translated in an elliptical oscillatory path. Disposed adjacent the rightmost end of each of the punch levers **629** is a corresponding plurality of interposers **633**, one for each of the punch levers **629**, slidably supported in a guide comb block **634**, which is fixedly secured to punch interposer guide block **621** by suitable means not shown. Each of the interposers **633** is substantially pinned to the lowermost end of an interposer arm **635**, the uppermost end of each of the interposer arms being rotatably supported on a shaft **636** contained within the guide block **621**. The uppermost end of each of the interposer arms **635** has an extension formed thereon which is adapted to be selectively engaged by armature **637** of a corresponding one of a plurality of electrically actuated solenoids, indicated generally as **638**, which are each supported by bracket **639** attached to side frames **618**.

In order to initiate a punching operation, selected ones of the solenoids **638** are simultaneously energized, so that the respective ones of the armatures **637** are each deflected to the right and thereby engages the projection of the related one of the interposer arms **635**. As a result of the simultaneous engagements, selected ones of the interposers **633** are simultaneously deflected to the left, from the position shown, and thereby present obstructions in the paths of travel of the rightmost ends of corresponding ones of the punch levers **629**. Due to the engagement with interposer **633**, the respective punch lever pivots about that particular interposer, instead of pivoting about punch **623**, as previously described. Thus, corresponding ones of punches **623** are thereby simultaneously deflected downwardly to perforate a digit representation in the tape **628**. Thereafter, the punches are returned to their normal positions due to the urgency of the springs **626** acting through the levers **624**. However, in order to positively insure punches **623** being restored to their normal positions, the leftmost end of each of the punch restoring levers is engaged by the leftmost end of a corresponding one of punch levers **37**, during translation thereof, and is thereby urged counter-clockwise to positively restore its corresponding punch to its normal position.

Located adjacent the front of interposer arms **635** is an interposer restoring cam **640**, which is continually rotated in a clockwise direction in synchronism with the eccentric **630**. The cam **640** is initially rotatably positioned with respect to the eccentric **630** in such a way that at approximately 135 degrees of each revolution of the eccentric **630**, the cam **640** engages the lowermost ends of those interposer arms **635** which have been deflected forwardly by their respective solenoids, and thus simultaneously restores each interposer arm **635** and its related armature **637** and interposer **633** to their normal positions. Located adjacent the rear of the guide comb block **634** is a permanent magnet **641**, which exerts a sufficient magnetic attraction to interposers **633** to normally maintain each in an ineffective position, the attraction, however, being insufficient to interfere with the shifting of the interposer arms **635** by the corresponding armatures of solenoids **638**.

Suitable driving means are provided for intermittently advancing the paper tape **628** by one unit of distance only during that period between approximately 258 to 360 degrees of revolution of the eccentric **630**; i.e., after each digital punching operation is completed. As shown, the paper tape **628** is fed from a supply reel **642** through the punching station and subsequently stored on a takeup reel **643**. Such feeding means comprises a ratchet wheel **644** attached to a feed line **645**, which is rotatably supported by the side frames **618**. Also mounted on feed line **645**, adjacent sprocket wheel **644**, is a second sprocket wheel, not shown, having a series of sprocket teeth formed on the periphery thereof for engaging corresponding sprocket holes punched in the paper tape. Also mounted on the end of feed line **645** is a manually operable knob, not shown, which provides manual control of the feed line.

The mechanism for advancing the ratchet wheel **644**, to provide a paper-feeding operation, includes a pawl **645** pivotally supported on a stud **646** and disposed within a slot, not shown, longitudinally formed in the lowermost end of a shaft **647**. The uppermost end of the shaft **647** is pinned to the lowermost end of a paper feed arm **648** rotatably mounted on the eccentric **630**. The shaft **647** is slidably disposed in support **675** and guide **676** in such a manner that elliptical translational movement of paper feed arm **648** by eccentric **630** is transmitted to shaft **647** to effect vertical oscillating movement thereof. The vertical oscillations of the shaft **647**, of course, oscillate the pawl **645** in a similar manner. The pawl **645** is normally spring urged in a counter-clockwise direction, as shown, into engagement with the teeth of ratchet wheel **644** by means of spring **677** connected to an extension thereof. However, in order to insure positive engagement of pawl **645** with the teeth of ratchet wheel **644**, the rightmost edge of the upper end of pawl **645** is engaged by a cam **678**, in such a manner that, when the cam **678** is rotated, the pawl **645** is thereby positively deflected to the left and into engagement with the ratchet wheel teeth.

At the beginning of a punching operation during which the eccentric **630** is rotated from 0 to 180 degrees, the shaft **647** is deflected downwardly and thereby withdraws the pawl **645** from engagement with the ratchet wheel **644**. As the pawl **645** continues its downward movement, the spring **677** pivots the pawl **645** counter-clockwise and thus positions the pawl beneath the next lower ratchet tooth. On the upstroke of shaft **647**, ratchet wheel **644** is indexed counter-clockwise, and thus the paper tape is advanced by one unit of distance. As just mentioned, pawl **645** is additionally positively deflected into engagement with the ratchet wheel **644** by means of cam **678**, thus assuring a proper feeding operation.

7. Magnetic Ledger Card Handling Portion of Computer

With reference back to FIG. 1, a magnetic ledger card handling mechanism, indicated generally as **14**, is mounted on the rightmost side of the just-described accounting machine portion of the computer, and is adapted to cooperate with the accounting machine to automatically effect insertion of ledger cards in the carriage thereof and to automatically effect ejection of the ledger cards when the carriage is within approximately two inches of its extreme right-hand position, as shown. The card handling mechanism is normally enclosed by a front cover **231**, which supports a smooth surfaced and horizontally disposed table **232**, onto which ledger cards are sequentially placed prior to their insertion into the accounting machine carriage.

With reference to FIG. 14, there is shown a front view of the card handling mechanism with cover **231** and table **232** removed. The mechanism comprises a vertically disposed left side-plate **233**, a vertically disposed right side plate **234**, a vertically disposed back side-plate **235**, a vertically disposed front side-plate **236**, and a diagonally disposed bottom side-plate **237**, all of which are connected together by suitable means, not shown, to form the basic support structure for the mechanism. The leftmost end of bottom side-plate **237** is affixed to the lowermost end of left side-plate **233** by suitable means, not shown, side-plate **233** being affixed to the rightmost bottom side of the accounting machine frame **65** by means of bolts **238** threaded therein. The leftmost side of back side-plate **235** is affixed to the right-hand extension of a rail **240**, which normally supports the rearward portion of the accounting machine carriage, by means of screws **239** threaded in the rail **240**. A platen **241** is journaled in the uppermost ends of side-plates **233** and **234**, and is also journaled in hubs **242** and **243**, which are respectively welded or otherwise affixed to side-plates **233** and **234**. The platen **241** is substantially identical in construction to conventional accounting machine platens and comprises essentially a centrally disposed metallic hub **244** projecting from either ends thereof. The hub **244** supports a metallic tube, not shown, which is provided with a layer of rubber, or the like, facing material **245** concentrically disposed thereabout.

Means are provided to effect the desired automatic rotational movement of the card handling platen **241** in order that the ledger cards, which are previously in contact with the uppermost surface thereof, are driven thereby into operative engagement with the accounting machine platen. Such means essentially comprises an electrically-operated card driving motor (CDM) which is mounted on the innermost surface of side-plate **234** by means of screws **247** threaded in the end bracket of the card drive motor, as shown in the fragmentary view of FIG. 31. The output shaft **248** of motor (CDM) extends through bracket **234** and is rotatably supported at its extended end by mounting bracket **249**, which is fixedly secured in a spaced parallel relationship with respect to bracket **234** by means of posts **250** interposed therebetween. One end of each of the posts **250** is welded or otherwise affixed to bracket **234**, whereas the opposite ends thereof are fixedly secured to plate **249** by means of screws **251** individually threaded therein. Freely mounted on the motor shaft **248** are a pinion gear **252** and a conventional friction clutch mechanism, indicated generally as **253**, which comprises a pair of metallic washers having a fiber washer interposed therebetween and glued or otherwise affixed to the innermost face of one of the metallic washers. A knurled friction-adjustment knob **254** is threaded on the motor shaft **248** intermediate friction clutch **253** and plate **249**. Thus, by rotational adjustment of the knob **254**, a predetermined amount of torque is transmitted from the motor shaft **248** to the pinion gear **252** in a well-known manner. With additional reference to FIG. 31, the pinion gear **252** meshes with an idler gear **255**, which, in turn, meshes with a spur gear **256** affixed to one end of shaft extension **257** by means of set screw **258** threaded in gear **256**. With reference to FIG. 21, the opposite end of the shaft extension **257** is connected to the hub **244** of the platen **241** by means of a set screw **259** in the hub **244**.

Therefore, it is evident that rotational movement of a predetermined torque is imparted to the card drive platen **241** by the card drive motor (CDM) through a friction clutch mechanism and a gear train interposed therebetween. However, for reasons to become more apparent hereinafter, it is desirable that rotational movement of the card drive platen **241** be immediately arrested at predetetermined times. Consequently, a fast-acting electrically-operated card drive braking means has been provided and comprises a braking solenoid, indicated generally by the reference character (CDB), which is attached to the outermost surface of the side plate **234** directly opposite the idler gear **255**, by suitable means not shown. The idler gear **255** is freely mounted on a longitudinally movable armature shaft **263** in such a manner that when solenoid (CDB) is de-energized, the card drive motor (CDM) is permitted to rotate the platen **241** in a normal manner. However, when the solenoid (CDB) is energized, the idler gear **255** is pulled inwardly, or to the left as viewed, and thereby locked against further rotational movement. Due to the normal action of the friction clutch **253**, any and all undesirable effects created by rotor inertia of motor (CDM) are precluded when platen **241** is suddenly brought to a standstill even though the motor is momentarily energized after the braking action occurs.

As shown in FIG. 1, the traveling paper carriage, for the before-described accounting machine, consists of a framework which includes a pair of end housings **264**. With reference to FIG. 17, the end housings **264**, only one of which is shown, are supported in a spaced and parallel relationship with respect to each other by means of an S-shaped cross-bar **265**, an L-shaped cross-bar **266**, and a C-shaped carriage rack rail **267**, to which both of the end housings are secured by suitable means, not shown. The cross-bars **265** and **266** are fixedly secured directly to the end housings **264**, whereas the rack rail **267** is fixedly secured at either end thereof to a bracket **268** by suitable means, not shown, the bracket **268** being bonded to a rubber block **269**, which, in turn, is bonded to a second bracket, **270** secured to end housing **264** by means of screws **271** threaded therein.

The thus-constituted carriage frame is supported for transverse sliding movement on the accounting machine frame by means of a tube **272** secured to the under side of the cross-bar **266** by suitable connecting members **273**. The tube **272** rides in and is horizontally supported by a series of grooved ball-bearing rollers **274**, which are rotatably secured to the back-rail **240**. Vertical and sidewise movement of tube **272** is prevented by means of a plurality of conically-shaped ball-bearing rollers, not shown, which are mounted on either side of and in engagement with the tube **272**. The forward portion of the carriage is supported by means of ball-bearing rollers **275**, which are received within a grooved formed in the cross-bar **267**. The rollers **275**, in addition to being rotatably supported by a vertically disposed frame-plate on the accounting machine, not shown, are also rotatably supported on the front side-plate **236**. The accounting machine carriage is yieldably driven in either direction by suitable fluid driving means (including motor **63**, FIG. 3A), operatively connected to a driving gear **277**, which engages a carriage-drive pinion rack **276** fixedly secured to the cross-bar **265**.

With additional reference to FIG. 20, a substantially V-shaped guide-plate **278** is fixedly secured to the innermost side of each of the end housings **264** and has formed therein an elongated slot **279**, in which rides a pin **280** affixed to an end-plate **281**. A rectangularly-shaped front feed chute **282** extends the full width of the carriage and is secured at either end thereof to both of the end-plates **281**. On the lowermost side of the accounting machine carriage, there is rotatably supported a square shaft **283** (FIG. 17) and a bail **284**, each operatively connected together and operatively connected to the lowermost end of the side-plates **281** in such a manner that when the carriage open key ("CARR. OPEN," FIG. 2) is depressed, rail **284** is deflected counter-clockwise, as viewed. Consequently, arm **285** (FIG. 16), and hence shaft **283**, are both freed to be rocked counter-clockwise and thereby allow the lowermost ends of the side-plates **281** to be deflected downwardly and around the accounting machine platen **71**. When the shaft **283** is rocked counter-clockwise (FIG. 17), lower compression rolls **288**, carried thereby, are moved out of engagement with the platen **71**. Due to the positional locations of the slots **279** in the side-plates **278** and pins **280** riding therein, when the lowermost ends of the side-plates **281** are deflected downwardly and around the platen **71**, the front feed chute **282** is lowered into the position shown in FIG. 18, thereby completing normal opening of the accounting machine carriage in the manner more fully shown and described in the before-referred-to Patent No. 2,626,749. When the accounting machine carriage is open and the card-drive mechanism latched in a driving position, as will later be described, a suitable ledger card is permitted to be driven downwardly to the left of chute **282**, as viewed, between paper guide plate **289** and the bottom side of platen **71**, and into paper guide channel **291**. Thereafter, closing of the carriage engages the compression rolls **288** with the platen **71** to secure the ledger card prior to the unlatching of the card-drive mechanism, as, again, will later be described.

In accordance with the present invention, an elongated housing **292**, extending substantially the full length of the carriage, as shown in FIG. 1, is pivotally mounted at either end thereof to the topmost side of the side-plates **281** by means of a pivot **293**, the housing **292** having affixed at either end thereof brackets **290** shown in FIG. 20. A second pin **294** is affixed to each of the brackets **290** and rides in a second elongated slot **295**, which is formed in each of the side-plates **278** just above the slot **279**, as shown more clearly in FIG. 20. A plurality of compression rolls **296** are rotatably supported by brackets **297** affixed to a shaft **298**, which is operatively supported by the housing **292**. The compression rolls **296** project through suitable openings formed in the front face of the housing **292**, as more clearly seen in FIG. 1.

During the just-described carriage opening operation when the lower end of each of brackets **281** is deflected in a path down and around platen **71**, bracket pin **280** and housing pin **294**, respectively riding in slots **279** and **295**, separate from the other. Therefore, when front chute **282** is in the position shown in FIG. 18, housing **292** is silted by a further amount, and, consequently, compression rolls **296** are positioned in close proximity to card-handling platen **241**, the relative positions of pins **280** and **294** being shown in FIG. 20 in their moved positions by dotted lines. With reference to FIG. 19, the amount of tilt of housing **292** is limited by a pair of arms **299**, each connected at opposite ends of, and projecting through, housing **292** as shown. The depending end of each of arms **299** respectively support a stud **300**. Thus, when housing **292** is in the position shown in FIG. 18, stud **300** engages the large diameter portion of one of a pair of cams **301** which are rotatable about shaft **257** and are disposed at either end of platen **241** as shown in FIG. 14. When cams **301** are simultaneously rotated counter-clockwise, as viewed and by suitable means to be described hereinafter, both of studs **300** are engaged thereby to further tilt housing **292** in a counter-clockwise direction and thereafter latch all of compression rolls **296** firmly against platen **241** as illustrated in FIG. 19. It is therefore obvious that as long as compression rolls **296** are latched against card-drive platen **241**, paper guide **282—292** is prevented from being raised to the position shown in FIG. 20.

Therefore, when compression rolls **296** are latched against card-drive platen **241**, card-drive motor (CDM) (FIG. 14) is energized and platen **241** is thereby rotated in a clockwise direction, as viewed in FIG. 19, the front edge of a ledger card, indicated as reference numeral **305** and normally resting on table **232** as shown in FIG. 22, is automatically driven between the lowermost side of front feed chute **282** and the uppermost sides of paper guides **302** and **303**, thence between the lowermost side of platen **71** and the uppermost side of paper-guide **289** (FIG. 17), and thereafter into chute **291**.

By reversing card drive motor (CDM) and driving the ledger card outwardly to a predetermined position, bail **284** is rotated clockwise thereby positively driving compression rolls **288** into engagement with platen **71** by suitable means not shown. Unlatching of upper compression rollers **296** allows paper-guide **282—292** to raise to the position shown in FIG. 17. Consequently the ledger card is automatically positioned by paper guide strips **302** for a subsequent posting operation.

With reference to FIGS. 14 and 16, means for selectively rotating latching cams **301** comprises an electrically-operated motor (PGM) mounted on vertically disposed bracket **310** by screws **311** threaded in the end housing of the motor, bracket **310**, in turn, being secured to front side-plate **236** by means of screws **312** threaded in bracket **310**. The output shaft of motor (PGM) projects through bracket **310** and has affixed thereto spur gear **313** in mesh with a second spur gear **314**. Gear **314** is fixedly secured to a hollow shaft **315** rotatably mounted on shaft **316** which is journaled in bracket **310** and right side-plate **234**. A second hollow shaft **317** is also rotatably mounted on shaft **316** and has spur gear **318** affixed on the end thereof adjacent side-plate **234**. Gear **318** meshes with the rightmost one of spur gears **319** affixed to either end of shaft **320** which is journaled in side-plates **233** and **234** and disposed parallel with respect to platen **241**. Gears **319** individually mesh with one of spur gears **321** which are rotatably mounted on a projection of each of side-plates **233** and **234**, as illustrated in FIG. 21, with respect to side-plate **234**.

Each of spur gears **321** is fixedly secured to one of cams **301**, by suitable means not shown, each of which is individually rotatably supported by the projection of one of side-plates **233** and **234**. As shown in FIG. 14, a coil spring **324** surrounds oppositely disposed end portions of hollow shafts **315** and **317** and either end thereof is affixed to shafts **315** and **317** by means of screws **322** and **323**, respectively, threaded therein.

Therefore, it is seen that rotational movement of the rotor of latching motor (PGM) is transmitted to both of latching cams **301** by way of gears **313**, **314**, spring **324**, and gears **318**, **319**, and **321**. As latching cams **301** are positively stopped by latching pins **309**, as previously described in connection with FIG. 19, adverse effects of rotor inertia of the latching motor are eliminated by spring **324** even though the motor may momentarily be energized after the latching operation is completed.

With reference to FIG. 16, control means have been provided for selectively energizing the latching motor to effect the previously described latching and unlatching operation of the compression rollers with the card drive platen. The control means consist essentially of an L-shaped bracket **327** fixedly secured at one end to front side-plate **236** by means of screws **328** threaded in bracket **327**. A substantially V-shaped member **329** is pivotally mounted at its center to the opposite end of bracket **327** by means of pin **330** and is spring-urged in a clockwise direction, as viewed, by means of spring **331**. The leftmost leg of member **329** is in contact with lever arm **332** which actuates the plunger of microswitch **333** affixed to the frame of the card handling equipment, by suitable means not shown.

Thus, during a normal carriage opening operation of the accounting machine, as previously described, paper guide housing **292**, in reaching the position shown in FIG. 18, rocks member **329** in a counter-clockwise direction, as viewed. When deflected counter-clockwise, member **329** releases the plunger of switch **333** to thereby permit energization of the paper guide motor (PGM, FIG. 14) and thus initiates counter-clockwise rotation of cams **301** to complete the latching operation as previously described.

Additional commutator type of control means have been provided to effect selective de-energization of the paper guide motor upon completion of the latching operation. With additional reference to FIG. 14, such control means consist essentially of a flat C-shaped electrically non-conductive support member 334 concentrically disposed about hollow shaft 317 and supported by side-plate 234 by means of posts 335 interposed therebetween and respectively connected thereto by suitable means, not shown. Affixed to the inside face of member 334 are four circumferentially disposed flexible metallic brushes 336, N, E, and 339 which are adapted to engage the face surface of commutator 340 concentrically disposed about, and affixed to hollow shaft 317. As shown in FIGS. 28A and 28B, the face of commutator 340 consists of a layer of electrically non-conductive material designated 341 having embedded therein a C-shaped electrically-conductive commutator bar 342.

When paper guide 282—292 is in the position as shown in FIG. 18 and compression rolls 296 are adjacent to, but not latched against, card drive platen 241, brushes 339 and E are electrically connected together by commutator bar 342, as shown in FIG. 28B. However, when paper guide motor PGM is energized, cam 301 (FIG. 19) is rotated counter-clockwise to complete the latching operation as previously described. When the latching operation is completed, communator 340, thence commutator bar 342, will have been rotated counter-clockwise to the position as shown in FIG. 28A, and, as a result, brushes N and 336 are electrically connected together by commutator bar 342 to effect de-energization of the paper guide motor in a manner to be more fully described hereinafter. With reference to FIG. 14, solenoid actuated reversing relays, designated by the reference characters (PGR) and (CDR), are each mounted on bracket 342 affixed to lower end-bracket 237, and are respectively utilized to control the directions of rotation of paper guide motor (PGM) and card drive motor (CDM), as will later be described.

As previously mentioned, in the mechanization of record keeping and accounting procedures, it is desirable to utilize a ledger card having historical, current, and fixed data recorded thereon in both machine readable and human readable languages. With reference to FIG. 108A, there is illustrated a machine-readable ledger card 305 which is adapted to be utilized in accordance with the present invention. Accordingly, ledger card 305 is provided with a vertical strip of ferromagnetic material 306 coated along the left margin of the face surface thereof, the magnetic strip being effectively divided into three magnetically isolated vertical channels which are each adapted for selective recording of information therein.

With reference to FIG. 19, a magnetic read-write assembly, indicated generally as 343, is supported by bracket 344 affixed to left side-plate 233 of the card handling mechanism and is so positioned that the magnetic strip on each ledger card passes over the top thereof each time a card is fed into or out of the accounting machine carriage. A metallic leaf spring 345 is interposed between paper guide 282 and the top of read-write assembly 343 in such a manner that each ledger card is guided thereunder, and, consequently, the magnetic strip on each ledger card is, at all times, lightly pressed against the top face of assembly 343 to insure consistent magnetic recording and reproduction characteristics, and for other well-known reasons.

As shown in FIG. 36, magnetic read-write assembly 343 comprises three encapsulated read-write heads 346, 347, and 348, which are equally-spaced from and disposed parallel with respect to one another. As shown in FIG. 33, each magnetic head comprises two substantially flat and generally C-shaped ferromagnetic pole pieces 349 and 350 having their lower legs abutting and their upper legs separated by a uniform air gap of approximately .005 inch. Each leg is respectively encompassed by one of a pair of serially-connected and inductively-aiding coils 351 and 352, each wound with approximately 600 turns of suitably electrically-insulated wire. The junction of coils 351 and 352 is connected to output terminal 353, whereas the remaining ends thereof are respectively connected to output terminals 354 and 355.

The air-gaps of magnetic read-write heads 346—348 are lengthwise aligned with the other, are properly positioned as shown in FIG. 19 so as to be contiguously disposed with respect to the magnetic strip of the ledger card passing thereover, and are lengthwise oriented substantially perpendicular with respect to the travel of the magnetic strip to effect longitudinal magnetic recording thereon.

In order to insure that the ledger card is properly aligned and fed into the card handling equipment by the operator, various photoelectric sensing means are incorporated into the control circuitry thereof. With reference to FIG. 22, a substantially rectangular-shaped ledger card guide assembly, indicated generally as 360, is affixed to the topmost surface of table 232, by suitable bolts 361 threaded therein, and extends substantially along the leftmost side thereof. As shown more clearly in the cross-sectional view of FIG. 23, guide assembly 360 comprises a pair of flat and rectangular-shaped plate sections 362 and 363, one mounted on top of the other, and a pair of flat and rectangular-shaped half-plate sections 364 and 365 disposed side-by-side with respect to one another and on the underneath side of plate 363; the thickness of plate 365 is less than the thickness of plate 364 by an amount equal to at least the thickness of a ledger card for reasons to become more apparent hereinafter. With reference to FIG. 29, plates 362 and 363 have suitable bores formed therein which form enclosures for housing photocells (PCA1) and (PCA2) whose sensitive areas are oriented downwardly and are properly positioned so as to respectively receive radiant energy from light sources 366 and 367, the radiant energy first passing through slots 368 and 369 formed in table 232.

In operation, photocells (PCA1) and (PCA2) normally respectively receive radiant energy from, and are thereby energized by light sources 366 and 367. However, when the operator of the computer desires to effect proper insertion of a ledger card in the carriage of the accounting machine, in a manner previously described, the ledger card is first placed on the table of the card handling equipment, face down and bottom first as shown in FIG. 22. The ledger card then is inserted in the slot formed between the lowermost side of plate 365 and the topmost side of table 232, as shown by the dotted line in FIG. 23, until the leftmost vertical edge of the ledger card is oriented flush against the front face of plate 364. At this time, the radiant energy as seen by each of photocells (PCA1) and (PCA2), is cut-off, and, consequently, both of the photocells are thereby de-energized. Photocells (PCA1) and (PCA2) are connected in an electrical control circuit in such a manner, as will later be described, that when they are both de-energized, paper guide motor (PGM FIG. 14) is permitted to thereafter be de-energized.

Additional photoelectric means is utilized to initiate energization of the paper-guide motor when the leading or bottom edge of the ledger card has been placed in a certain prescribed position with respect to card-drive platen 241 after certain other prescribed conditions, not yet described, have been met. As shown in FIG. 22, a groove 370 is circumferentially formed in platen 241 and in this groove is mounted an inverted and substantially V-shaped acrylic and transparent plastic rod having independent and disconnected leg portions 371 and 372, as shown in FIG. 17. A light source 373 is mounted on bracket 236 and is disposed at the lowermost end of leg portion 371, whereas, a photoelectric cell (PCS) is also mounted on bracket 236 with the light-sensitive area thereof disposed adjacent the lowermost end of leg portion 372.

Photoelectric cell (PCB) is normally de-energized, however, when the leading edge portion of the ledger card is positioned by the operator on top of the card-drive platen **241** adjacent the upper junction of leg portions **371** and **372**, radiant energy from light source **373** is first projected upwardly through leg portion **371** and out of the uppermost end thereof. Thereafter, the radiant energy is reflected downwardly by the leading edge portion of the ledger card through leg portion **372**, and, consequently, impinges the active area of, and thus energizes photocell (PCB). When photocell (PCB) is energized, assuming the before-mentioned prescribed conditions are met, paper guide motor (PGM) is energized and thereby rotates latching cams **301** (FIG. 19) in a counter-clockwise direction, as previously described, to latch upper compression rolls **296** against card drive platen **241** with the leading edge of the ledger card fixedly secured therebetween.

After the lapse of a time delay of approximately one-quarter of a second following energization of paper guide motor (PGM), card drive motor CDM (FIG. 14) is thereafter energized and thereby rotates the card drive platen to automatically feed the ledger card into the accounting machine carriage in such a manner that the magnetic strip thereof is passed over read-write heads **346–348** (FIG. 19), thereby effectively establishing three information channels in the magnetic strip which are disposed parallel with respect to one another and extend substantially the entire length of the left margin of the ledger card.

Photoelectric means are also utilized to selectively energize suitable electronic amplifiers associated with each of magnetic pick-up heads **346–348** at a predetermined time substantially coincidental with the first passage of the leading edge of the ledger card over the read-write assembly. Accordingly, leg portion **371** is provided with a branch leg portion **374** through which radiant energy from light source **373** is directed. A slot **375** is formed in paper guide **303** and a photoelectric cell, designated (PCC), is mounted at the side of, and in substantially parallel alignment with respect to branch portion **374**, the light-sensitive area thereof being oriented substantially co-planar with the uppermost end of leg portion **374**.

In operation, photocell (PCC) is normally de-energized. However, when the leading edge of the ledger card is adjacent slot **375** in paper guide **303**, the radiant energy from light source **373** is first directed upwardly through leg portion **374** and out of the uppermost end thereof through slot **375**. Thereafter, the radiant energy is reflected by the leading edge portion of the ledger card and thereby projected downwardly and onto the sensitive area of photocell (PCC) to effect the desired energization of the amplifiers associated with pick-up leads **346–348**.

Photocell (PCB) is also utilized to detect the time at which the ledger card has been driven to a predetermined position in the card handling equipment and to initiate, at that time, reversal of the card-drive motor to effect ejection of the ledger card or to effect positioning of the card in the accounting machine carriage, depending upon the "instructions" previously given to the computer as will later be seen. For these purposes, as seen in FIG. 22, ledger card **305** is provided with a dark area or "reversal spot" **376** which is located in the top margin thereof in alignment with photocell (PCB). Consequently, when reversal spot **376** is adjacent the junction of arms **371** and **372**, radiant energy from light source **373** is absorbed by the dark area thereof and photocell (PCB) is thus de-energized to initiate the desired reversal of the card-drive motor.

8. *Electromechanical Control Means*

As will fully be described hereinafter, various control means are utilized by the computer to allow automatic insertion of a ledger card into the accounting machine carriage only when the carriage is resting approximately two inches from its maximum right-hand position, as shown in FIGURE 1; this position will be referred to hereinafter as the "home" position of the carriage. As

it is necessary that the accounting machine carriage remain in home position during operation of the card handling equipment, locking means are utilized to selectively lock and unlock the carriage while in home position.

With reference to FIG. 29, an L-shaped bracket **380** is welded or otherwise secured to back side-plate **235** and pivotally supports cam **381** at the lowermost end thereof by means of pin connection **382**. Connecting rod **383** is slidably mounted in a bore formed in bracket **380** and plate **235**, and is pivotally connected at one end thereof to the upper leftmost portion of cam **381** by means of pin connection **384**. Spring **385**, disposed about rod **383**, is interposed between collar **386** affixed to rod **383** and bracket **380**, thus spring-biasing rod **383** to the left as viewed. The remaining end of rod **383** is pivotally connected to the lowermost end of latch **387** by means of pin connection **388**, the uppermost end of cam **387** being pivotally connected to bracket **389** by means of pin connection **390**. Bracket **389** is fixedly secured to plate **391** which, in turn, is fixedly secured to plate **236** by means of screws **392** threaded therein. Fastened to bracket **391** by means of nuts **393**, is an electrically-actuated rotary type solenoid (CHP) which is substantially identical in construction as the solenoid designated **L46** which was previously described with respect to FIG. 9. A cylindrical-shaped plunger housing **395** is fixedly secured to bracket **391** by means of screws **396** threaded in the side of housing **395**. As shown more clearly in FIG. 30, slidably mounted in the bore formed in housing **395** is a rod-shaped plunger **397** which is spring-biased upwardly, as viewed, by means of spring **398** disposed thereabout and interposed between the shoulder formed on the upper end thereof and the bottom of the bore formed in housing **395**. Threaded on the end of plunger **397** is a truncated conical member **399** which is normally engaged by projection **400** formed on the lowermost end of latch **387**.

Thus, when the accounting machine carriage is in home position and a carriage opening operation is initiated by counter-clockwise rotation of bail **284**, as previously described, cam **381** is engaged by bail **284** and is rocked clockwise thereby. Clockwise rotation of cam **381** shifts connecting-rod **383** laterally to the right and thereby disengages latch projection **400** from member **399**, and thus permits spring **398** to deflect plunger **397** upwardly into engagement with the teeth formed on the lowermost side of carriage rack rail **267**, thereby locking the accounting machine carriage in home position. When it is desired to selectively unlock the machine carriage, solenoid (CHP) is selectively energized and consequently an off-set pin **404**, carried by the rotary armature thereof, engages member **399** and deflects plunger **397** downwardly to a point where it is again latched by latch projection **400**. After solenoid (CHP) is de-energized, plunger **397** remains latched in its downwardly deflected position, if bail **284** has previously been restored to the position shown in FIG. 29, until another carriage opening operation is initiated.

As previously mentioned with respect to FIG. 2, upper, middle and lower motor-bars **23**, **27** and **28**, respectively, are enclosed by cover **35** and are thereby rendered non-accessible to the operator of the machine. With reference to FIG. 38 there is shown a side-elevational view of the motor-bars along with a portion of the cycle-initiating mechanism associated therewith. FIG. 38 essentially corresponds to FIG. 119 of the before referred-to Patent 2,626,749, and, as a full and complete description of each of the various motor-bar mechanisms is available therein, a further detailed description thereof is not deemed essential for a full and complete understanding and appreciation of the present invention.

However, in accordance with the present invention, key stems **401** thru **403** of motor-bars **23**, **27**, and **28**, respectively, are respectively operated by the rotary armature of electrically-actuated solenoids (MBϕ), (MB1), and (MB2) by means of linkages **407** thru **409**, respectively connected therebetween and extending through suitable openings in base **65** of the accounting machine. Solenoids (MBϕ), (MBI), and (MB2) are each affixed to bracket **410** which is bolted to the underneath side of base **65** and are essentially of identical construction as the previously described rotary-type solenoid (L46) shown in FIG. 9. Thus, it is evident that by selectively energizing one of the motor-bar solenoids for a predetermined time, the corresponding motor-bar is automatically depressed for a corresponding predetermined time and thereby initiates either a "touch" or a "hold" operation which, in turn, selects a predetermined cycle of operation of the accounting machine, as previously described.

With reference to FIG. 35, a pair of rotary-type electrically-actuated solenoids (OH1) and (OH2), substantially of identical construction as those previously described, are each mounted on one of brackets **417** and **418** by means of nuts **419** and **420**, respectively; brackets **417** and **418**, in turn, are secured to suitable flanges, depending from the underneath side of frame **52**, by means of screws **421** and **422**, respectively threaded therein. As shown more clearly in FIGS. 32 and 34, the rotatable armatures of solenoids (OH1) and (OH2) are individually provided with a projecting leg **424** and **423**, respectively, which as shown in FIG. 35, are adapted to respectively engage the tails of the eighth and the eleventh ones of order-hooks **76**, counting from right to left. As previously described, the eighth and eleventh order-hooks, respectively control the operation of the eighth and the eleventh ones of type sectors **67** which respectively print the sixth and ninth-order digits of the amount row, with each digit printed being followed by a comma.

Thus, by selective energization of solenoid (OH1), the armature thereof is rotated counter-clockwise, as viewed in FIG. 32, to unlatch all order-hooks, from the third to and including the eighth one, from their respective type-sectors. Likewise, by selective energization of solenoid (OH2), the armature thereof is rotated counter-clockwise, as viewed in FIG. 34, to unlatch all order-hooks, from the third to and including the eleventh one, from their respective type-sectors.

As previously described in connection with FIGS. 16 and 17, during a carriage opening operation, square shaft **283** is normally rocked counter-clockwise by arm **285** and thus effects the disengagement of lowermost compression rolls **288** from the accounting machine platen **71**. However, for reasons to become more apparent hereinafter, means have been provided for selectively preventing the lower compression rolls from being disengaged from the platen. As shown in FIG. 16, a rotary-type electrically-actuated solenoid (X), of substantially identical construction as those previously described, is mounted on an L-shaped bracket **426** by means of nuts **427**, bracket **426**, in turn, being fixedly secured to back side-plate **235** by means of screws **428** threaded in bracket **426**. An S-shaped bracket **429** is fixedly secured to back side-plate **235** by means of screws **430** threaded in the foot of bracket **429**. As shown in FIG. 15, bracket **429** slidably supports a rectangular-shaped slide **431** by means of studs **432** extending through elongated slots formed in slide **431** and affixed to bracket **429**. Slide **431** is normally spring-biased to the right, as viewed, by means of spring **438** connected between pin **433** carried by slide **431** and block **434** welded or otherwise secured to plate **435** of the machine frame.

When slide **431** is in its normal position, as shown in FIG. 15, a projection on the uppermost side of the rightmost end thereof is directly underneath a projection formed on arm **285**, as further shown in FIG. 16. Consequently, counter-clockwise rotation of arm **285** is thereby prevented and thus the lower compression rolls are maintained in engagement with platen **71**. However, when solenoid (X) is selectively energized, pin **436**, carried by the rotatable armature thereof, engages flange **437** formed on the lowermost side of the left end of slide **431** opposite arm **285**, as viewed in FIG. 15, and thus deflects slide **431** to the left and thereby removes the obstruction from beneath arm **285** to permit the arm to be rocked counter-clockwise when desired. When solenoid (X) is de-energized, spring **438** urges slide **431** back toward its home position as shown.

In the just-described accounting machine, it is possible to take a true negative total from any one of the totalizers numbered "one" through "nine" inclusive, due to the incorporation therein of an overdraft control mechanism. By the use of this mechanism, the machine is caused to automatically operate through two cycles of operation. On the first machine cycle, the negative amount, which at this time is in a complementary form, is taken from the selected totalizer and is subtracted from the cleared X-totalizer, thereby causing the true negative total to be set up on the wheels of the latter totalizer. During the second machine cycle, the total is taken from the X-totalizer and the selected totalizer is "non-added," thereby causing the absolute value of the negative total or credit balance to be printed-out. Hence, at the end of the overdraft operation, the selected totalizer is cleared, as is also the X-totalizer, and the absolute value of the credit balance is printed on the record material. If desired, a sub-total of the negative amount may be taken instead of the total, thereby leaving the selected totalizer in the same condition at the end of the overdraft operation, as it was at the beginning.

Therefore, when it is desired to take a total or a sub-total from one of the totalizers and overdraft indicators **34** (FIG. 2) visually indicate that the number held by the particular totalizer is a negative amount, in addition to depressing the proper totalizer-selecting key and the credit balance key, the total key pertaining to the particular totalizer involved must also be depressed. If a sub-total operation is desired, the sub-total key must be depressed in addition to the previously-mentioned keys, so as to cause the negative amount to be placed back into the totalizer. Thereafter, the proper motor-bar is depressed so as to put the machine through a two-cycle credit balance operation and thus cause the absolute value of the negative amount to be printed-out.

With reference to FIG. 37, there is shown a fragmentary view of a portion of the overdraft control mechanism linkage which is fully shown in FIG. 22 of the aforementioned Patent 2,626,749. However, due to the fact that the overdraft control mechanism is fully shown and described in the aforementioned patent, it is believed that but a brief description thereof is necessary to herein be presented in order to make readily apparent the significance of the modifications thereof made in accordance with the present invention.

As shown, the lower end of overdraft pitman **442** extends through an opening formed in base **65** of the accounting machine and is pivotally connected to bell-crank lever **443**, the bell-crank lever being rotatably mounted on rod **444** which is secured to the under side of base **65** by suitable means not shown. The remaining arm of bell-crank **443** is providled with a stud **445** which cooperates with a slot formed in the upper edge of overdraft trip-arm **446**, trip arm **446** being rotatably supported on overdraft cam shaft **447** which is journaled in brackets **448** secured to the bottom side of base **65**. Trip arm **446** has a notch formed on the end thereof which is normally urged by spring **449** into engagement with stud **450** carried by the lower end of lever **451**. Lever **451** is pivotally mounted on rod **444** and has rotatably pinned to the lowermost end thereof, roll **452** which is urged by spring **449** toward the periphery of a plate cam, not shown, which, in turn, is secured to overdraft cam shaft **447**.

Consequently, as the credit balance key ("CR. BAL."—FIG. 2) is operatively connected to the uppermost end of pitman **442**, manual depression of the credit balance key causes pitman **442** to be depressed at the beginning of the cycle of operation of the machine. When pitman **442** is depressed, bell-crank **443** is rocked clockwise, thereby causing stud **445** to rock overdraft trip-arm **446** counter-clockwise and thus release lever **451** to the urgency of spring **449**. Due to the action of spring **449**, lever **451** is rocked clockwise to engage roll **452** with the periphery of a plate cam, mentioned above, to effect the carrying-out of the remainder of the cycle of operation, as fully described in the aforementioned Patent 2,626,749. Near the end of the second cycle of operation, a high point on the plate cam effects re-engagement of stud **450** with the notch formed on the end of overdraft trip-arm **446**, thus completing the cycle of operation.

In accordance with the present invention, means are provided for automatically effecting selective depression of overdraft pitman **442** to automatically initiate the previously-described overdraft cycle of operation. Such means includes a rotary-type electrically-actuated solenoid (CBS) which is essentially of identical construction as those previously described. Solenoid (CBS) is mounted on bracket **454** by means of nuts **455**, bracket **454** being fixedly secured to the under side of base **65**. The rotatable armature of solenoid (CBS) is provided with an extension **456** which engages roller **457** pinned to the end of bell-crank **443**. Consequently, each time solenoid (CBS) is energized, bell-crank **443** is rocked clockwise and thus initiates a credit balance cycle of operation, thereby automatically effecting a print-out of negative numbers under the electronic control of the computer, as will be more fully described hereinafter.

With reference to FIG. 38, when the carriage throat shaft **458** is rotated counter-clockwise, as viewed, control plate **459**, fixedly secured to shaft **458**, is also rocked counter-clockwise, thereby tripping a carriage throat clutch mechanism, not shown, thus causing the throat mechanism to be operated by a power shaft, all in a manner fully described in the referred-to Patent 2,626,749.

In accordance with the present invention, a rotary-type electrically-actuated solenoid (Y), essentially of identical construction as those previously described, is fixedly secured to bracket **461** by suitable means, not shown. Bracket **461** is fixedly secured to bracket **462** by means of screws **463** threaded therein, bracket **462**, in turn, being fastened to the under side of base **65** by means of screws **464** threaded therein. The rotatable armature of solenoid (Y) is effectively connected to control plate **459** by means of linkage **465** connected therebetween. Consequently each time solenoid (Y) is energized, plate **459** is rocked counter-clockwise, as before. As a result, the carriage throat clutch is tripped thereby, as mentioned before, and the opening and closing of the carriage of the accounting machine is automatically effected solely under control of the computer.

In order to initiate a total-taking operation as previously described, it is necessary that the zero stop pawls **46** (FIG. 3A) be rocked counter-clockwise to release the differential actuators **49** and permit rearward movement thereof. During the rearward movement of the actuators, selected ones of the totalizers are rotated in a reverse direction until stoped by suitable "add" tripping pawls, not shown, to thereby position the differential actuators **49** and the printing sectors **67** in accordance with the amount indicated by the rotatable positions of the totalizer wheels. In order to initiate an addition or subtraction operation, the zero stop pawls **46** are removed by depression of the amount keys, as previously described. In total-taking operations, however, no amount keys are depressed, and thus it is necessary to utilize other means for releasing the actuators from the zero stop pawls.

With reference to the fragmentary view of FIG. 40, which is fully illustrated in FIG. 36 of the previously mentioned Patent No. 2,626,749 and fully described therein, means are provided for moving the key release slide **470** at the outset of the machine cycle, thereby causing the stud **471**, affixed to the upper end thereof, to deflect the plate **472** clockwise, as viewed. The plate **472** is fixedly secured to the shaft **473**, as is the plate **474**, which supports a stud **475** on the opposite end thereof. Therefore, when the plate **472** is deflected clockwise, the plate **474** is likewise deflected clockwise, thereby causing the stud **475** to act against the uppermost surface of the cam **476**, which is provided on each of the control slides **44**, thereby moving the slides toward the rear of the machine and thus releasing the zero stop pawls **46**, as described in connection with FIG. 3A. At the same time, the studs **475** cooperate with suitable cam projections, not shown, which are provided on the latching slides in the amount key banks, so as to release any amount keys which may have accidentally been depressed prior to the initiation of the total-taking cycle of operation. Consequently, the square studs **41** on the lowermost end of the key stems are withdrawn and thus are prevented from coacting with the shoulders **50** on the differential actuators **49**, and, therefore, are not permitted to stop the actuators in positions other than those dictated by the amounts standing on the totalizer wheels.

The actuation of the key release slide **470** near the beginning of the machine cycle, rather than near the end thereof, is accomplished by means of a key release latch **477**, pivotally mounted on a stud **478**, which is carried by the rear end of the slide **470**. The rear end of the latch **477** is formed into a hook **479**, which engages a stud **480**, mounted in a depending arm of a lever **481**, which, in turn, is pivotally mounted between the side frames of the machine. The lever **481** is operatively connected to a roller, not shown, which engages a cam mounted on the main cam shaft of the machine. Hence, when the main cam shaft is rotated in one direction, the latch **477** is deflected to the left, as viewed, by the stud **480**, and thus the key release slide **470** is operated for the beforementioned purpose.

In order to prevent the just-described early key release from occurring during an addition or subtraction operation, the selecting slides for the various totalizers are each provided with suitable abutments, not shown, which overlie the bail **484** when the slides are located in their total-taking positions. As partially illustrated and as shown more clealry in FIG. 39, the bail **484** is secured at its ends to a shaft **485** journaled at either end in the side walls of a bracket **486**. Although not shown, the shaft **485** is operatively connected to be rocked clockwise, as viewed in FIG. 40, by the main cam shaft of the machine. Consequently, a stud **487**, mounted in an arm **488**, which is pinned to the right-hand end of the shaft **485**, deflects the key release latch **477** counter-clockwise, thereby uncoupling the hook **479** from the stud **480** and thus preventing early release of the amount keys. However, if any of the totalizer control slides is located with its abutment positioned above the bail **484**, clockwise rotation of the shaft **485** is prevented, and the key release latch **477** remains coupled to the stud **480**, so as to bring about an early release of the amount keys.

With reference to FIG. 39, in accordance with the present invention, means are provided for automatically bringing about an early release of the amount keys by the provision of an electrically-actuated solenoid (RRS) secured to a bracket **490** by means of screws **491** threaded in the frame thereof, the bracket **490**, in turn, being fixedly secured to the bracket **486** by means of screws **492** threaded therein. A substantially Z-shaped armature **493** of the solenoid (RRS) is pivotally mounted on a pin **494** in such a manner that, each time the solenoid (RRS) is energized, the rightmost leg of the armature **493**, as viewed, is moved upwardly to present an obstruction to the bail **484**. Consequently, clockwise rotation of the bail **484** is prevented, even thorugh there are no totalizer control slides which are located with their abutments positioned above the bail **484**, as previously described.

As will be described more fully hereinafter, there are certain "instructions" which are utilized and carried out by the computer and which effect a different cycle of operation of the accounting machine portion thereof. For proper control of the accounting machine portion, however, it is necessary that means be provided to "sense" the states or conditions of the various portions of the machine either before, during, or after a cycle of operation thereof. Such means include a plurality of switches located at various appropriate places within the machine and will, at this time, briefly be described only as to their location and function, together with their manner of actuation. The following to-be-described switches, along with the just-described electrical components such as the photocells, solenoids, etc., will hereinafter again be described in connection with their actual incorporation into the electrical control circuitry of the computer.

For example, neglecting for the moment the tape-punching mechanism, there are three switches which are utilized in a logically "ANDED" combination to indicate by their closure when depression of a particular motor bar will cause a cycle of operation of the accounting machine. With reference to FIG. 38, a bell crank or "able arm" **497** is pivoted about a stud **498** and, in a manner fully described in the aforementioned Patent No. 2,626,749, is operatively connected to the various mechanisms of the motor bars **23**, **27**, and **28** in such a way that, when deflected in a clockwise direction, as viewed, all motor bars are disabled and thus rendered ineffective. The able arm **497** is deflected clockwise when a key on the accounting machine keyboard is partially depressed, when the carriage is off a stop and the "non-select" key is not depressed, when the carriage is returning, when the electrical power to the accounting machine driving means is turned off by the ON-OFF switch, when the machine is in a cycle of operation, or when a motor bar is depressed. In order to detect the condition of the able arm **497**, a switch **496** is fixedly secured to the side frame of the machine by suitable means, not shown, and is adapted to have its movable arm **499** actuated by linkage **500**, which is connected to the upper end of the able arm. Normally, the contacts of the switch **496** are closed; however, when the able arm **497** is deflected clockwise for reasons previously mentioned, the contacts of the switch **496** are opened, thereby indicating that the motor bars are disabled.

In order to enable the accounting machine to perform its record-keeping functions, the traveling paper carriage is adaptable to be tabulated from one columnar position to another columnar position, while traveling in either a forward or a reverse direction, so as to enable amounts to be printed in the proper column of the record-keeping form. The tabulation control mechanism utilized is extremely flexible in character and is selectively controlled either from special keys on the keyboard, from the motor bars, or from the stops of the traveling paper carriage itself.

Consequently, as the able arm does not disable the motor bars during a carriage tabulation cycle, it is therefore possible to manually hold the carriage over a carriage stop, even though the carriage is attempting to tabulate, and thereafter depress a motor bar, thereby initiating a machine cycle. Therefore, to prevent the electrical controls for the computer from initiating a depression of a particular motor bar while the carriage is tabulating over a carriage stop, a switch, shown in FIG. 43 as **510**, is utilized to sense a carriage tabulation cycle. As shown, the forward tabulation lever **501** is rotatably and slidably supported by means of an elongated slot formed therein cooperating with a frame stud **502** secured to an angle bar **503**. The lever **501** is normally urged to the right, as viewed, by means of a spring **504** and is further urged in a clockwise direction about the stud **502** by means of a spring **505**. The action of the springs **504** and **505** is resisted by a square stud **506**, which lies beneath the right-hand end of the lever **501** and is secured to the rear face of the carriage support plate **503**. The stud **506** cooperates with a shoulder **507** formed on the lever **501** and thereby prevents the lever **501** from being moved either to the right or to be rotated under the influence of the springs **504** and **505**.

In operation, when tabulation is initiated depressed, among other things not pertinent hereto, the lever **501** is first lifted off the stud **506** and is thereafter moved to the right by the spring **504** and rocked clockwise by the spring **505**, to free the carriage for movement and to effect completion of the tabulation cycle. Upon completion of the carriage tabulation cycle, the lever **501** is restored to the position shown.

To sense a tabulation cycle, an extension **508** is pivotally connected at one end to the end of the lever **501** and is rotatably and slightly mounted by means of an elongated slot formed therein cooperating with a stud **509** mounted in the support plate **503**. A switch **510** is mounted on the support plate **503** in such a manner that the contacts thereof are opened by extension **508** due to the clockwise movement of the lever **501**, thus indicating that the carriage is operating in a tabulating cycle.

As previously mentioned in connection with FIG. 38, all previously-mentioned conditions being met, the contacts of the switch **496** are closed by the able arm **497** after the machine has progressed through approximately seventy-five percent of its cycle of operation. Consequently, another switch is utilized to indicate when the machine cycle has been sufficiently completed to the extent that depression of a motor bar will cause another cycle of operation. With reference to FIG. 41, there is shown a slide **513**, which controls the reverse tabulation mechanism of the accounting machine and may therefore be referred to as the "reverse tabulation" slide which initiates the return of the carriage. The slide **513** is provided with elongated slots by means of which it is slidably mounted on suitable studs, not shown, which, in turn, are mounted on an auxiliary frame plate of the machine. The slide **513** is resiliently urged toward the front of the machine, or to the left, as viewed, by means of a spring **514**, but is normally retained in its rearward position, against the urgency of the spring **514**, by means of a rod **515**, secured to the upper end of a lever **516**. The lever **516** is pivotally mounted on a stud **517**, which is secured to the auxiliary frame plate, and its lower end is pivotally connected to the forward end of a link **518**. The rear end of the link **518** is pivotally connected to the upper end of a follower arm **519**, which is rotatably mounted on a shaft **520** and is provided with rolls **521** and **522**, which respectively cooperate with a pair of companion cam plates **523** and **524**, secured to the main cam shaft **525** of the accounting machine.

Hence, during each cycle of operation of the accounting machine, the cam shaft **525** is rotated clockwise, as indicated by the arrow, and thereby deflects the follower arm **519** clockwise to the position indicated by the dash-dot line. Consequently the link **518** is shifted to the right, and the lever **516** is deflected counter-clockwise thereby and thus moves the rod **515** to the front of the machine. Consequently, the slide **513** is permitted to be moved forwardly under the influence of the spring **514** and to be differentially positioned under the control of the motor bars. The lever **516** also carries an aligner bar **526**, which is adapted to engage aligner notches **527** formed on the lowermost edge of the slide **513**, so as to align the slide firmly in the position which it has assumed while being sensed by a feeling mechanism, which, again, is fully shown and described in Patent No. 2,626,749. Upon completion of the cycle of operation, the follower arm **519** is deflected counter-clockwise to again engage the rod **515** with the slide **513**, as shown.

In order to sense whether the machine is in home position ready for another cycle of operation or whether the machine, at the moment, is in the process of carrying out a cycle of operation, the switch **528** is utilized and is fixedly secured to a bracket **529** by means of nuts **530**. The bracket **529**, in turn, is fixedly secured to a bracket **531** by means of screws **532** threaded therein, bracket **531** being secured to the base **65** of the machine by suitable means, not shown. As shown more clearly in FIG. 42, the link **518** is provided with a stud **533**, which is adapted to engage a roller **534**, rotatably carried by the actuator of the switch **528**. Consequently, when the machine is in home position, as shown in FIG. 41, the stud **533** is in engagement with the roller **534**, and the contacts of the switch **528** are thereby closed, indicating that the machine is ready to start another cycle of operation. However, when the machine starts to initiate another cycle of operation, the pressure exerted by the stud **533** on the roller **534** is relieved, and the contacts of the switch **528** are opened thereby.

The three just-described switches **496**, **510**, and **528** do not detect when a motor bar is disabled due to one of the item counters being off home position, or when one of the manual counter control keys is partially depressed. Consequently, even though the collective closure of the three switches **496**, **510**, and **528** indicates that a motor bar is permitted to be depressed by the computer, yet the motor bar may still be disabled mechanically. In fact, if a motor bar were to be depressed at this particular time, it is possible that the electronic control portion of the computer would receive an indication that the accounting machine portion thereof had started into a cycle of operation, whereas, in actuality, the machine did not even start to cycle.

To avoid the necessity of many additional switches to detect why the accounting machine portion of the computer did not cycle, a single switch is utilized to detect when the machine did not cycle, regardless of the reasons why it did not cycle. Consequently, if the accounting machine portion does not cycle, for some reason or another, after the electronic portion of the computer has effected depression of a motor bar, the electronic portion "senses" the fact that the machine did not cycle and thereafter continually effects periodic depression of the motor bar until the machine finally cycles.

With reference to FIG. 26, a flat plate **537** is secured to a shaft **538**, which is journaled in the side frames, not shown, of the electric typewriter portion of the accounting machine. Also secured to the shaft **538** is an arm **539**, which is operatively connected to the accounting machine mechanism in such a manner that, each time the accounting machine starts into a cycle of operation, the arm **539** is rocked counter-clockwise, as viewed, and thereby positioned as indicated by the dot-dash line. However, when the lever **539** is rocked counter-clockwise, the plate **537** is thereby moved into blocking position between the typewriter keys to prevent a typewriter key from being depressed while the machine is carrying out a cycle of operation. Upon completion of a cycle of operation, the lever **539** resumes its position, as shown.

Thus, in order to sense whether a cycle of operation is being carried out, a switch **540** is fixedly secured to the base **65** by means of screws **541** threaded therein, and is positioned so that a roller **542**, carried by the actuator thereof, is engaged by the lever **539** when the lever is in a raised position and the contacts of the switch **540** are operated thereby. As will be more fully described hereinafter, the contacts of the switch **540** are operated in the range of 50 to 200 milliseconds after a motor bar solenoid has been energized if the machine has started into a cycle of operation. Upon energization of a motor bar solenoid, the corresponding motor bar is depressed thereby, as previously described. After a time lapse of approximately 250 milliseconds after the motor bar solenoid is energized, the condition of the switch **540** is sampled. If the contacts of the switch **540** are not found to be operated, the motor bar solenoid is de-energized and thereby allows the depressed motor bar to resume its normal position. When the motor bar has resumed its normal position, such that re-depression thereof may cause a machine cycle, the corresponding motor bar solenoid is allowed to be re-energized to again repeat the sequence of operations.

As previously described with respect to FIG. 17, square shaft **283** is rocked counter-clockwise, as viewed, to effect the disengagement of lower compression rollers **288** and platen **71**. With reference to FIG. 19, arm **542** is fixedly secured to the underneath side of shaft **283** by means of screws **543** threaded therein and engages actuating lever **544** of switch **545** which is fixedly secured to bracket **546** by means of screws **547** threaded therein, bracket **546** being fixedly secured to bracket **236** by means of screws **548** threaded therein. When square shaft **283** is rocked counter-clockwise, its contacts are operated, thus indicating that lower compression rolls **288** (FIG. 17) are out of engagement with platen **71**.

With reference to FIG. 41, switch **550** is fixedly secured to bracket **551** by means of screws **552** threaded therein. Bracket **551** is fixedly secured to bracket **553** by means of screws **554** threaded therein, whereas bracket **553** is fixedly secured to the base **65** of the machine by means of screws **555** threaded therein. Switch **550** is positioned so that the contacts thereof are effectively operated by follower arm **519** during the time that the follower arm is rocked clockwise into engagement with the actuator **556** thereof, as shown by the dot-dash line, thus indicating that the accounting machine is in the process of carrying out a cycle of operation such that aligner bar **72** is in engagement the notches formed on the lower end of arm **58** (FIG. 3A), as previously described.

Also shown in FIG. 41 is a portion of the linkage utilized by the accounting machine carriage opening and closing mechanism, which is fully shown and described in connection with FIG. 139 of the before-referred-to Patent No. 2,626,749. Even though not fully shown in FIG. 41, lever **557** is pivotally mounted on stud **558** and has its leftmost end operatively connected to carriage open key "CARR. OPEN" (FIG. 2) in such a manner that depression thereof causes lever **557** to be rocked counter-clockwise.

In accordance with the present invention, switch **559** is fixedly secured to bracket **553** by means of screws **560** threaded therein and has its roller-type actuator **561** in engagement with the rightmost end of lever **557**. Consequently, when lever **557** is rocked counter-clockwise, the contacts of switch **559** are actuated thereby, thus indicating that the operator desires the carriage to be opened.

With reference to FIG. 44A, there is shown a fragmentary view of the particular mechanism controlled by stops on the traveling paper carriage for selecting the totalizers for addition, subtraction, or total-taking operation, the mechanism being fully shown and described with respect to FIG. 56 of the before-referred-to Patent No. 2,626,749. Briefly, however, stem **564** of totalizer reverse key "REV" also shown in FIG. 2, carries a stud **565**, which cooperates with the bifurcated end of lever **566**. Lever **566**, in turn, is pivotally mounted on stud **567** secured to partition plate **568** and has a depending arm, which cooperates with a stud **569**, mounted in an arm **570**, which, in turn, is loosely journaled on a shaft **571**. Thus, when the reverse key "REV" is depressed, arm **570** is rocked counter-clockwise, as viewed, and thereby effects reversal of the add and subtract selection mechanism of the totalizers from the carriage stops, as fully described in the above-referred-to patent.

In accordance with the present invention, switch **572** is fixedly secured to bracket **573** by means of screws **574**, bracket **573** being fixedly secured to partition plate **568** by means of screws **575** threaded therein. Switch **572** is properly positioned in such a way that its roller-type actuator **576** is deflected by stud **569** and thus operates the contacts thereof each time the reverse key "REV" is depressed.

In addition, a rectangularly-shaped slot **562** is formed in the partition plate 568 and is properly oriented to have slidably disposed therein a square stud 563 projecting from the lowermost end of key stem 564. An elongated arm 577 has one end thereof pivotally supported by a screw 578 threaded in the partition plate 568 and has supported on its opposite end a pair of studs 585, which are disposed above and below the square stud carried by the key stem 564. Also supported on the opposite end of the arm 577 is an electrically non-conductive stud 586, which is adapted to engage the movable arm 587 of a switch mechanism, indicated generally as SC46, which, in turn, is supported by the plate 568 by suitable means, not shown.

Therefore, each time the reverse key "REV" is depressed, the square stud on the lowermost end of the stem thereof deflects the arm 577 clockwise, as viewed, and, consequently, the movable contacts of the switch SC46 are actuated by stud 586. When the reverse key is released and allowed to assume the position shown, the movable contacts of switch SC46 are likewise allowed to assume their normal positions, as shown.

With reference to FIGS. 24 and 25, there is shown a partial plan view of the rear rack 276 and the driving gear means 277, which were previously described in connection with FIG. 17, for effecting linear translation of the accounting machine carriage. However, in order to sense the home position of the accounting machine carriage, a switch 579 is fixedly secured to the plate 240 by screws 580 and, as more clearly shown in FIG. 25, is so positioned that its roller-type actuator 581 is adapted to be engaged by a stud 582, depending from the lower side of the rack 276, each time the carriage is in home position. Consequently, the normally-opened contacts of the switch 579 are closed as long as the carriage is in home position and are opened as soon as the carriage leaves home position.

With reference to FIG. 27, manually-operable means are provided to selectively lock the typewriter carriage-return keys, numbered (1) through (3). Such means comprises essentially a substantially flat plate 583, which is normally spring-urged to a position under the key stems, as shown, to prevent depression thereof. However, when lever 584 is moved forward, or to the right as viewed, its tail end cams the plate 583 outwardly to permit selective depression of the keys.

9. *Building Blocks Generally*

Before going into a detailed description of the individual units making up the electrical circuitry of the computer shown by the block diagram thereof in FIG. 45, there will now be described the building blocks with which these individual units and interconnections therebetween are constructed. A building block is herein defined as a structure which is basic to the operation of the individual units of the computing system, and is used repeatedly in the system for amplification, logic, or other control purposes. By establishing a set of building blocks and understanding the rules for their interconnection, such a complex computing system as that heretofore generally described is more readily understood. The building blocks that are of particular importance are inverters, amplifiers, triggers, and logical circuits such as "ANDS" and "ORS," etc.

10. *Inverter Building Blocks*

The building block to be first described is the inverter, which, as the name implies, effects an inversion of the signal applied to the input thereof. In a transistor circuit, the grounder-emitter amplifier is capable of performing such an operation. Thus, in order for the inverter to perform such a task, it must produce an output signal that is "up" when the input signal is "down," or, stated another way, it must produce a signal indicative of binary "one" upon the application of a signal to its input indicative of binary "zero"; conversely, it must produce a binary "zero" signal upon the application of a binary "one" to its input. However, throughout the following description of the electrical circuitry of the instant computer, for the purposes of clarity, when the output signal is "up"—i.e., substantially at or above ground potential—it will be referred to as being "TRUE," indicative of a binary "one." However, when the output signal is "down"—i.e., below ground potential at a potential of approximately —6 volts, it will be referred to as being "FALSE," indicative of a binary "zero."

With reference to FIG. 46A, there is shown a simple grounded-emitter type inverter comprising transistor 650 having its base connected to input terminal 651 and its collector connected to output terminal 652. The symbol for this type of inverter is the notation "I1" enclosed by a circle, as shown at the bottom of the block. When a negative potential is applied to the base of transistor 650 via input terminal 651, base current flows outwardly from the base thereof toward input terminal 651. If the collector electrode of transistor 650 is connected to a negative potential via output terminal 652, emitter current flows from the emitter to the collector of transistor 650. Assuming a sufficient flow of base current, the resistance between the collector and the emitter becomes a very low value, and, as a result, output terminal 652 is essentially at ground potential, indicative of a TRUE signal. When either ground or a positive potential is applied to the base of transistor 650, the resistance between the collector and the emitter, and the resistance between the base and the emitter, are all of a substantially high value. Consequently, the values of both the base and collector currents go essentially to zero, and the voltage at output terminal 652 is thereafter negative.

Inverter I2 shown in FIG. 46B is substantially the same as inverter I1 of FIG. 46A, with the exception that the emitter electrode of transistor 650 is now returned to terminal 663 instead of ground potential. Inverter I3 of FIG. 46C is essentially the same as inverter I1 but is additionally provided with a clamping diode 653, connected between the collector of transistor 650 and bias terminal 654, to "clamp" the collector to the negative operating potential at bias terminal 654. Inverter I4 is essentially the same as I1 but, additionally, has a dropping resistor 659 in its output circuit. Inverter I5 is essentially the same as I1 but is additionally provided with a dropping resistor 655 connected between the collector of transistor 650 and bias terminal 656. I6 is substantially the same as I1, with dropping resistor 660 in its input circuit. I9 is substantially the same as I6 but further includes resistor 661, connected between the base of transistor 650 and bias terminal 662, to insure that the transistor remains non-conductive when so rendered. Inverters I7, I8, I10, and I12 through I16 are substantially the same as I1, except that they are additionally provided with various combinations of bias dropping resistors 655, 657, 658, and 661, clamping diodes 653, and input dropping resistor 660. I11 is substantially the same as I5, with dropping resistor 664 connected between the emitter of transistor 650 and ground potential. Inverter I17 is substantially the same as I7, except for the addition of input coupling condenser 667 and base resistor 668. The purpose of condenser 667 is to provide a large surge current in order to overdrive the transistor momentarily, to decrease the rise time of the output signal and thus increase switching speed. I18 through I23 and I25 utilize various combinations of the dropping resistors, clamping diodes, and coupling condensers previously described. In I24, diode 671 keeps the base of transistor 650 substantially at ground potential until the potential at terminal 651 becomes negative to render the transistor conductive. In I26, bypass conductor 665, connected between collector and ground of the transistor, provides a short circuit to eliminate or attenuate undesirable high frequencies. In I27, condenser 672 and resistor 670, each connected between collector and base of transistor 650, provide an inverse feedback to limit the gain of the stage at high frequencies. In I28, condenser **666** and resistor **664** provide a suitable negative feedback voltage proportional to emitter current to limit the gain of the inverter stage, and also provide a degree of temperature compensation thereof; resistors **655**, **658**, **668**, and **670** provide the necessary operating bias potentials thereto; condensers **665**, **672**, and **680** function as filter condensers which remove unwanted extraneous signals; and condenser **667** functions as a coupling condenser which a-c couples input terminal **651** to the base of transistor **650**. Inverter I29 is essentially a conventional emitter-input grounded-base amplifier with a series dropping resistor **678** in its base circuit.

11. *Emitter-Follower Building Blocks*

The next building block to be described is the emitter follower, which gets its name from its vacuum-tube counterpart, the cathode follower. The emitter follower type of amplifier provides power gain, has no phase inversion at low frequencies, has a slight attenuation of the input potential signal, and has a slight change in the d-c level of the signal from input to output. With reference to FIG. 47B, there are shown various types of emitter follower circuit configurations, each having a symbol which comprises the designation "E" normally followed by a numeral and both designations encompassed by a circle, as shown at the bottom of each sub-section (*a*) through (*k*) of FIG. 47B.

As shown in sub-section (*a*), emitter-follower E1 comprises a transistor **650** having its emitter returned to ground or reference potential through resistor **664**. The collector is connected to bias terminal **656** to normally receive a suitable negative d-c operating potential therefrom. Input terminal **651** is connected to the base, and output terminal **652** is connected to the emitter, of the transistor.

In operation, if the input signal and consequently the base of the transistor are TRUE—i.e., at ground potential—the transistor **650** is rendered substantially nonconductive, and the output at terminal **652** is therefore substantially at ground potential, or TRUE. However, when the input signal is FALSE—i.e., at a potential of −6 volts—the transistor is rendered conductive, and the potential at output terminal **652** is substantially equal to the negative potential on the base. Thus, when the input is TRUE, the output is TRUE, and, when the input is FALSE, the output is FALSE. Emitter follower **E2** in sub-section (*b*) is essentially the same as E1, with the exception that the lower end of resistor **664** is returned to bias terminal **663** instead of ground potential. In E3, output terminal **652** is connected to the emitter of transistor **650** through series dropping resistor **664**. E4 is substantially the same as E2 but for the addition of resistor **673** between output terminal **652** and the emitter of the transistor. E5 is essentially the same as E3, with the addition of resistor **661** connected between the base of the transistor and bias terminal **662**. E6 is substantially the same as E2 but for the addition of bias dropping resistor **655**. E7 is substantially the same as E4, except that, in this instance, resistor **673** is now paralleled with bypass condenser **666**. Emitter followers E8 through E10 and E1 utilize various combinations of dropping and bias resistors and coupling and bypass condensers, as previously described. E1, however, functions somewhat as both an inverter and an emitter follower in that an inverted output appears at output terminal **674**, while simultaneously an output potential appears at terminal **652** which is of the same polarity or phase as the input signal.

For the sake of simplicity and brevity, the preceding building blocks were discussed but briefly. However, for a more detailed and comprehensive description thereof, reference is made to the "Handbook of Semiconductor Electronics," by L. P. Hunter, published by McGraw-Hill Book Company, Inc., wherein many similar circuit configurations are found.

12. *Logical OR Building Blocks*

Performing logical operations is a well-known fundamental to the operation of a computer. The most basic logical operations are those of "AND" and "OR." Even though many of the logical operations performed in digital computers are concerned with arithmetic operations such as addition, subtraction, division, and multiplication, herein many of the logical operations are assembled into decision-making and information-routing circuits. In order to perform such logical operations, it is generally necessary to utilize nonlinear circuit elements, such as diodes and transistors and the like.

With reference to sub-sections (*a*) through (*g*) of FIG. 47A, there are shown various logical "OR" circuit configurations. In (*a*), there is illustrated a logical "OR" building block of type D1 comprising a plurality of diodes **679**, each having its cathode connected to the other. A "pull-down" resistor **690** is connected to the junction of the diodes **679** and to bias terminal **689**, which normally has a negative d-c operating potential applied thereto. A plurality of input terminals, in this instance shown as **675** through **678**, are individually connected to the anode of a different one of diodes **679**, and output terminal **688** is connected to the junction of the diodes.

In operation, if the input signal to any one or more of terminals **675** through **678** is "TRUE," or at a positive potential, the corresponding diode in each instance is conductive, its internal impedance is reduced substantially to zero, and consequently the potential appearing at output terminal **688** is substantially equal to the input potential. However, if the input signals to all of input terminals **675** through **678** are each at a negative potential, the output signal at terminal **688** goes negative via the negative potential at terminal **689** and resistor **690**. However, the negative potential at terminal **688** is not permitted to go more negative than the least negative potential at one of input terminals **675** through **678**. Thus, it is evident that the output of a logical "OR" circuit is "TRUE" (or positive) if any of the input signals to terminals **675**, **676**, **677**, or **678** is "TRUE" (or positive); the output is effectively "FALSE" only when all of the inputs are "FALSE." The symbol used herein for a logical "OR" is a semicircle designation having the representative input lines extending through the half-circle and terminating at the circumference thereof, as illustrated at the bottom of each of sub-sections (*a*) through (*g*).

Logical "OR" of type D2, shown in (*b*) of FIG. 47A, is substantially the same as type D1, but for the addition of dropping resistor **691** connected between the junction of didoes **679** and resistor **690**. In (*c*), a type C logical "OR" comprises a plurality of condensers **683**, each connected to the other and connected to output terminal **688**. A plurality of input terminals, illustratively shown as **675** through **678**, are each connected to the remaining end of one of the condensers **683**. In a type CR logical "OR," the condenser **683** and the resistor **687** are each connected to output terminal **688**, input terminals **675** and **676** being individually connected to the remaining end of the condenser and the resistor. Type CRD logical element is similar to type CR except for the addition of diode **679** connected between resistor **687** and output terminal **688**. Type R "OR" comprises but a single resistor **691** connected between input terminal **675** and output terminal **688**, and type RD1 "OR" comprises a series-connected resistor **691** and diode **679**, connected between input terminal **675** and output terminal **688**.

13. *Logical "AND" Building Blocks*

With references to FIG. 48A, there is illustrated a logical "AND" building block of type D1 which comprises a plurality of diodes **679**, each having their anodes connected together and connected to output terminal **688**. A plurality of input terminals, illustratively shown as **675** through **678**, are each individually connected to the cathode of a different one of diodes **679**. A "pull-up" resistor **690** is connected to the junction of diodes **679** and to bias terminal **689**, which is normally supplied with a suitable positive d-c operating potential having a magnitude, in most instances, of 6 volts.

In operation, when the input signal to each of the input terminals **675** is "TRUE"—i.e., at zero or ground potential—the output signal at terminal **688** goes substantially to ground potential via the positive potential at bias terminal **689** and resistor **690**. However, if any one or more of the input signals goes "FALSE" (−6 volts), the corresponding diode is conductive, and, consequently, the output signal at terminal **688** is substantially at −6 volts ("FALSE"). Therefore, it is seen that in a logical "AND" circuit, as just described, the output signal remains "FALSE" as long as any one or more of the input signals are "FALSE"; the output signal goes "TRUE" only when all of the input signals go "TRUE" and at no other time.

Type D2 "AND" circuit shown in (*b*) is essentially the same as the D1 type except for the addition of dropping resistor **691** connected between the anodes of the diodes **679** and the output terminal **688**. The D3 type shown in (*c*) is substantially the same as the D2 type but for the addition of bypass condenser **695** connected in parallel with the dropping resistor **691**. In the D4 type of logical "AND" circuitry shown in (*d*), pull-up resistor **690** has been omitted and replaced with a circuit comprising transistor **700**, having its base connected to the junction of the diodes **679**. The collector of the transistor **700** is connected to bias terminal **689**, having a negative d–c operating potential of approximately 6 volts applied thereto. The emitter of transistor **700** is connected to output terminal **688** and is also connected to one end of resistor **701**, the opposite end of resistor **701** being connected to bias terminal **702** having a positive d-c operating potential of approximately 6 volts applied thereto. By the use of such an emitter follower type amplifier essentially as a pull-up resistor, a power gain of the output signal results, the input and output signals thereto being at all times of the same polarity as previously described with respect to the emitter follower type building blocks.

The D5 type "AND" circuit shown in (*e*) is substantially the same as the D1 type except for the addition of bias resistor **692**, connected between input terminal **675** and bias terminal **693**, which has applied thereto a negative d-c operating potential. The DR type "AND" shown in (*f*) is somewhat different, in that resistor **696** now constitutes the pull-up resistor even though it is connected to one of the input terminals such as **675**. In operation, if the input signal to either or both of input terminals **675** and **676** is "FALSE" (−6 volts), the signal at output terminal **688** is likewise "FALSE." However, if the input signal to both of the input terminals is "TRUE" (ground potential), the signal at output terminal **688** is likewise "TRUE," as in the logical "AND" circuits just described.

In (*g*) there is shown a resistor type of logical "AND" circuitry R1 which comprises a plurality of resistors **696**, each connected to output terminal **688** and having the remaining end thereof individually connected to one of a plurality of input terminals, illustratively shown as **675** through **678**. A pull-up resistor **690** is connected at one end to the junction of resistors **696** and is connected at its remaining end to bias terminal **689**, which has applied thereto a positive d-c operating potential. This particular type of "AND" circuitry, however, does not render a "TRUE" or "FALSE" output signal respectively represented by a zero or a negative potential. In this instance, the "TRUE" or "FALSE" representation is by a positive or a negative output signal which is fed as an input signal to a suitable amplifier which re-establishes the proper voltage levels therefrom; i.e., zero and minus potentials indicative of "TRUE" and "FALSE," respectively.

The operation of the device is as follows: The various resistor values are properly chosen so that, when all of the input signals to terminals **675** through **678** are "TRUE" (ground potential), the output signal at terminal **688** is positive by an amount proportional to the resistor divider network. However, should any one or more of the input signals be "FALSE" (negative), the output signal is also negative by an amount proportional to the resistor divider network, consequently "FALSE" also. "AND" circuit R2 shown in (*h*) is substantially the same as type R1 except for the addition of dropping resistor **691** connected between the junction of resistors **696** and output terminal **688**.

In (*i*) there is shown a transistor type logical "AND" T1 which utilizes a pair of transistors **700** and **709**. Input terminal **675** is connected to the base of transistor **700** through coupling condenser **703**, whereas input terminal **676** is connected to the base of transistor **709** through series connected resistors **711** and **712**. Bias resistor **713** is connected to the junction of resistors **711—712** and is connected to bias terminal **714**, which has applied thereto a suitable positive d-c operating potential. The base of transistor **700** is returned to ground potential through resistor **710** and is also connected to the collector thereof through parallel-connected resistor **704** and condenser **705**. The emitter of transistor **700** is connected to the collector thereof through series-connected resistors **707** and **708**. The junction of resistors **707—708** is returned to a-c ground through condenser **706** and is connected to bias terminal **689** through dropping resistor **690** connected therebetween, bias terminal **689** having applied thereto a suitable negative d-c operating potential. The emitter of transistor **700** is directly connected to the collector of transistor **709**, whereas the emitter of transistor **709** is directly returned to ground potential. Output terminal **688** of the logical "AND" gate is directly connected to the collector of transistor **700**.

In operation, if the input signal at terminal **676** is sufficiently negative, transistor **709** is rendered conductive and thus presents a very low resistance from the emitter of transistor **700** to ground. Consequently, transistor **700** functions in the same manner as a conventional "class-A" inverter type amplifier such that a-c signals of the proper frequency appearing at terminal **675** appear amplified and inverted at output terminal **688**. However, if the input signal at terminal **676** is ground, transistor **709** is rendered non-conductive and thus appears as a very high resistance from the emitter of transistor **700** to ground. Consequently, since no emitter current can flow through transistor **700**, signals appearing at input terminal **675** are not amplified by transistor **700**. Thus it is seen that, when input terminal **676** is negative, a-c signals appearing at terminal **675** appear amplified and inverted at output terminal **688**; when input terminal **676** is ground, a-c signals appearing at terminal **675** do not so appear as amplified and inverted at output terminal **688**.

In (*j*) there is shown a second type of transistor logical "AND" T2, which also utilizes a pair of transistors **700** and **709**. Input terminal **675** is connected to the base of transistor **700** through coupling condenser **703**, whereas input terminal **676** is connected to the base of transistor **709** through series-connected resistor **711**. The emitter of transistor **709** is returned through series-dropping resistor **713** to bias terminal **714** having a suitable positive unidirectional operated potential applied thereto. The emitter of transistor **700** is returned to ground potential through a parallel-connected resistor and condenser network **664** and **666**, whereas the base of transistor **700** is returned through series-dropping resistor **710** to bias terminal **689**, which has a suitable negative unidirectional bias potential applied thereto. The collector of transistor **700** is connected to bias terminal **689** through series-dropping resistor **690** and is also connected to the collector of transistor **709** with both of the collectors of transistors **700** and **709** being connected to output terminal **688**.

In operation, if the input signal at terminal **676** is "TRUE," transistor **709** is rendered non-conductive and appears as a very high resistance from output terminal **688** to ground. Transistor **700** normally is conductively-biased near saturation in such a manner that output terminal **688** is near ground potential. Thus, when positive-going pulses appear at input terminal **675**, they appear amplified and inverted at output terminal **688**. However, if input terminal **676** is "FALSE," transistor **709** is rendered conductive, and its collector current flows through resistor **690** to bias terminal **689**. This collector current is of sufficient magnitude that, even though no collector current were present in transistor **700**, the potential at output terminal **688** is substantially ground. Thus, when positive-going signals appear at terminal **675**, the resulting change in collector current of transistor **700** produces essentially no appreciable change of potential of output terminal **688**. Therefore, when input terminal **676** is "TRUE," positive-going pulses at input terminal **675** appear as amplified and inverted at output terminal **688**; however, when input terminal **676** is "FALSE," output terminal **688** remains essentially at ground potential, even though input positive-going signals appear at terminal **675**.

14. *Flipflop and Multivibrator Building Blocks*

The "flipflop" building block is normally composed of two inverters, as shown in FIG. 49, which are connected in cascade with the input of each one being connected to the output of the other. As shown in section (*a*) of FIG. 49, flipflop of the **F1** type consists of two grounded emitter transistors **722** and **723** having input terminals **717** and **718** respectively connected to the base electrode, and output terminals **719** and **720** respectively connected to the collector electrode of transistors **722** and **723**. The base of transistor **723** is connected to the collector of transistor **722** through parallel connected resistor **734** and condenser **735**, whereas the base of transistor **722** is connected to the collector of transistor **723** through parallel connected resistor **736** and condenser **737**. The base of each of transistors **722** and **723** is returned to ground potential through resistors **729** and **728**, respectively, whereas the collector thereof is respectively connected through dropping resistors **725** and **727** to bias terminals **724** and **726**, which have applied thereto a suitable negative unidirectional operating potential.

In order to describe a mode of operation, it will first be assumed that transistor **722** is non-conductive and transistor **723** is conductive. Thus, when the input signal to terminal **717** goes negative, transistor **722** is now rendered conductive, the internal impedance thereof drops essentially to zero, and consequently the output signal at terminal **719** is thereafter "TRUE" (i.e., at ground potential). In addition, when transistor **722** is rendered conductive, the base of transistor **723** is momentarily driven positive via coupling condenser **735**, and, consequently, transistor **723** is rendered non-conductive. Transistor **723** being non-conductive, the output signal at terminal **720** is "FALSE." A negative bias potential applied from terminal **726** to the base of transistor **722** via resistor divider network **727—736—729** insures that transistor **722** remains conductive when the input signal to terminal **717** is no longer present. Consequently, as terminal **719** is essentially at ground potential, very little base current is supplied to transistor **723** via resistor **734**, and, as a result, transistor **723** remains non-conductive. Thus, the flipflop is now operating in one of its two stable states. It is to be noted at this point that during normal operation, a crystal diode, forming a part of a suitable triggering circuit, to be described later, is usually connected in series with each of input terminals **717** and **718** in such a manner that the states of both of transistors **722** and **723** remain unchanged even though the input signals to terminals **717** and **718**, via their respective triggering circuit diode, effectively changes from "FALSE" back to a "TRUE" condition, respectively. In other words, the only time the states of transistors **722** and **723** change is only when the input signal to one of the terminals **717** and **718** goes from "TRUE" to "FALSE." Now, when the input signal to terminal **718** goes negative, transistor **723** is rendered conductive, the output at terminal **720** is now "TRUE," and the base of transistor **722** is momentarily driven positive and thereby renders transistor **722** non-conductive. Transistor **722** being non-conductive, the output at terminal **719** is "FALSE," and a negative bias is applied from terminal **724** to the base of transistor **723** via network comprising resistors **725**, **734**, and **728**, and coupling condenser **735**, to insure that transistor **723** remains conductive. As before stated, insufficient base current is supplied to transistor **722**, and, consequently, transistor **722** remains non-conductive. Thus, the flipflop is now operating in the other of its two stable states.

To summarize: When the input signal at terminal **717** momentarily goes negative, the flipflop is triggered to one of its two stable states, so that the output at the terminal **719** is TRUE (if not already TRUE) and the output at terminal **720** is FALSE; however, when the input to terminal **718** momentarily goes negative, the flipflop is reset to its initial state, so that the output at terminal **719** is now FALSE and the output at terminal **720** is now TRUE. If, for some reason, the input signals to both of terminals **717** and **718** simultaneously go momentarily negative, the flipflop will change state regardless of the state in which it was initially.

In order to simplify the following description of the computer circuitry as much as possible, input terminal **717** will hereinafter be referred to as the "reference input," input terminal **718** the "prime input," output terminal **719** the "reference output," and output terminal **720** the "prime output."

In (*c*), flipflop of type **F3** is essentially the same as **F1** with the omission of base resistor **728**, whereas, in (*e*), flopflop of type **F5** is essentially the same as **F1** with the omission of base resistor **729**. In (*b*), (*d*), and (*f*), flopflops of type **F2**, **F4**, and **F6** are essentially the same as flipflops **F1**, **F3**, and **F5**, respectively, except for the addition of clamping diodes **731** and **733**, which are utilized to fix the "OFF" level of the collector voltages of transistors **722** and **723**, respectively, as heretofore stated, when the transistor is non-conductive. In (*g*), flipflop of type **F7** is essentially the same as flip-flop **F1** with the omission of coupling condensers **735** and **737**.

With reference to (*h*), there is shown a "single-shot" type flipflop **F8** which is somewhat of the same circuit configuration as **F1**, except that in **F8** there are omitted base resistors **728** and **729**, resistor **736**, prime input terminal **718**, and reference output terminal **719**. However, there have been added a bias resistor **739**, connecting the base of transistor **722** with bias terminal **738**, which has a negative unidirectional operating potential applied thereto, and a coupling condenser **752**, coupling the base of transistor **722** to reference input terminal **717**.

The mode of operation of **F8** is not, however, as straightforward as the mode of operation of the just-described flipflops. In **F8**, as a negative unidirectional bias potential from terminal **738** is normally maintained on the base of transistor **722**, it is normally conductive and thus maintains the base of transistor **723** substantially at ground potential via resistor **734**. The base of transistor **723** being at ground potential, the transistor is therefore normally non-conductive, and, consequently, the prime output at terminal **720** is normally FALSE. However, when the reference input goes from FALSE to TRUE (i.e., at ground potential), the base of transistor **722** is momentarily driven positive by condenser **752**, and thus transistor **722** is immediately cut off and applies a negative potential spike via condenser **735** to turn on transistor **723**. When transistor **723** is rendered conductive, the prime output at terminal **720** immediately goes TRUE. Since the charge on condenser **737** cannot change instantaneously, the base of transistor **722** is driven to some positive potential to maintain transistor **722** non-conductive even though the input signal at terminal **717** is no longer present. After a predetermined time delay, determined primarily by the discharge rate of condenser **737**, a negative bias potential again appears on the base of transistor **722** to render it conductive. When transistor **722** is rendered conductive, the base of transistor **723** is momentarily driven positive and thereafter maintained essentially at ground potential by resistor **734** to turn off transistor **723**, and thus the prime output at terminal **720** is again FALSE.

To summarize: The prime output is normally FALSE; a positive-going reference input gives an immediate TRUE prime output; however, after a predetermined time delay, the prime output changes from TRUE back to FALSE.

In section (*a*) of FIG. 50, there is shown a 50-kc. multivibrator building block (MV1) which comprises two grounded-emitter transistors **722** and **723**, each having its collector electrode respectively connected, through dropping resistors **725** and **727**, to bias terminals **724** and **726**, which have negative unidirectional operating potentials applied thereto. The collector output of transistor **722** is connected to the base input of transistor **723** via coupling condenser **735**, whereas the collector output of transistor **723** is connected to the base input of transistor **722** via coupling condenser **737**. The base of transistors **722** and **723** are respectively connected through dropping resistors **742** and **750** to bias terminals **741** and **749**, each having a negative unidirectional operating potential applied thereto. The base of transistor **722** is connected to the cathode of crystal diode **748**, whose anode is connected to the junction of resistors **744** and **746**. Resistor **744** is connected at its opposite end to bias terminal **743**, which has a positive unidirectional operating potential applied thereto, whereas resistor **746** is connected at its opposite end to bias terminal **745**, which has a negative unidirectional operating potential applied thereto.

The mode of operation of multivibrator (MV1) is as follows: When the circuit is first activated, the positive unidirectional operating potential at terminal **743** is the first to be turned on to apply a positive bias to the base of transistor **722**. Therefore, when the negative bias potentials at **724**, **741**, and **749** are simultaneously turned on, transistor **722** is rendered non-conductive, the reference output at terminal **719** is FALSE, and a negative potential spike is applied to the base of transistor **723** via coupling condenser **735** to cause transistor **723** to be conductive. When transistor **723** is rendered conductive, the prime output at terminal **720** is TRUE, and condenser **737** begins to charge to a potential substantially equal to the negative bias potential at terminal **741**. When condenser **737** is sufficiently charged, the potential on the base of transistor **722** is negative to the extent that transistor **722** is rendered conductive, the reference output goes from FALSE to TRUE, transistor **723** is rendered non-conductive, and consequently the prime output simultaneously goes from TRUE to FALSE. When condenser **735** is sufficiently charged from the negative source at terminal **749**, the base of transistor **723** is then negative to the extent that the transistor is rendered conductive, the prime output goes from FALSE back to TRUE, transistor **722** is rendered non-conductive, and the reference output goes from TRUE back to FALSE. This sequence of events is repeated at a rate determined by the time constants in the charging and discharging paths of coupling condensers **735** and **737**. In the instant computer, multivibrator (MV1) operates at a frequency of 50 kilocycles.

In (*b*), multivibrator (MV2) is essentially the same as (MV1) except that bias resistors **744** and **746** and crystal diode **748** are omitted and replaced by coupling condensers **752** and **753** respectively connected to the base of transistors **722** and **723**. A dropping resistor **754** is connected at one end to the junction of condensers **752** and **753** and is connected at the remaining end thereof to input terminal **751**.

Ignoring for the moment condensers **752** and **753** and resistor **754**, the mode of operation of multivibrator (MV2) is substantially the same as that of (MV1), the frequency of (MV2) being determined, as before, by the time constants in the charging and discharging paths of coupling condensers **735** and **737**. In the instant computer, multivibrator (MV2) operates at a frequency of 400 cycles per second. However, in order to synchronize the change of state of the output of (MV2) with a "clocking" network, to be described hereinafter, a "clocking" signal from such network is fed as the input to terminal **751** of (MV2) and synchronization of the 400-cycle output of (MV2) with the "clocking" network takes place.

Assuming that transistor **723** has just been rendered conductive and transistor **722** has just been rendered non-conductive, the base of transistor **722** is driven positive via condenser **737**, as previously described, and condenser **737** is now charging toward the negative potential at terminal **741**. At the same time, a rectangularly-shaped clock pulse at input terminal **751** causes positive-going and negative-going potential spikes to be superimposed on the potential appearing on the base of transistor **722** in such a manner that, when the potential on the base of transistor **722** is approaching the value sufficient to render transistor **722** conductive, a succeeding negative-going potential spike, caused by the clock pulse at terminal **751**, is sufficient to render transistor **722** conductive. Thus, even though the time period of (MV2) is much longer than that of the synchronizing clock pulse, the change of state of (MV2) takes place at the same time the clock input goes from TRUE to FALSE.

The single-shot flipflop of type **F9**, as shown in (*c*), comprises a pair of grounded emitter transistors **722** and **723**, whose collectors are individually returned to negative bias terminals **724** and **726** through dropping resistors **725** and **727**, respectively. The base of transistor **722** is returned to ground potential through resistor **729** and is also connected to input terminal **717**. The base of transistor **722** is further connected to the collector of transistor **723**, whereas the base of transistor **723** is connected to the collector of transistor **722** through series-connected resistor **755** and condenser **735**, the base of transistor **723** being further connected to a negative bias terminal **749**.

The mode of operation of flipflop **F9** is essentially the same as that of the previously-described flipflop **F8**, and, consequently, a further detailed description thereof is not deemed necessary. The essential difference between **F9** and **F8** is that in **F9** the mode of triggering is such that a momentary negative-going potential at input terminal **717** causes a change of state of the flipflop, whereas in flipflop **F8** a momentary positive-going potential at input terminal **717** causes a change of state thereof.

Flipflop of the **F10** type, as shown in (*d*), is of substantial identical circuit configuration with **F9**, except for the addition of bypass condenser **756**, which bypasses to ground undesirable extraneous signals. Flipflops **F11** through **F13** are essentially the same as the previously described flipflops. The main difference is the addition of a third transistor **758**, which is connected in series with the emitter circuit of transistor **722**. In a sense, transistors **722** and **758** operate together in the form of a transistor type of logical AND circuitry, in that it is necessary for both to be conductive in order for the reference output to be TRUE. As is evident from the previous description, this can happen only when the inputs to both of terminals **717** and **751** are simultaneously negative. Flipflop of type **F14**, shown in (*h*), is essentially the same as the previously described F1 type, except for the omission of base resistors **728** and **729**. Flipflop of type **F15** is essentially the same as flipflop of type **F8**, except that in F15 condenser **735** is removed.

15. *Miscellaneous Building Blocks*

With reference to (*a*) and (*b*) of FIG. 51, there are shown two types of pulse-shaping triggering circuits TC1 and TC2 which are utilized to provide "spike-type" triggering impulses for the various flipflops, in (*c*) and (*d*) are simple resistor-condenser networks, and in (*f*) is shown a resistor-condenser filter and biasing circuit which is used in conjunction with the various push-button controls, all of which are to be further shown hereinafter. As the modes of operation of the various types of circuitry (*a*) through (*d*) and (*f*) are well known to those skilled in the art, a detailed description thereof is not deemed necessary for a full and complete understanding of the present invention.

As previously described with respect to FIG. 7, photocell **160** is utilized in conjunction with the auxiliary timing rack **151** and light source **164** to provide electrical impulses which are indicative of the change of position of the accounting machine amount racks. In section (*g*) of FIG. 51 there is shown a photocell circuit configuration (PCA) utilizing photocell **160** as the controlling element therein. In such circuitry, the anode of photocell **160** is grounded, and its cathode element is connected to output terminal **793**. The cathode of photocell **160** is also connected through dropping resistor **797** to bias terminal **796** having a positive unidirectional operating potential applied thereto. The anode of crystal diode **795** is connected to the junction of output terminal **793**, resistor **797**, and photocell **160**. Its cathode is bypassed to ground potential through condenser **799**. The cathode of diode **795** is also connected to input terminal **792** through dropping resistor **794**, and resistor **800** is connected between input terminal **792** and ground potential. As will be seen later, input terminal **792** is adapted to be selectively "open-circuited" or "short-circuited"; when open-circuited, the resistance between terminal **792** and ground is determined by the value of resistor **800**, whereas, when short-circuited, the resistance between terminal **792** and ground is zero.

The mode of operation of (PCA) is somewhat straightforward. Assuming that terminal **792** is open-circuited and photocell **160** is non-exposed to the radiant energy from light source **164**, the internal resistance of photocell **160** is at a maximum value such that the output potential at terminal **793** is at a maximum positive level. However, when photocell **160** is exposed to the radiant energy from light source **164**, its internal impedance is decreased to a low value, and, consequently, the output at terminal **793** is decreased to a smaller value of positive potential. As will be seen later, the output from terminal **793** is normally fed as an input to a previously described inverter type amplifier which establishes the proper TRUE and FALSE logic signal levels in a well-known manner. The value of resistor **797** is properly chosen so as to be substantially greater than the value of resistor **794**. Consequently, when input terminal **792** is short-circuited to ground potential, the output signal level at terminal **793** drops essentially to zero potential whether or not photocell **160** is exposed to light source **164**.

With reference to FIG. 51*e*, a second type of photocell circuit configuration includes a transistor **650** having its emitter electrode connected to output terminal **652** and having its collector electrode connected through dropping resistor **655** to bias terminal **656**, which has a negative unidirectional operating potential applied thereto. The base electrode of transistor **650** is connected to the junction of the anode of photocell **681** and bias dropping resistor **661**, the opposite end of resistor **661** being connected to bias terminal **662**, which has a positive unidirectional operating potential applied thereto. The cathode of photocell **681** is connected directly to bias terminal **684**, which has a negative unidirectional operating potential applied thereto.

The mode of operation of such a circuit configuration is as follows: When photocell **681** is non-exposed to radiant energy, so that its resistance is at a maximum value, the potential on the base of transistor **650** is at a maximum positive value, and transistor **650** is thereby rendered non-conductive. However, when photocell **681** is exposed to radiant energy, its resistance is reduced to a substantially low value, so that a negative potential from terminal **684** is applied to the base of transistor **650** and thereby renders transistor **650** conductive.

In section (*h*) there is shown a photocell pre-amplifier circuit configuration (PCPA) which comprises a transistor **802** having its collector electrode connected to bias terminal **803**, which has a negative unidirectional operating potential applied thereto. The emitter thereof is directly connected to output terminal **801** and is also connected through series dropping resistor **805** to bias terminal **804**, which has a positive unidirectional operating potential applied thereto. The base electrode of transistor **802** is connected to one electrode of photoresistive element **806**, whose remaining electrode is bypassed to ground potential through condenser **807**. The junction of condenser **807** and photoresistor **806** is connected to the junction of resistors **808** and **809**, remaining end of resistor **809** being grounded and the remaining end of resistor **808** being connected to bias terminal **810**, which has applied thereto a negative unidirectional operating potential.

The mode of operation of photocell pre-amplifier (PCPA), again, is somewhat similar to the mode of operation of (PCA), previously described. When photoresistor **806** is exposed to radiant energy from a suitable light source, its internal resistance becomes a very low value, and, consequently, a negative potential is applied to the base of transistor **802**, so that the output potential at terminal **801** is substantially that which appears on the base thereof, all in the same manner as previously described with respect to emitter-follower E2 shown in section (*b*) of FIG. 47B. However, when photoresistor **806** is not exposed to radiant energy, transistor **802** is virtually non-conductive, and the polarity of the output potential at terminal **801** is positive. As before stated with respect to photocell network (PCA), the output at terminal **801** appears as an input to a suitable amplifier which establishes the proper TRUE and FALSE logical signal levels.

With reference to section (*i*) of FIG. 51, there is illustrated a "peak-detector" circuit configuration PD which is utilized to detect the peaks of negative-going impulses, the reasons for which will become more apparent hereinafter. Such a detector comprises a grounded-emitter transistor **813**, whose collector electrode is connected to output terminal **812** and is also connected through dropping resistor **815** to bias terminal **814**, which has a negative unidirectional operating potential applied thereto. The base input of transistor **813** is returned to ground potential through series-connected resistors **816** and **817**, and is also connected to one end of a parallel-connected condenser and resistor network **818—819**. The opposite end of the condenser-resistor network **818—819** is connected to the anode of crystal diode **820**, whose cathode is coupled to input terminal **811** through condenser **825**. The junction of condenser **825** and crystal diode **820** is connected through series-connected resistors **822** and **823** to bias terminal **824**, which has a positive unidirectional operating potential applied thereto. Finally, a resistor **821** is connected between the junction of resistors **822—823** and the junction of resistors **816—817**.

The relative values of the various resistor components are properly chosen so that a relatively small-valued positive bias appears on the base of transistor **813** to render the transistor normally non-conductive, so that the output at terminal **812** is normally FALSE (i.e., negative). However, a positive bias of slightly greater value appears on the cathode of crystal diode **820** to "back-bias" and thereby render diode **820** non-conductive. Thus, when the leading slope of a negative-going impulse appears at input terminal **811**, crystal diode **820** remains non-conductive until the leading slope reaches a predetermined threshold level, thereby preventing any undesirable low-level noise impulses from entering the system. However, after the leading slope of the negative-going impulse exceeds the negative threshold level, diode **820** is rendered conductive, the incoming impulse is differentiated, and the negative-going potential "spike," derived from the differentiation, is applied to the base of transistor **813** and simultaneously begins to charge condenser **818**. When the negative-going spike is applied to the base of transistor **813**, the transistor is rendered conductive, and the output at terminal **812** is thereafter TRUE. However, when the peak of the negative-going impulse begins to appear at terminal **811**, the slope of the impulse is now much less than before, so that condenser **818** begins to discharge and thereby decreases the magnitude of the negative bias potential on the base of transistor **813**. When the peak of the incoming impulse appears at terminal **811**, its slope being zero, condenser **818** is now fully discharged, and a positive bias is applied to the base of transistor **813** to render the transistor non-conductive. As a result, the output at terminal **812** reverts from a "TRUE" condition back to a "FALSE" condition at a time coincidental with the arrival of the negative peak of the incoming impulse.

During operation of the memory of the present computer, it was found that, when the ambient temperature of a ferromagnetic memory core is increased above normal room temperature, less drive current is needed to "switch" the core from one stable state of magnetic remanence to the other. Conversely, when its operating temperature is decreased, a drive current of greater magnitude is necessary. Consequently, means are provided for automatically increasing and decreasing the drive current to the memory, depending upon whether its temperature has respectively decreased or increased above nominal room temperature.

With reference to the final building block in section (*j*), there is shown a temperature-compensated current regulator (TCCR) which comprises a current control section **840**, a regulator section **841**, and a "dummy load" voltage section **842**. The current control section **840** comprises a transistor **827**, whose base is connected to the anode of a "Zener" diode **833**, the cathode of diode **833** being returned to ground potential. The emitter of transistor **827** is connected to terminal **867** and is also returned to ground potential through resistor **835**. The collector and base electrodes of transistor **827** are respectively returned to bias terminal **832** through resistors **831** and **834**, bias terminal **832** having a suitable negative operating potential applied thereto. Two series-connected thermistors **829** and **830** are connected between terminal **868** and ground potential, whereas resistor **836** is connected between terminal **868** and the anode of "Zener" diode **837**, the cathode of diode **837** being returned to ground potential. A dropping resistor **838** is connected between the junction of resistor-diode **836—837** and bias terminal **839**, which has a suitable negative unidirectional operating potential applied thereto.

Regulator section **841** comprises a transistor **828** having its collector electrode connected to terminal **867**, the base electrode connected to terminal **868**, and the emitter electrode connected to the base of transistor **844**. The emitter of transistor **844** is returned to ground potential through resistor **849** and also through series-connected resistor **850** and potentiometer **851**. The collector of transistor **844** is connected to output terminal **843** through dropping resistor **845**, and is also connected through dropping resistor **846** to the anodes of parallel-connected crystal diodes **847** and **848**. Each cathode of diodes **847** and **848** is connected to terminal **869**, which is bypassed to ground potential by means of filter condenser **860**.

The dummy-load voltage section **842** of the current regulator comprises transistor **852**, whose emitter is connected to terminal **869** and is also returned to ground potential through resistor **859**, which is bypassed by condenser **866**. The collector of transistor **852** is connected through dropping resistor **857** to bias terminal **856**, which has a suitable negative unidirectional operating potential applied thereto. The base of transistor **852** is connected to the emitter of transistor **853** and is also connected through dropping resistor **855** to bias terminal **854**, which has a negative unidirectional operating potential applied thereto. The collector of transistor **853** is directly connected to bias terminal **865**, which has a negative unidirectional operating potential applied thereto, whereas its base is connected to the movable arm of potentiometer **861**. One end of potentiometer **861** is returned to ground potential through resistor **862**, and its opposite end is connected through resistor **863** to bias terminal **864**, which has a negative unidirectional operating potential applied thereto.

During operation, thermistors **829** and **830** are physically positioned in close proximity to the memory matrix, and, being in such contiguous relationship with respect to the ferrite cores of the memory, the resistance of each thermistor either increases or decreases in response to a corresponding decrease or increase of the operating temperature of the memory cores. "Zener" diode **833** maintains a regulated bias potential of approximately −6 volts on the base of transistor **827**, and, as transistor **827** is being operated as an emitter follower type amplifier, a regulated collector bias potential of approximately −6 volts is also applied to the collector of transistor **828**, which is likewise operating as an emitter follower type amplifier. "Zener" diode **837** maintains a regulated bias potential of approximately −6 volts at the junction of resistors **836** and **838**. Consequently, the variation of bias potential on the base of amplifier **828** is dependent solely upon the combined values of the resistances of thermistors **829** and **830**, which, in turn, are dependent upon the operating temperature of the computer memory. Therefore, it is evident that the negative bias voltage on the base of transistor **844** is likewise dependent primarily upon the operating temperature of the computer memory.

During operation of the computer, there are times when there is no information either being "stored" or being "read out" of memory. Consequently, there are times when transistor **844** is not supplying any drive current to the memory. Due to the fact that the "alpha cut-off" characteristic of transistor **844** is less than one megacycle, its switching speed is no sufficient for the desirable switching requirements of the ferrite cores. Therefore, transistor **844** is maintained conductive at all times, and its output current is switched to the ferrite cores, when needed, or is adsorbed by dummy load **842**, as the situation dictates.

The two emitter follower type amplifiers **852** and **853** of dummy load **842** function together essentially as a 600 ma. low impedance bias supply and provides a bias potential of approximately −8.5 volts at terminal **869**, depending upon the setting of potentiometer **861**, which adjusts the magnitude of the output voltage. Under the condition in which the various memory "driver" and "grounder" amplifiers, which are to be described in detail hereinafter, are non-conductive, diodes **847** and **848** are biased in the forward direction by the negative potential at terminal **869**, and, consequently, the drive current, delivered by transistor **844**, passes through diodes **847** and **848** and is adsorbed by the dummy load. However, when the memory drivers and grounders are rendered conductive, the collector bias voltage on transistor **844** becomes approximately −10 volts and, consequently, is then more negative than the bias voltage at terminal **869**. Therefore, diodes **847** and **848** are rendered non-conductive and prevent any further drive current from flowing therethrough. As a result, all of the drive current supplied by transistor **844** flows from terminal **843** and is delivered to the memory drivers and grounders.

The emitter bias on transistor **844** is adjusted by potentiometer **851** so that a "half-select" drive current of approximately 170 ma. is delivered thereby at a core operating temperature of 30 degrees centigrade. However, due to the action of thermistors **829** and **830** as previously described, the base bias on transistor **844** is automatically adjusted so that a range of "half-select" drive currents of from approximately 145 ma. at a temperature of 50 degrees centigrade to approximately 210 ma. at a temperature of 20 degrees centigrade is supplied to the memory cores via the driver and grounder amplifiers. Due to the fact that transistor **844** is operating as a Class-A amplifier, a constant current is supplied thereby at a given temperature even with changing load conditions thereon. Thus it is seen that transistor **844** is continually supplying a constant current for a given temperature, which current is automatically switched from the memory drivers and grounders to the dummy load, and the magnitude of current not varying substantially with a switching of loads.

In the instant computer, it is necessary to utilize three such constant-current regulators: one is utilized to supply half-select drive current to the ferrite cores located in addresses "A" and "B" of the memory; one is utilized to supply half-select drive current in the "X" direction through the memory cores located in each of the forty rows; and the other is utilized to supply half-select drive current in the "Y" direction through memory cores in addresses ϕϕ through 99, all of which will be described in detail hereinafter. However, to avoid the necessity of constructing three identical temperature-compensated current regulators, as just described, it has been found expedient to construct but a single current regulator which is substantially identical to the one shown and described, with the exception that three identical regulator sections **841** are utilized instead of one, and all three are connected in parallel at terminals **867, 868,** and **869**. Output terminal **843** of each of the three regulator sections is connected to one of lines "CRA," "CRX," and "CRY" shown in FIGS. 60 and 66. However, for the sake of simplicity of description and illustration, each of the temperature-controlled current regulators **7000, 7001,** and **7002** is illustrated as separate units even though, in fact, all are part of a composite unit as just described.

16. *Magnetic Core Memory*

As previously mentioned, binary information is magnetically stored in the computer in a ferrite core memory. The core memory consists of a multiplicity of toroidally-shaped cores which possess a substantially square magnetic hysteresis-loop characteristic. Each core has an outside diameter of .080 inch and an inner diameter of .050 inch, and is approximately .025 inch thick. It is well known to those skilled in the art that, due to the fact that the ferrite core possesses a substantially square hysteresis loop characteristic, it therefore has two stable states of magnetic remanence, each of which is respectively indicative of a binary "one" and a binary "zero." If a current-carrying conductor is threaded through the hole in the core, current in one direction through the conductor causes the core to be permanently magnetized in one direction, whereas current flow in the opposite direction through the conductor causes the core to be permanently magnetized in the opposite direction. Thus, a core is permanently magnetized in one direction or the other, depending upon the direction of current flow in the conductor threaded through the hole in the core. To summarize, when a core is magnetized in one direction, it represents a binary "one," and when the core is magnetized in the other direction it represents a binary "zero." Four cores are utilized as a group to collectively represent a single decimal digit, although, with the use of the binary form of notation, each group of four cores could possibly represent a number from "0" to "15." However, in the present computer, only the numbers "0" through "9" are actually stored in the memory.

In order to reverse the magnetic remanence of the core from a binary one to a binary zero notation, or from a binary zero to a binary one by reversing the direction of magnetization thereof, a magnetomotive force of approximately three hundred and sixty milliampere turns is required at 25 degrees centigrade. This is obtained by the equivalent of passing 360 milliamperes of current through a single conductor threaded through the aperture of the core. A magnetomotive force of one half this value—i.e., 180 milliampere-turns—does not appreciably disturb the magnetic remanence of the core.

After a core has been magnetized in a predetermined direction to represent a binary "one" or a binary "zero," it remains in that state of magnetic remanence indefinitely. The magnetic remanence state of the core is reversed only by a "full-amplitude" current impulse passing through the conductor threaded through the core. However, at certain predetermined times, it is necessary to "read" or determine the state of magnetic remanence of each core. Therefore, an additional conductor or "sense wire" is also threaded through the core aperture, so that, whenever the magnetic state of the core is changed from a binary "zero" to a binary "one" representation, or from a binary "one" to a binary "zero" representation, a voltage impulse is induced in the "sense wire" indicative of the reversal of magnetic remanence of the core. Thus it is seen that an impulse is induced in the "sense wire" only when the magnetic state of the core is reversed. In order to determine whether a particular core has been previously magnetically conditioned to represent a binary "one" or a binary "zero," a full-amplitude current of approximately 360 milliamperes is sent through the first-mentioned conductor in a direction to "set" the core to a binary "zero" representation. However, if the core had been previously set to a binary "zero" representation, a voltage impulse is not induced in the "sense wire." However, if the core was previously in a binary "one" representation, its magnetic remanent state is reversed, and, consequently, a voltage impulse is induced in the "sense wire" indicative of the reversal of the magnetic state of the core. After the magnetic state of the core is sensed, however, its magnetic state is immediately thereafter indicative of a binary "zero" representation regardless of its initial state. It is therefore necessary to resore the core to its original magnetic state normally by sensing a full-amplitude current impulse through the first-mentioned conductor to set the core back to a binary "one" representation if the core was previously in that particular state.

The core memory, utilized by the instant computer, has a capacity of one hundred ten-decimal-digit numbers which are hereinafter termed "words." Due to the fact that a negative word is stored in the memory as a positive complement, there is no need for a binary sign bit, as is conventional in most computers. A word is selectively stored in the memory in one of a hundred different locations which are called "addresses" and which are numbered from ϕϕ through 99. In addition to the one hundred normal addresses, there are two additional addresses labeled "A" and "B." The words stored in addresses ϕϕ through 99 not only represent arithmetic data, but also represent "instructions" which are utilized by the computer to dictate the sequence of operations therein. The words previously stored in addresses ϕϕ through 99 of the memory may be changed by instructions given by the computer; however, the words located in addresses A and B are used by the computer to store intermediate results during arithmetic computations and cannot be changed directly by instructions.

With reference to FIGS. 52*a* and 52*b*, there is schematically shown the ferrite core memory utilized by the present computer. Such a memory comprises four thousand and eighty type N–400–080 ferrite cores at present manufactured by the assignee of the instant application.

The ferrite cores of the memory are selectively arranged in a pattern of rectangularly-shaped configuration having one hundred two vertical columns and forty horizontal rows thereof. The first one hundred columns, as viewed from left to right, respectively represent memory address locations $\phi\phi$ through 99, whereas the two remaining rightmost columns respectively represent address locations A and B. In each of the columns, from the topmost to the bottommost core, the forty cores are arranged in a succession of ten groups, with each group containing four cores. Each group of four cores is representative of a decimal digit of a predetermined order, whereas all of the groups of a particular column collectively represent all of the orders of a word ten decimal digits in length. Each of the ten successive decimal digits of the word located in any address is respectively identified as the "first-order" digit through the "tenth-order" digit, where the first order is the lowest or "penny" digit and the tenth order is the highest order digit of the number or word. The four cores used to store the first-order digit of a word which is residing in an address in the memory are located in the first row through the fourth row, counting from bottom to top. The four cores used to store the second-order digit of the word are located immediately above in the fifth through the eighth rows; in the drawing, only the fifth and eighth rows are shown, rows six and seven being omitted for the sake of simplicity. The four cores used to store the third-order digit are located immediately above in rows nine through twelve, where rows nine and twelve are the only ones illustrated, rows ten and eleven being omitted for the reasons just mentioned. The location of each higher-order digit progresses upward, as just described, so that the tenth or highest-order digit is located at the top of the column in the last four rows shown; i.e., thirty-seven through forty. Therefore, it is seen, the maximum storage in each memory address is a ten-digit number which will hereinafter be called a "word," the lowest-order digit thereof being located at the bottom of each address and the highest-order digit thereof being located at the top of each address.

As the instant computer utilizes the well-known "8421 binary-coded-decimal-digit" system of numerical representation, each of the four cores in a group making up a particular order decimal digit of the word is known as a "binary-bit." The binary-bits are consecutively labeled (*a*), (*b*), (*c*), and (*d*), where bit (*a*) is the lowest-order of representing a particular order numerical digit havin rows #1, #5, #9, and #37, and bit (*d*) is the highest-order bit represented by the magnetic state of the cores located in rows #4, #8, #12 . . . and #40. As previously mentioned, each group of four cores is capable of representating a particular order numerical digit having a value from "0" through "9." For example, to represent a decimal digit having a value of "0," all of the four cores in a group are selectively set to binary bit representations (0000); for a decimal digit having a value of "1," the cores are set to (0001) binary bit representations; a decimal digit having a value of "2," the cores are set to bit representations (0010); a "3" is represented by (0011); "4" by (0100); "5" by (0101); "6" by (0110); "7" by (0111); "8" by (1000); and finally, to represent a decimal digit having a value of "9," the cores are sequentially set to binary bit representations (1001).

To illustrate, supposing that the word 0000000695 is stored in memory at address $\phi\phi$. In the leftmost vertical column, the four lowermost cores, indicated by reference numerals **885** through **888**, are respectively conditioned to collectively represent binary bits (0101) indicative of the first-order digit "5." In other words, core **885**, representative of binary bit "*a*," is set to binary "1"; core **886**, representative of binary bit "*b*," is also set to binary "0"; core **887**, representative of binary bit "*c*," is set to binary "1"; and, finally, core **888**, representative of bit "*d*," is set to binary "0" representation. Collectively, the magnetic state of cores **885** through **888** represents the penny digit "5." The next group of four cores, located directly above group **885** through **888**, are set to binary bit representations (1001) and collectively represent the second-order decimal digit "9." The next group of four cores are set to binary bit representations (0110), collectively representing the third-order decimal digit "6." All of the remaining cores in address $\phi\phi$ are individually set to binary "0" representations. Thus, with the non-significant decimal digits removed, the word becomes "695."

In one mode of operation, when a column of cores is to be "read out" to determine the word stored in that particular memory address, the cores are sequentially read, one at a time, starting with the core at the bottom of the address and concluding with a core at the top of the address. For example, in address $\phi\phi$, core **885** is read first to produce the binary value of bit "*a*" of the first-order digit; core **886** is read next to produce bit "*b*" of the first-order digit; core **887** is read next to produce bit "*c*"; and core **888** is next read to produce bit "*d*" of this digit. The next four bits, representing the second-order digit, are successively read out, and the remainder of the bits for successive higher-order digits are thereafter sequentially read out in the same sequence, concluding with bit "*d*" of the tenth-order decimal digit.

After a core has been read out, it is generally desired to restore the core to its original magnetic state as before being read. That is, after each bit-representing core is read out, if the core was originally set to represent a binary "1," the core is returned to the "1" state following reading thereof. Therefore, following the reading of each bit, a predetermined unit of time is permitted to lapse before the next bit is read out; it is during this time lapse that the core, just read, is returned to the "1" state if it had previously been set to that state.

For the sake of simplicity of the following description, the vertical or column orientation of the memory will hereinafter be termed the "Y" direction, and the horizontal or row orientation will hereinafter be termed the "X" direction.

Even though, in reality, each of the memory cores has five conductors threaded therethrough for purposes to be more fully described hereinafter, only three of these conductors are to be considered at this time in describing a mode of operation of the memory in terms of the electrical schematic diagram thereof shown in FIG. 52. The first of the three just-mentioned conductors is a "sense winding" **871**, which starts at terminal **872** at the lower left corner of the memory and is alternately threaded through each row of cores bounded by address $\phi\phi$ through 99. As shown, sense winding **871** is successively threaded from left to right through all of the cores of the first row, is threaded from right to left through the cores of the second row, from left to right through the third row, and alternately continues on from row to row. Finally, sense winding **871** is threaded from right to left through the topmost or fortieth row of cores, and terminates at terminal **873**. All of the cores in address $\phi\phi$ are threaded in the "Y" direction by common current-carrying conductor **878**, which is shown positioned to the left within the apertures thereof. Also, all of the cores located in the first row bounded by address $\phi\phi$ and address "B" are threaded in the "X" direction by a common current-carrying conductor **879**, which is shown centrally positioned within the apertures thereof, address "B" being the rightmost one of the memory addresses.

It will now be assumed that it is desired to read out the contents of address $\phi\phi$. Thus, in order to read bit "*a*" of the first-order decimal digit of the word stored in address $\phi\phi$, a half-amplitude or "half-select" current impulse is delivered to conductor **878**, and, simultaneously therewith, a half-select current impulse is present in conductor **879**. The two half-select impulses are in such directions that the magnetomotive forces associated therewith are additive in the region of core **885**, with the resultant force being of sufficient magnitude to magnetically saturate the core in a direction indicative of a binary "zero." Due to the fact that core **885** is the only core in the entire memory that has received the necessary magnetomotive force to cause a reversal of its magnetic state, all of the remaining cores essentially remain magnetically undisturbed. It is known that whenever the state of magnetic remanence of any one of the memory cores is reversed, a voltage impulse is induced thereby in sense winding **871**. However, due to the fact that in the present memory only one core at a time is sensed, there is no ambiguity as to which core was responsible for the impulse induced in the sense winding.

Therefore, if core **885**, whose magnetic state is indicative of bit "*a*" of the first-order digit of the number, is storing a binary "1," its magnetic state is reversed by the coincidental "X" and "Y" read impulses, and, consequently, a voltage impulse appears between terminals **872—873** of the sense winding **871**, indicating that core **885** was previously storing a binary "1." If core **885** had previously been storing a binary "0," however, a voltage impulse does not appear across the output terminals of the memory sense winding, thus indicating a binary "0" storage.

After core **885** has been read, its magnetic state is thereafter indicative of a binary "0," as previously mentioned. Thus, if the magnetic state of the core was indicative of a binary "0" before being read, there is no need for "resetting" the core after it is read. However, if the state of the core was indicative of a binary "1" before being read, it is often necessary to "reset" the core to a binary "1" representation after it has been read. To do this, the directions of both of the current impulses applied through the "X" and "Y" conductors are effectively reversed simultaneously. This causes a corresponding reversal of the additive magnetomotive force in the vicinity of the core, which reverses the magnetic state thereof. Again, as only a half-select current impulse is applied to each of the "X" and "Y" conductors, core **885** is the only core in the entire memory that is magnetically affected thereby.

Now that core **885** has been read and afterwards reset to its initial state, core **886** is read next to determine the binary value of the bit "*b*" of the first-order digit of the word. Again, a half-select current impulse is applied in the "Y" direction to conductor **878**, and, essentially, a half-select current impulse is simultaneously applied in the "X" direction to the conductor of the second row corresponding to **879**. As before, both half-select currents are in such directions to effect storage of a binary "0" in core **886**. If the state of core **886** was indicative of a "1" before being read, the current in each of the "X" and "Y" conductors threaded therethrough is reversed to reset core **886** to "1" after it is read. This reading and writing sequence of operation is sequentially continued from core to core until all forty cores of address ϕϕ have been read and thereafter restored to their respective magnetic states.

The just-described combined reading and writing cycle of operation of each core is hereinafter called a "read-write" cycle. The time required to complete a read-write bit cycle in the instant computer is approximately 40 microseconds. During the first 30 microseconds of the read-write cycle, the core is set to a binary "0" representation; during the remaining 10 microseconds of the cycle, the core is often reset to a binary "1" representation if it was originally in that state. However, if the core was originally in a binary "0" state before being read, there is no resetting operation necessary during the last 10-microsecond interval. Therefore, to read out an entire ten-digit word from an address in memory requires a total time of 40×40, or 1,600, microseconds.

Due to the fact that, in the present memory, it is not desired to reverse the current flow in a conductor, an additional conductor is individually threaded in the "X" direction through each of the rows of cores. This is illustrated by conductor **880**, shown threaded through the cores of the first row and disposed parallel with respect to conductor **879**, previously described. By the same token, an additional conductor is also individually threaded in the "Y" direction through the cores of each of memory addresses ϕϕ through 99, and also addresses "A" and "B." This conductor is illustrated by conductor **881**, shown threaded through the cores of address ϕϕ and disposed parallel with respect to conductor **878**, previously described. Thus, two conductors are threaded in the "X" direction and two conductors are threaded in the "Y" direction through each of the cores of the memory; each of the four wires transmits current in only one direction, as will be shown hereinafter. The fifth wire through memory cores in addresses ϕϕ through 99 is sense wire **871**, heretofore described. A separate sense winding **891**, which originates at terminal **892** and terminates at terminal **893**, is alternately threaded, in one direction and then the other, through the forty rows of cores making up memory addresses "A" and "B" in the same manner as memory sense winding **871**.

Before attempting to give a more detailed description of the memory, it is to be appreciated that each of the rows of cores, from the first to the fortieth row, is connected in a same circuit configuration with respect to the others. Likewise, each of the columns of cores, from addresses ϕϕ through 99 and including addresses "A" and "B," is also of the same circuit configuration with respect to the others. Thus, a description and full comprehension of the mode of operation of address ϕϕ with respect to the first row of cores for producing bit "*a*" of the first-order digit of the word stored in address ϕϕ should suffice for the remaining bits of that digit, and also the remaining digits of that word. As the mode of operation of address ϕϕ is exactly the same as that of the remaining memory addresses, a further description of the remaining addresses, again, would result only in unnecessary repetition. It is also to be appreciated that, in an attempt to simplify the schematic representation of the memory as shown in FIG. 52, addresses ϕϕ through ϕ9 are consecutively shown, reading from the extreme left to the right; addresses 1ϕ and 19 are shown next, with addresses 11 through 18 being omitted, as indicated by the vertical "break" in the drawing between addresses 1ϕ and 19; following in sequence are addresses 2ϕ, 29, 3ϕ, 39, 4ϕ, 49, 5ϕ, 59, 6ϕ, 69, 7ϕ, 79, 8ϕ 89, and 9ϕ through 99, and finally addresses "A" and "B"; addresses 21–28, 31–38, 41–48, 51–58, 61–68, 71–78, and 81–88 are omitted. Starting from the bottom of the drawing, the first four rows are shown which make up the low-order digit of the word stored in memory; rows #5 and #8 are shown next, with rows #6 and #7 omitted as indicated by the horizontal "break" in the drawing between rows #5 and #6; following in sequence are rows #9, #12, #13, #16, #17, #20, #21, #24, #25, #28, #29, #32, #33, #36, and #37 through #40; rows #10–#11, #14–#15, #18–#19, #22–#23, #26–#27, #30–#31, and #34–#35 are omitted.

With particular attention directed to the first row of cores, on emergence from the rightmost end of the first row of cores, conductor **880** is connected to conductor **882**, which passes below the first row and is connected at its opposite end to line XD*a*, located at the bottom left corner of the memory. The leftmost end of conductor **879** also is connected to line XD*a*, and its opposite end is connected to the cathode of one of crystal diodes **876**, its anode being connected to line XGϕ; and, finally, the remaining left end of conductor **880**, on emergence from the leftmost end of the first row of cores, is connected to the cathode of one of crystal diodes **877**, whose anode is connected to line (XGϕ)′.

Whenever line XD*a* is activated, that line is effectively connected to a source of potential of approximately —12 volts. Thus, if line XGϕ is simultaneously activated therewith, that line is effectively connected to terminal **843** of a temperature-compensated current regulator, of the type previously described with respect to FIG. 51*g*, so that a half-select drive current impulse flows from line XGϕ through diode **876** from right to left through conductor **879** and out at line XD*a*. However, if line (XGϕ)′ is simultaneously activated instead of line XGϕ, line (XGϕ)′ is effectively connected to terminal **843** of the current regulator, so that a half-select current impulse from line (XGϕ)′ flows through diode **877**, thereafter flows from left to right through the upper loop of conductor **880**, and is returned to line XD*a* by way of conductor **882**. Consequently, it is seen that, when line XD*a* is activated, the half-select drive impulse flows in one of two directions through the cores of the first row, depending upon whether XGϕ or (XGϕ)′ was simultaneously activated with line XD*a*.

The electrical connections for rows #2 through #4, respectively representing bits "*b*," "*c*," and "*d*" of the first-order digits of the words stored in memory, are exactly the same as for row #1, the anodes of the four crystal diodes **877** each being connected to line (XGϕ)′ and the anodes of the four crystal diodes **876** being connected to line XGϕ. When line XD*b* is activated, the line is effectively connected to a potential source of —12 volts, as was line XD*a*. Thus, when line XGϕ is activated simultaneously with XD*b*, the half-select current impulse flows from right to left through the cores of the second row; if (XGϕ)′ is activated instead of XGϕ, a half-select current impulse flows from left to right through the cores of the second row. By the same token, if XD*c* and XGϕ are simultaneously activated, a half-select current impulse flows from right to left through the third row of cores; a simultaneous activation of XD*c* and (XGϕ)′ effectively causes a reversal of current flow through the third row of cores. Finally, XD*d* and XGϕ, or XD*d* and (XGϕ)′ operate together to effectively send a half-select impulse from right to left or from left to right, respectively, through the fourth row of cores in the same manner as just described.

As just described, the first four of diodes **877** (i.e., the ones respectively associated with rows #1 through #4) have their anodes connected together and returned to line (XGϕ)′. The second group of four diodes **877**, associated with rows #5 through #8, of which only rows #5 and #8 are illustrated, all have their respective anodes connected to line (XG**1**)′. The anodes of the third group of diodes are each connected to line (XG**2**)′, the anodes of the fourth group being connected to (XG**3**)′, and so on up the column, with the anodes of the tenth group of diodes **877** being connected to line (XG**9**)′. Also, as shown, the left end of each of the conductors corresponding to **879** and **882** associated with rows #1, #5, #9, #13 . . . #33, and #37, which rows successively correspond to bit "*a*" of each successive-order decimal digit of the word stored in memory, are connected together and returned to line XD*a*. Likewise, although not fully illustrated, the left end of each of the conductors corresponding to **879** and **882** but associated with rows #2, #6, #10, #14 . . . #34, and #38, which rows successively correspond to bit "*b*" of each successive-order decimal digit of the word in memory, are connected together and returned to line XD*b*. By the same token, although not fully illustrated, the left end of each of the conductors corresponding to **879** and **882** but associated with rows #3, #7, #11, #15 . . . #35, and #39, which rows successively correspond to bit "*c*" of each successive-order decimal digit of the word in memory, are connected together and returned to line XD*c*. And finally, the left end of the conductors corresponding to **879** and **882** but associated with rows #4, #8, #12, #16 . . . #36, and #40, which rows successively correspond to bit "*d*" of each successive-order decimal digit of the word in memory, are connected together and returned to line XD*d*.

With respect to the vertically-disposed column of diodes located to the right of address "B" and indicated as **876**, the first bottom group of four diodes thereof associated with rows #1 through #4 each has its anode connected to line XGϕ; the second group of four diodes **876** located directly above the first group have their anodes returned to common line XG**1**; the anodes of the third group of diodes **876** are connected to line XG**2**, and so on, with the anodes of the last four diodes **876** at the top of the column and associated with rows #37 through #40 being connected to common line XG**9**.

Referring to the lowermost row of diodes **874** located below row #1, counting from left to right, the anodes of the first ten of diodes **874** respectively associated with addresses $\phi\phi$ through ϕ9 are connected to line YGϕ. With respect to the next group of diodes **874** located immediately to the right of the first group of diodes and respectively associated with addresses 11 through 19, of which only addresses 11 and 19 are illustrated and addresses 12 through 18 are omitted, their anodes are connected to line YG**1**; the anodes of the third group of diodes **874** respectively associated with addresses 20 through 29 are connected to line YG**2**; and so on, with the anodes of the last group of ten diodes **874**, respectively associated with addresses 9ϕ through 99, being connected to line YG**9**. With respect to the upper row of diodes **875**, which are located directly above row #**40**, the first group of ten diodes which are respectively associated with addresses $\phi\phi$ through 99 have their anodes connected to line (YGϕ)′ in the same manner as the lower row of diodes **874**. The anodes of each successive group of ten diodes **875**, counting from left to right, are respectively connected to lines (YG**1**)′ through (YG**9**)′.

To complete the electrical connections to memory addresses $\phi\phi$ through 99, the upper ends of vertical conductors corresponding to **878** and **883**, which are respectively associated with addresses $\phi\phi$, 1ϕ, 2ϕ, 3ϕ, . . . 8ϕ, and 9ϕ, are connected to line YDϕ. The upper ends of vertical conductors corresponding to **878** and **883**, which are respectively associated with addresses 01, 11, 21, 31, . . . 81, and 91, are connected to line YD**1**. Only addresses ϕ1 and 91 of this group are illustrated, the remaining addresses of this group being omitted for simplicity purposes. Likewise, the upper ends of vertical conductors **878** and **883**, which are respectively associated with addresses ϕ2, 12, 22, 32, . . . 82, and 92, are connected to line YD**2**. This sequence of connections continues from lines YD**3** through YD**9**, where YD**9** is connected to vertical conductors corresponding to **878** and **883**, which are respectively associated with addresses ϕ9, 19, 29, 39, . . . 89, and 99.

To summarize: Selective energization of line YDϕ essentially selects all addresses whose low-order digit is a "0"; YD**1** essentially selects all addresses whose low-order digit is a "1"; YD**2** essentially selects all addresses whose low-order digit is a "3"; and so on from YD**3** through YD**9**, where YD**9** essentially selects all addresses whose low-order digit is a "9." Forgetting about "prime-notations" for the present, selective energization of line YGϕ essentially selects all addresses whose high-order digit is a "0"; YG**1** essentially selects all addresses whose high-order digit is a "1"; YG**2** essentially selects all addresses whose high-order digit is a "2"; and so on from YG**3** to YG**9**, where selective energization of line YG**9** essentially selects all addresses whose high-order digit is a "9." Thus, suppose that it is desired to select a particular address; say address 9ϕ. To accomplish this, YDϕ is energized corresponding to the low-order digit "0" of the address, and, simultaneously therewith, YG**9** is energized corresponding to the high-order digit "9" of the address. Thus, when YDϕ and YG**9** are simultaneously energized, a binary "1" representative half-select current flows upwardly from line YG**9** through all of the cores in address

9ϕ and out at line $YD\phi$. To select address 99, YD9 and YG9 are energized simultaneously.

As far as "prime notations" are concerned, it is to be pointed out at this time that throughout the electrical circuitry of the computer signal lines bearing a "prime" notation—i.e., $(XG\phi)'$ and $(YG\phi)'$ et al.—essentially have diametrically opposite states or energization conditions to their respective "primeless" counterparts; i.e., $XG\phi$, $YG\phi$, et al. For example, lines $(YG\phi)'$ through $(YG9)'$ are effective to selectively send a half-select drive-current impulse downwardly through any one of addresses $\phi\phi$ through 99 to "half-select" a particular address to a binary "0" representation, whereas lines $YG\phi$ through YG9 are effective to send a half-select impulse upwardly through any one of the same respective addresses to "half-select" that particular address to a binary "1" representation. It is also to be appreciated at this point that, when the state of a signal line bearing a "prime" notation is de-energized or is FALSE, the "primeless" signal line counterpart is respectively "energized" or is TRUE, and vice versa. In other words, when one is energized, the other is de-energized; when one is TRUE, the other is FALSE, etc.

In the "X" direction of the memory, line XD*a* selects those ten rows corresponding to bit "*a*" of each order decimal digit of the word; line XD*b* selects those ten rows corresponding to bit "*b*"; line XD*c* selects the ten rows corresponding to bit "*c*"; and, finally, line XD*d* selects the final ten rows corresponding to bit "*d*" of each order decimal digit of the word. Lines $(XG\phi)'$ through $(XG9)'$ are effective to send a half-select drive-current impulse from left to right through any one of rows #1 through #40 to "half-select" a particular row to a binary "0" representation, whereas lines $XG\phi$ through XG9 are effective to send a half-select impulse from right to left through any one of the same respective rows to "half-select" that particular row to a binary "1" representation.

Thus, combining "X" and "Y" selection of the memory, in order to store a binary "1" representation in a core, say core **885** as an example, which is located at the junction of row #1 and address $\phi\phi$, lines $YD\phi$, $YG\phi$, XD*a*, and $XG\phi$ are all simultaneously energized. To store a binary "0" representation therein, lines $YD\phi$, $(YG\phi)'$, XD*a*, and $(XG\phi)'$ are all effectively energized simultaneously. To store binary information in core **886**, line XD*b* is energized instead of line XD*a*. The manner of "X" and "Y" selection of addresses "A" and "B" is exactly the same as the manner of selection of memory addresses $\phi\phi$ through 99. Consequently, further detailed description thereof is not deemed necessary.

Before proceeding further with the description of the logical control circuitry of the computer, it is to be pointed out that an attempt has been made to simplify and thus alleviate the inherent complexity of the schematic representation in order to facilitate a full and complete understanding of the circuitry, both as to its organization and as to its mode of operation. For example, throughout the drawings, all input lines are appropriately labeled and are positioned to the leftmost side of the various building blocks previously described, and the various output lines thereof also are appropriately labeled and are positioned to the rightmost side of the building blocks. However, for illustrative purposes only, but a selected few of the input and output lines are actually shown connected. It is, of course, to be understod that, in order to obtain a full and complete electrical circuit diagram of the computer, it first is necessary to substitute the appropriate type of building block circuitry (taken from FIGS. 46 through 51) for each of the correspondingly-labeled building blocks which are logically illustrated throughout the drawings. Afterwards, all like-labeled lines (whether input lines, output lines, or otherwise) are to be connected together, thus forming a complete circuit diagram from the schematic representation thereof.

17. *X-Drivers*

With reference to the lower right-hand section of FIG. 60, there are schematically illustrated four "X-drivers," which are utilized to selectively energize input lines XD*a* through XD*d*, previously described in connection with the core memory shown in FIG. 52. Each of the X-drivers comprises a two-input logical AND of type "**R1**," an inverter amplifier of type "**I12**," and an emitter follower amplifier of type "**E3**," each connected in cascade with respect to the others. More specifically, the X-driver for bit "*a*" includes logical AND **1108**, inverter **4237**, and emitter follower **4250**; the X-driver for bit "*b*" comprises logical AND **1109**, inverter **4238**, and emitter follower **4251**; the X-driver for bit "*c*" comprises logical AND **1110**, inverter **4239**, and emitter follower **4252**; and, finally, the X-driver for bit "*d*" comprises logical AND **1111**, inverter **4250**, and emitter follower **4253**.

During a read-write cycle of operation for reading a word from an address in memory, line SMC goes TRUE for 1600 microseconds to allow line XDW to effectively "condition" the selected X-driver to effectively be turned "ON" at, and for, the proper amount of time. At 40 microseconds intervals, lines BaM through BdM sequentially go TRUE for a period of 40 microseconds each, and then respectively go FALSE. Consequently, at the beginning of the read-write cycle, output line XD*a* is energized, and a potential of approximately -12 volts appears thereon. Line XD*a* stays energized for a maximum period of 40 microseconds and then is de-energized. Forty microseconds after line XD*a* was first energized, output line XD*b* is energized for a maximum period of 40 microseconds, and then is de-energized. Forty microseconds after line XD*b* was first energized, line XD*c* is energized for a maximum period of 40 microseconds. Finally, 40 microseconds after line XD*c* was first energized, line XD*d* is energized for a maximum period of 40 microseconds, lines XD*a* through XD*d* of FIG. 60, of course, being identical to lines XD*a* through XD*d* of FIG. 52, as previously stated. Thus it is evident, as a maximum total time of 160 microseconds is required to read each digit out of memory, that a maximum of 1600 microseconds is required to read a ten-digit number out of a particular address in memory.

18. *Bit-Counter*

The bit-counter, logically shown in the upper portion of FIG. 62, includes two type "**F2**" flipflops **6041** and **6042**, connected as a scale-of-four binary counter, and is utilized to select the binary bit of the word to be read out of an address in memory during a read-write cycle, as illustrated in the block diagram of FIG. 45. When the computer is first turned "ON," or when pushbutton RS1 (FIG. 76) is actuated, flipflops **6041** and **6042** are unconditionally set TRUE by reset line (RS)' going FALSE. That is, the states of flipflops **6041** and **6042** are such that reference output lines BC*a* and BC*b*, respectively therefrom, are TRUE, and prime output lines (BC*a*)' and (BC*b*)' are both FALSE.

The bit-counter usually counts in a forward direction, as will be shown later. However, there are times when the bit-counter is required to effectively count in a reverse direction. When counting in a reverse direction, the read-write word cycle begins by reading out the high-order bit of the high-order digit; i.e., bit "*d*" of digit #9. The signal which provides for this reverse operation of the bit-counter comes from line DBD. When line DBD is TRUE (line (DBD)' thus being FALSE), the bit-counter counts in a reverse direction. Conversely, when line DBD is FALSE (line (DBD)' thus being TRUE), the bit-counter counts in a forward direction.

If it be desired that the bit-counter is to count in a forward direction, lines (DBD)', BC*a*, and BC*b* must first be TRUE. Therefore, as all three inputs to logical AND **1158** are simultaneously TRUE, lines B*d* and B*d*M are also TRUE, the remainder of output lines B*a* through B*c* and B*a*M through B*d*M being FALSE. When input line CYC goes from TRUE to FALSE, the prime inputs to both of flipflops **6041** and **6042** go from TRUE to FALSE. Consequently, both flipflops **6041** and **6042** change state, so that lines BC*a* and BC*b* go FALSE and lines (BC*a*)′ and (BC*b*)′ go TRUE. As lines (DBD)′, (BC*a*)′, and (BC*b*)′ are now TRUE, all of the inputs to logical AND **1152** are simultaneously TRUE, and thus output line B*a* is TRUE, with the remaining outputs B*b* through B*d* being FALSE. When line CYYC goes from FALSE back to TRUE, the state of both flipflops remains unchanged. However, when line CYC again goes from TRUE to FALSE, the reference input to flipflop **6041** goes from TRUE to FALSE. At this time, lines BC*a* and (BC*b*)′ are TRUE, and lines (BC*a*)′ and BC*b* are both FALSE. Thus, as inputs (DBD)′, BC*a*, and (BC*b*)′ of logical AND **1154** are simultaneously TRUE, output line B*b* is TRUE, and remaining outputs B*a*, B*c*, and B*d* are FALSE. When line CYC goes from TRUE to FALSE a third time, all input lines to logical AND **1156** are simultaneously TRUE, and output line B*c* is likewise TRUE, with lines B*a*, B*b*, and B*d* being FALSE. Finally, when CYC goes from TRUE to FALSE a fourth time, line B*d* is TRUE, with lines B*a* through B*c* being false.

To summarize, only readout line B*d* is initially set TRUE when the computer is first turned "ON" or when pushbutton RS1 (FIG. 76) is actuated; the remaining readouts, B*a* through B*c*, are FALSE. At this time, for each successive occurrence of a TRUE-to-FALSE reversal of line CYC, the bit-counter is incremented by a count of one binary bit. Thus, with four such reversals of line CYC, the bit-counter counts B*a*, B*b*, B*c*, and back to B*d*. When the bit-counter is to count backwards, line DBD is rendered TRUE instead of line (DBD)′. Consequently, at this time, line B*a* instead of line B*d* is TRUE; remaining lines B*b* through B*d* are FALSE. Thereafter, for each successive occurrence of a TRUE-to-FALSE reversal of line CYC, the counter is decremented by a count of one bit. Thus, with four such reversals of line CYC, the bit-counter counts B*d*, B*c*, B*b*, and back to B*a*. It is to be appreciated, of course, that lines B*a*M through B*d*M are at all times of the same state as lines B*a* through B*d*, respectively. It is also to be appreciated, as before stated, that lines B*a*M through B*d*M of FIG. 62 are respectively identical to lines B*a*M through B*d*M of FIG. 60 and anywhere else they may appear in the computer circuitry.

19. *X and Y Grounders*

In the left portion of FIG. 66, there are diagrammatically shown two sets of ten X-grounders, one set of ten represented by output lines XGϕ through XG9, and the other represented by output lines (XGϕ)′ through (XG9)′. Each of the twenty X-grounders comprises a serially-connected network of a two-input logical AND of type "R1," an inverter amplifier of type "I12," and an inverter amplifier of type "I2." The emitter electrodes of inverters **3268** through **3287** are connected together and returned via line CRX to output terminal **843** of temperature-compensated current regulator **7001**, which has previously been shown and described in detail in connection with section (*g*) of FIG. 51.

The mode of operation of each of the X-grounders is somewhat straightforward. When both inputs to any one of logical AND's **1281** through **1300** are simultaneously TRUE, that particular AND circuit effectively causes a constant-current operating potential of approximately −10 volts to be applied to its respective output lead; i.e., one of leads XGϕ through XG9 or (XGϕ)′ through (XG9)′.

The Y-grounders shown in the right section of FIG. 66 are of essentially identical circuit configuration to the just-described X-grounders, and, as their mode of operation is also essentially the same as that of the X-grounders, a detailed description thereof would result in unnecessary repetition. The prime difference between the X and Y-grounders is that lines WR and (WR)′ are individually ANDED with lines DϕM, D1 through D8, and D9M for the X-grounders and individually ANDED with lines W1ϕ through W19 for the Y-grounders, lines DϕM, D1 through D8, and D9M being the output leads of a digit counter to be described next.

20. *Digit Counter*

With reference to FIG. 63, there is shown a scale-of-ten binary counter, herein called a "digit counter," which is utilized, among other purposes, to select a particular order digit of the word stored in an address in memory. The digit counter includes four type "F2" flipflops **6043** through **6046** effectively connected in cascade in a manner such that, each time line AD goes from TRUE to FALSE, the following takes place: flipflop **6043** changes state; except during a so-called "recycling" mode of operation, which will be described hereinafter, flipflop **6044** changes state if the state of line D*a* goes from TRUE to FALSE; flipflop **6045** changes state if the state of line D*b* goes from TRUE to False; and flipflop **6046** changes state if the state of line *Dc* goes from TRUE to FALSE.

For example, for a count of "0," flipflops **6043** through **6046** are in a FALSE state; that is, their states are such that lines D*a* through D*d* are FALSE and lines (D*a*)′ through (D*d*)′ are all TRUE. If it is desired for the digit counter to count in a forward direction (i.e., 0, 1, 2, 3, . . . 7, 8, 9), line (DBD)′ is selectively set TRUE. After flipflops **6043** through **6046** are set FALSE, all of the inputs to logical AND **1178** are simultaneously TRUE. Thus, readout line Dϕ is also TRUE and thereby represents a count of "0." Thereafter, the first time line AD goes from TRUE to FALSE, flipflop **6043** changes state, and line D1, at that time, is the only readout line that is TRUE, thus indicative of a count of "1." The second time line AD goes from TRUE to FALSE, flipflops **6043** and **6044** both change states, so that line D2 is TRUE, with the remaining readout lines being FALSE. The counting continues until a count of "9" has been reached, as indicated by lines D*a* and D*d* being TRUE. Thereafter, the next time line AD goes from TRUE to FALSE, flipflops **6043** and **6046** both change state, so that lines D*a* through D*d* are thereafter FALSE and lines (D*a*)′ through (D*d*)′ are thereafter TRUE, thus representing the original count of "0."

When the computer is first turned "ON," or when pushbutton RS1 (FIG. 76) is actuated, the digit counter is preset to a count of "9" by reset line (RS)′ going from TRUE to FALSE. There is no need to preset the digit counter to "9" at any other time, as it is always returned to a count of "9" at the conclusion of every "word cycle" or "sub-command," which are to be described hereinafter.

To summarize: the ten readout lines Dϕ through D9 individually and sequentially go from FALSE to TRUE, representing counts from "0" through "9" respectively. That is, line Dϕ is TRUE when the digit counter is at a count of "0," line D1 is TRUE when the digit counter is at a count of "1," etc. When the digit counter counts in a forward direction, it starts at a count of "9," and for each successive TRUE-to-FALSE change of state of line AD, output line Dϕ goes TRUE, and, thereafter, a successively higher-order readout line goes TRUE, the remaining readout lines, of course, being FALSE in each instance.

At certain times, it is desirable to read a word out of memory in reverse order by starting with the tenth-order digit thereof. In this instance, flipflops **6043** through **6046** still essentially count in a forward direction, however; only the manner of decoding the outputs thereof is changed. Such a change is made simply by causing line DBD to be TRUE instead of line (DBD)'. When line (DBD)' is TRUE, the digit counter counts in a forward direction; however, when line DBD is TRUE, the digit counter starts at a count of "0." Thereafter a succession of TRUE-to-FALSE transitions of line AD produces a retrogressive count of 9, 8, 7, 6, . . . 1 and back to 0.

As illustrated in the block diagram of FIG. 45, readout lines Dϕ through D9 are also utilized by the computer for various control purposes, in addition to memory digit selection. In addition to the read-out lines following the forward-reverse logic, there is an additional readout line D9L, which is TRUE each time lines D*a* and D*d* are simultaneously TRUE, regardless of whether the digit counter is operating in a forward or a reverse direction.

21. *Y-Drivers*

In the rightmost section of FIG. 65 there are shown ten Y-drivers, whose output lines are respectively labeled YDϕ through YD9. Each driver includes a serially-connected network comprising a two-input type "R1" logical AND, a type "I12" inverter, and a type "E3" emitter follower amplifier. As previously described, by selective energization of line YDϕ, all addresses whose low-order digit identification is a "zero" are effectively "selected"; i.e., addresses ϕϕ, 1ϕ, 2ϕ, . . . 8ϕ, and 9ϕ, by selective energization of line YD1 all addresses ending in a "one" are effectively selected; i.e., 11, 21, 31, . . . 81, and 91; by selective energization of line YD2, all addresses ending in a "two" are effectively selected; and so on, with line YD9 effectively selecting all addresses ending in a "nine"; i.e., addresses ϕ9, 19, 29, . . . 89, and 99. Selective energization of lines YDϕ through YD9 is effected by logically ANDING line YDW with each of lines Wϕϕ through Wϕ9 which originate at the decoder portion of the low-order digit section of a "word-selecting" register next to be described.

22. *Word-Selecting Register*

The word-selecting register is a two-digit register which is utilized to temporarily store a two-decimal-digit number representing the address location of the word in memory from which reading is to take place. That is, the word located in memory at the address corresponding to the number stored in the word-selecting register will be the next word read out upon occurrence of a read-write work cycle. The word-selecting register includes eight type "F2" flipflops, four of which effectively store, in binary-coded form, the low-order digit, and the remaining four effectively store, in binary-coded decimal form, the high-order digit of the address-representing number stored therein.

With reference to FIG. 65, the four flipflops which are utilized to store the low-order decimal digit of the number in the word-selecting register are **6051** through **6054**. Flipflop **6051** effectively stores low-order bit "*a*," flipflop **6052** effectively stores second-order bit "*b*," flipflop **6053** effectively stores the third-order bit "*c*," and flipflop **6054** effectively stores the fourth or high-order bit "*d*" of the low-order decimal digit of the number.

Output lines Wϕ*a* through Wϕ*d* and (Wϕ*a*)' through (Wϕ*d*)' from flipflops **6051** and **6054** are connected as inputs to a decoder network comprising ten type "R1" logical ANDS **1261** through **1270**, which are individually cascaded with a type "D1" logical OR, a type "I5" inverter, and a type "I15" inverter. Thus, when the low-order section of the word-selecting register is storing a count of "zero," output lines Wϕ*a* through Wϕ*d* are FALSE, and lines (Wϕ*a*)' through (Wϕ*d*)' are TRUE. As all of the input lines to logical AND **1261** are simultaneously TRUE, output line Wϕϕ is TRUE, indicative of the "zero" storage in flipflops **6051** through **6054**. In a like manner, it is seen that only one of output lines Wϕϕ through Wϕ9 of the decoder is TRUE at any given instant, and a TRUE state of each is indicative of a corresponding one of decimal digits "0" through "9" being stored in the low-order section of the word-selecting register.

There are two "presets" for the word-selecting register. The first preset is by means of selective energization of line (PWϕ)', which unconditionally presets the word-selecting register to "00." The second preset is by means of selective energization of line ST1, which presets the word-selecting register to "01" representation as indicated by a TRUE state of line Wϕ1, lines Wϕϕ and Wϕ2 through Wϕ9, of course, being FALSE. However, both the low-order section and the high-order section of the word-selecting register must first be set to zero via line (PWϕ)' in order for line ST1 to be effective to preset the word-selecting register to "01." The reason for this is that, as shown, line ST1 is effective only in presetting the state of flipflop **6051** and does not affect the states of the remaining flipflops of both sections of the word-selecting register.

With reference to FIG. 64, the four flipflops which are utilized to store the high-order decimal digit of the number stored in the word-selecting register are **6047** through **6050**. The state of flipflop **6047** represents the low-order bit "*a*," the state of flipflop **6048** represents the second-order bit "*b*," the state of flipflop **6049** represents the third-order bit "*c*," and the state of flipflop **6050** represents the fourth or high-order bit "*d*" of the decimal digit. Output lines W1*a* through W1*d* and (W1*a*)' through (W1*d*)' from flipflops **6047** through **6050** are connected as inputs to a decoder network comprising ten type "R1" logical ANDS **1226** through **1235**, which are individually cascaded with a type "D1" logical OR, a type "I5" inverter, and a type "I15" inverter. As in the low-order section of the word-selecting register, as previously mentioned, its high-order section is also unconditionally preset to "zero" by line (PWϕ)', which presets the reference outputs of flipflops **6047** through **6050** FALSE.

Included in the high-order section of the word-selecting register are ten readouts which decode the decimal digit representation as represented by the states of flipflops **6047** through **6050**, the readouts being by way of output lines W1ϕ through W19. For example, when the high-order digit in the word-selecting register is a "0," only line W1ϕ is TRUE. Line W11 is TRUE only when the high-order digit is a "1," and so on, so that line W19 is TRUE only when the high-order digit is a "9." Then ten output lines W1ϕ through W19 are each logically ANDED with each of lines WR and (WR)', as shown in FIG. 66, to selectively energize the Y-grounders which effectively select the high-order digit of the address in memory of the word to be read out upon occurrence of the next read-write cycle.

23. *Summary of Word-Selection*

There has now been shown and described the mode of selection of a particular string or addresses of cores and how a particular binary bit of a particular digit of the word is selected during a read-write cycle of operation. However, in summary, the word-selecting register effectively stores a two-digit decimal number which corresponds to the address in memory that is to be effected during the next read-write cycle. Even though there are, in fact, only twenty readout lines Wϕϕ through W19 for the word-selecting register, in combination they effectively function as one hundred readouts. In other words, one of lines W1ϕ through W19 is TRUE, indicative of the high-order digit of addresses ϕϕ through 99. At the same time, one of lines Wϕϕ through Wϕ9 is TRUE, indicative of the low-order digit of the address. As an illustrative example, address 96 is represented by lines W19 and Wϕ6 simultaneously being TRUE; address 48 is represented by lines W14 and Wϕ8 simultaneously being TRUE, and so on. Thus, a half-select current is allowed to flow in the "Y" direction only in the vertical string of cores located in the address corresponding to the number stored in the word-selecting register. The number in the word-selecting register remains therein during the entire time the ten-digit word is being read out of memory. Other suitable means have been provided for selecting addresses "A" and "B," as will later be seen.

The bit and digit counter combination determines which row of cores is to receive a half-select current impulse during a read-write cycle. That is, the output of the bit counter determines which bit is to be read out—i.e., bit "*a*," "*b*," "*c*," or "*d*," and the digit counter determines which order digit is to be read out. For example, if the bit counter is at "*a*" and the digit counter at "1," a half-select current impulse is permitted to flow only through the row of cores corresponding to the low-order bit of the second-order digit. If the bit counter is at "*d*" and the digit counter at "9," a half-select current impulse is permitted to flow only through the row of cores corresponding to the high-order bit of the tenth-order digit, and so on.

24. *Synchronizing Clocks*

Before going into a detailed description of a complete read-write cycle of a word stored in a particular address in memory, it is deemed desirable, at this point, to briefly describe the various "clocks" which are utilized by the computer for synchronization purposes, to insure the proper sequence of data handling and transfer.

With reference to FIG. 84, a 50-kilocycle multi-vibrator **6125**, substantially identical to the one heretofore shown and described in connection with section (*a*) of FIG. 50, has the reference output thereof connected through AND gates **1778** and **1779** to both the reference and the prime inputs of flip-flop **6127**. As multivibrator **6125** is essentially a 50 kc. square-wave generator, the signal voltage appearing on the reference output lead MV thereof is as shown by the topmost waveform in FIG. 87B. If at "TIME–1" the state of line MV is reversed from TRUE to FALSE, the state of line MV reverses from FALSE back to TRUE ten microseconds later at "TIME–2." Ten microseconds later, at "TIME–3," the state of line MV again reverses from TRUE to FALSE, and so on. As heretofore mentioned, the state of line (MV)′ is effectively 180 degrees out of phase with the state of line MV.

If it is assumed that directly preceding TIME–1 the states of lines MV and C31 are both TRUE, as shown, at TIME–1 the reference input to flip flop **6127** is reversed from TRUE to FALSE. Consequently, flip flop **6127** is triggered so that the state of line C31 is rendered FALSE and the state of line C13 is rendered TRUE. At TIME–3, when the state of line MV again is reversed from TRUE to FALSE, the state of line C31 is rendered TRUE and the state of line C13 is rendered FALSE. This sequence of events is continually repeated during operation of the computer.

At TIME–2, the state of line (MV)′ is reversed from TRUE to FALSE, and, consequently, flip flop **6128** is triggered so that the state of line C42 is thereby rendered FALSE and the state of line C24 is thereby rendered TRUE. At TIME–4, when the state of line (MV)′ again reverses from TRUE to FALSE, the states of lines C42 and C24 are reversed. At TIME–1, when the state of line C31 reverses from TRUE to FALSE, flip flop **6129** is triggered so that the state of line C41 is thereby rendered FALSE, and, simultaneously therewith, the state of line C41 is rendered TRUE. Thirty microseconds later, the state of line C24 reverses from TRUE to FALSE. Therefore, at TIME–4, flip flop **6129** is reset to its initial state, so that immediately thereafter the state of line C41 is rendered TRUE and the state of line C14 is simultaneously rendered FALSE.

As shown, output lines C31, C13, C42, and C24 are variously connected as inputs to flip flops **6129** through **6132** to provide suitably phased clock pulses on output lines C41, C14, C43, C34, C23, and C12. It is to be noted that the highest-order numeral designation thereof denotes the time at which the state of that particular line is rendered TRUE and the lowest-order numeral designation denotes the time at which the state of that particular line is rendered FALSE. For example, the state of line C41 is rendered TRUE at each TIME–4 and FALSE at each TIME–1, the state of line C23 is rendered TRUE at each TIME–2 and FALSE at each TIME–3, and so on.

The 400-cycle multivibrator **6126** has a period of approximately 1.24 milliseconds. Consequently, due to the synchronizing action with respect to the state of line C41, as heretofore described in connection with section (*b*) of FIG. 50, the state of line ICC is rendered TRUE and the state of line (ICC)′ is rendered FALSE at TIME–1; approximately 1.24 milliseconds thereafter, at TIME–1, the state of line ICC is rendered FALSE and the state of line (ICC)′ is rendered TRUE. Again, approximately 1.24 milliseconds thereafter, at TIME–1, the state of line ICC again is rendered TRUE and the state of line (ICC)′ is again rendered FALSE. As this sequence of events is continually repeated during operation of the computer, it is evident that the output from the 400-cycle multivibrator **6126** is effectively synchronized with the output from the 50 kc. multivibrator **6125**.

25. *Detailed Read-Write Cycle*

With reference to FIG. 60, during a complete read-write cycle of a word stored in a particular address in memory, line SMC is TRUE for a period of 1600 microseconds, and line MYW is TRUE for a period of 1650 microseconds. Therefore, as all the inputs to logical AND **1096** are simultaneously TRUE at TIME–3, line YDW also goes TRUE at TIME–3 and stays TRUE for a period of 20 microseconds until TIME–1; at TIME–1, line YDW goes FALSE and stays FALSE for 20 microseconds until TIME–3; at TIME–3 line YDW again goes TRUE, and so on. Thus, it is seen, at TIMES **1** and **3**, line YDW goes FALSE and TRUE, respectively, for periods of 20 microseconds each. This is done to permit one of Y-driver output lines YDϕ through YD9 (FIG. 65) to be selectively energized for a period of 20 microseconds by selected ones of output lines Wϕϕ through Wϕ9 of the decoding portion of the low-order section of the word-selecting register, as will more fully be described hereinafter.

As previously described in connection with FIG. 66, selected pairs of Y-grounder output lines YGϕ—(YGϕ)′, YG1—(YG1)′ . . . YG9—(YG9)′ are conditioned at predetermined times by a corresponding one of output lines W1ϕ through W19 of the decoding portion of the high-order section of the word-selecting register (FIG. 64), a selected one of each pair being selectively energized by one of lines WR and (WR)′. As shown in FIG. 66, lines WR and (WR)′ alternately effect energization of only one of the Y-grounder output lines. That is, only one of the Y-grounder output lines is energized at any given time; when that particular Y-grounder output line is de-energized, another Y-grounder output line is energized, and so on. This is clearly illustrated in the timing chart of FIG. 87A, which is to be referred to, from time to time, during the following detailed description of a complete read-write cycle.

It will be assumed that it is desirous to read a word out of memory address ϕϕ, that the reading operation is to begin with the lowermost core in address ϕϕ, and that all data read out is to be re-stored in address ϕϕ. Thus, to begin the reading operation, lines (WR)′ and W1ϕ (FIG. 66) simultaneously go TRUE, thereby energizing Y-grounder output line (YGϕ)′ at TIME–1, as shown in the just-mentioned timing chart of FIG. 87B. Simultaneously therewith, lines XDW and B*a*M (FIG. 60) go TRUE, thereby energizing X-driver output line XD*a* at TIME–1. Also simultaneously therewith, lines DϕM and (WR)′ (FIG. 66) are TRUE to energize X-grounder output line (XGϕ)′ also at TIME–1. Consequently, as both the X-driver, XDa, and the X-grounder (XGϕ)′ (FIG. 52A) are simultaneously energized at TIME–1, a current impulse of half-select magnitude (approximately 180 ma.) flows from left to right through the bottom row of cores to "half-select" each of the cores in the row toward a binary "zero" representation, as previously described, the binary value of the low-order bit of the first-order digit of the word stored in memory address $\phi\phi$ being represented by the initial state or direction of magnetization of core **885**, as before stated.

Twenty microseconds later, at TIME–3, line YDW (FIG. 60) goes TRUE to effect energization of Y-driver output line YDϕ (FIG. 65). Thus, as the Y-driver output line YDϕ and Y-grounder output line (YGϕ)′, shown in FIG. 52A, are both, for the first time, simultaneously energized at TIME–3, a current impulse of half-select magnitude (approximately 180 ma.) flows downwardly through the leftmost column of cores to "half-select" each of the cores in address $\phi\phi$ toward a binary "zero" representation. As the magnetommotive forces produced by the two half-select currents are additive in the region of core **885** only, immediately following TIME–3 core **885** is the only core in the entire memory that is magnetically set to a binary "zero" representation at this time. If it is assumed that core **885** previously was in a magnetic remanent state representative of a binary "one," the state of magnetization of the core is reversed immediately following TIME–3. Consequently, with respect to terminal **872**, a negative-going output pulse is induced in memory sense winding **871** (FIG. 52) having a peak amplitude occurring approximately 3 microseconds after TIME–3.

The negative-going output pulse induced in memory sense winding **871** renders memory sense amplifier **4399** (FIG. 67A) conductive during presence thereof, so that the uppermost input to logical AND **1163** goes TRUE during presence of the pulse from memory, as shown in FIG. 87. Assuming that the signal on the lowermost input to AND **1163** from flipflop **6149** is also TRUE, both of the common-load transistors making up AND **1163** (FIG. 48j) are rendered non-conductive, so that the signal applied to the reference input of flipflop **6055** goes FALSE approximately 3 microseconds after TIME–3 and flipflop **6055** is triggered thereby, so that output line MSA goes TRUE, indicating that a binary "one" had previously been stored in core **885** (FIG. 52); output line (MSA)′, when TRUE, indicates that the core was previously storing a binary "zero."

At TIME–4, line (WR)′ goes FALSE (FIG. 66) and line WR goes TRUE. Therefore, X and Y-grounder lines (XGϕ)′ and (YSϕ)′ are both de-energized, and simultaneously therewith, X and Y-grounder lines XGϕ and YGϕ are energized. As core **885** (FIG. 52A) was previously storing a binary "one," X and Y-driver output lines XDa and YDϕ remain energized, and, consequently, a full-select current impulse flows through the core in the opposite direction to "re-set" the core back to its initial binary "one" state immediately following TIME–4.

However, if the core had been initially storing a binary "zero," instead of a binary "one," a voltage impulse would not have been induced in the memory sense winding, and, consequently, output line MSA (FIG. 67A) remains FALSE instead of going TRUE, thus indicative of the binary "zero" storage in core **885**. In that instance when line MSA remains FALSE, line MXW (FIG. 60) likewise remains FALSE, and, consequently, not all of the inputs to logical AND **1098** are TRUE, as before. As a result, line XDW goes FALSE and causes X-driver output line XDa to be de-energized. This is illustrated in FIG. 87 by the dotted portion of the output signal representation of X-driver line XDa between TIME–4 and TIME–1 for bit "a." X-driver line XDa now being de-energized at TIME–4, a binary "one" representing half-select current flows through core **885** in the "Y" direction only, and, consequently, the core is not "reset" to a binary "one" state, but, instead, remains in its initial binary "zero" state.

It is to be noted that, during a read-write cycle, half-select current is first applied in the "X" direction for a period of 20 microseconds—i.e., from TIME–1 to TIME–3—before a coincidental half-select current is applied in the "Y" direction through the core being "read." One reason for operating the computer memory in this manner is due to the physical placement of memory sense winding **871** (FIG. 52). Due to the fact that the sense winding is oriented parallel with respect to the conductors threaded through the core in the "X" direction, when a half-select current is first applied in the "X" direction, a voltage impulse is induced in the sense winding even though the magnetic state of each core remains unchanged. In fact, the amplitude of this voltage may be from ten to fifteen times greater than the amplitude of the voltage induced in the sense winding due to a "change-of-state" of one of the cores. Consequently, it is desirable to allow sufficient time to lapse after application of a half-select current in the "X" direction so that this unwanted voltage impulse is dissipated and thus alleviate any undesirable effects therefrom.

Another reason for this 20-microsecond delay is that a half-select current through a core causes a noise impulse to be induced in the sense winding especially if the previous half-select current through the core was in the opposite direction. The amplitude of this noise impulse is, in most instances, approximately 7 millivolts per core and requires from ten to fifteen microseconds to be dissipated. As there are 100 cores in the "X" direction per row, and, as the noise from all of the cores in each row is additive, the total maximum amplitude of the noise impulse is approximately 700 millivolts.

In an attempt to simplify the description and to insure a complete understanding of the inherent complexity of the mode of operation of the computer memory, it has been assumed to this point that all of the cores making up memory addresses $\phi\phi$ through 99 and addresses "A" and "B" are magnetized in the same direction to represent either a binary "one" or a binary "zero." However, in the actual operation of the computer memory, this is not the case, the reasons for which are as follows:

As previously described with respect to FIG. 52, memory sense winding **871** is sequentially threaded in the same direction through every other one of the forty rows of cores making up the entire memory. Thus, if every core in the memory were magnetically set in the same direction to represent, say, a binary "one," opposite polarity output pulses are induced in the memory sense winding during a read-write cycle, all of which pulses individually represent a binary "one" storage in the respective core. For example, the cores making up the odd-numbered rows are responsible for output pulses of one polarity indicative of a binary "one" storage, whereas the cores making up the even-numbered rows are responsible for output pulses of an opposite polarity also indicative of a binary "one" storage.

Also, as previously mentioned, it has been found that a subsequently-applied half-select current impulse through a core causes a noise impulse to be induced in the sense winding if the previously-applied half-select current impulse through the core was in the opposite direction to the subsequently-applied half-select current impulse; however, the subsequently-applied half-select current impulse through the core causes a much smaller amplitude noise impulse to be induced in the sense winding if the subsequently-applied half-select current impulse is in the same direction as the previously-applied half-select current impulse.

Therefore, to alleviate the need for sensing amplifiers which must be responsive to input impulses of opposite polarity, and also, during reading of a particular core, in order to subsequently apply a half-select current impulse therethrough in the "Y" direction, which is in the same direction as the half-select current impulse previously applied in the "Y" direction therethrough during writing of a previously-read core, it has been found desirable during the operation of the memory of the present computer for all of the cores making up the odd-numbered rows to be magnetically set in one direction to represent storage of a binary "one" and for all of the cores making up the even-numbered rows to be magnetically set in the opposite direction to represent storage of a binary "one." Consequently, during a read-write cycle of operation, binary "one" signifying impulses of the same polarity are induced in the memory sense winding, and thus the sense amplifiers need only be responsive to single polarity impulses; and, also, much smaller amplitude noise impulses are induced in the sense winding during the reading operation.

With the foregoing in mind, the next portion of the read-write cycle will be described and directed to bit "*b*" of the low-order digit of the word stored in address $\phi\phi$.

Now that core **885** (FIG. 52A) has been read and then "reset" to represent its initial binary "one" storage, line CYC (FIG. 61) goes from TRUE to FALSE at the following TIME–**1**, measured with respect to bit "*b*." As previously described in detail with respect to the bit counter shown in FIG. 62, when line CYC goes FALSE for the second time, line B*a*M goes FALSE, line B*b*M goes TRUE, and lines B*c*M and B*d*M remain FALSE. As line XDW (FIG. 60) also goes TRUE at TIME–**1**, X-driver output line XD*a* is de-energized and output line XD*b* is energized at TIME–**1**, as shown in FIG. 86. As line YDW (FIG. 60) goes FALSE at each TIME–**1**, Y-driver output line YDϕ (FIGS. 65 and 52A) is de-energized, and consequently, the half-select current in the "Y" direction through the cores of address $\phi\phi$ is thereby not permitted to flow. However, as X-grounder line XGϕ (FIG. 52B) remains energized during TIME–**1**, and as X-driver line XD*b* is energized for the first time at TIME–**1**, a half-select current flows from right to left through core **886** to "half-select" the magnetic state of core **886** indicative of a binary "zero." It is to be noted that the half-select current in the "X" direction through core **885** flowed from left to right, instead of right to left, to magnetically "half-set" core **885** toward a binary "zero" representation. Thus, it is seen, the half-select read current flows in one direction through the cores in the even-numbered rows and in the opposite direction through the cores in the odd-numbered rows.

At TIME–**2**, flipflop **6055** (FIG. 67A) is reset by line **C42** so that output line MSA is thereafter FALSE. At TIME–**3**, line YDW (FIG. 60) goes TRUE to again effect energization of line YDϕ (FIG. 65). Thus, as Y-driver output line YDϕ and Y-grounder line YGϕ (FIG. 52) are both simultaneously energized at TIME–**3**, a current impulse of half-select magnitude flows upwardly through core **886** to cause "full-selection" of the magnetic state of core **886** to a binary "zero" representation. If it is assumed that the magnetic state of core **886** is indicative of a binary "one," the state of magnetization of the core is reversed immediately following TIME–**3**, and, consequently, a negative-going impulse is again induced in memory sense winding **871** (FIG. 52) with respect to terminal **873**, approximately three microseconds after TIME–**3**, as before.

As before, the negative-going impulse induced in memory sense winding **871** causes the reference input to flipflop **6055** to go FALSE shortly after TIME–**3**, and, consequently, flipflop **6055** is triggered so that line MSA is TRUE, indicating that a binary "one" had previously been stored in core **886**. At TIME–**4**, line (WR)′ goes FALSE (FIG. 66) and line WR goes TRUE. Consequently, X and Y-grounder lines XGϕ and TGϕ are both de-energized, and, simultaneously therewith, lines (XGϕ)′ and (YGϕ)′ are energized. As core **886** was previously storing a binary "one," X and Y-driver output lines XD*b* and YDϕ remain energized for ten microseconds longer to effect "resetting" of core **886** back to its initial binary "one" state immediately following TIME–**4**.

The previously-described sequence of events is now repeated with respect to cores **887** and **888** and also with respect to the remaining ones of the forty cores making up address $\phi\phi$. In the illustrative example given, as evidenced by the presence and absence of memory sense amplifier pulses (FIG. 87A), the binary number stored in cores **885** through **888**, corresponding to low-order bit "*a*" through the high-order bit "*d*," is "0111," thus indicating that the low-order digit of the word stored in address $\phi\phi$ is a "seven."

To summarize: Alternate cores of each memory address are magnetically set in one direction to represent a binary "one," whereas the remaining cores in the address are magnetically set in the opposite direction to represent a binary "one." When the core representing bit *a* of the low-order digit is read (i.e., core **885**), a half-select "read" current in the "X" direction flows to the right through core **885** at the first TIME–**1**. At TIME–**3**, a half-select read current in the "Y" direction flows downwardly through core **885** at the first TIME–**3** to magnetically set core **885** toward binary "zero." If the core was originally storing a binary "one," its magnetic state is therefore reversed, and a negative-going voltage pulse is induced in memory sense winding **871**, which causes the half-select currents in the "X" and "Y" directions through core **885** to be reversed to magnetically reset core **885** to binary "one" at TIME–**4**. At the second TIME–**1**, a half-select current flows to the left in the "X" direction through core **886**. At the second TIME–**3**, a half-select current flows upwardly through core **886** to magnetically set core **886** to binary "zero." If core **886** was originally storing a binary "one," a second negative-going potential impulse is induced in the memory sense winding to cause the half-select currents in the "X" and "Y" directions through core **886** to be reversed to reset core **886** to binary "one" at the second TIME–**4**. As before stated, this sequence of events is sequentially repeated from the lowermost to the uppermost of the cores making up address $\phi\phi$.

26. *Sense-Amplifier Strobe*

With reference to FIG. 67A, inverter **4393** is effectively connected in series with memory sense winding **871**, and inverter **4395** is effectively connected in parallel with respect to the memory sense winding. As shown, inverter **4395** is rendered non-conductive from TIME–**3** to TIME–**4** by line **C34**, and inverter **4393** is rendered conductive from TIME–**3** to TIME–**4** by line **C43**. Thus, as inverter **4395** is non-conductive during the time each of the memory cores is being read, the inverter exerts essentially no effect on sense winding **871**; however, as inverter **4393** is conductive during the time each core is being read, the lower end of sense winding **871**—i.e., terminal **872**—is effectively grounded thereby during that time to complete the circuit therethrough. From TIME–**4** to TIME–**3**, sense winding **871** is effectively open-circuited by inverter **4393** and effectively short-circuited by inverter **4395**, thus rendering sense winding **871** non-effective at all times except for the time interval during which reading of a memory core is to take place.

As previously described with respect to FIG. 87A, if a binary "one" was previously stored in the core being read, the output of a memory sense amplifier is TRUE after a time lapse of approximately three microseconds after TIME–**3**. Thus, to prevent undesirable noise impulses from entering the system, the output of sense amplifier **4399** is sampled, or "strobed," for a period of approximately three microseconds, starting approximately three microseconds after each TIME–**3**. Strobing is effected by a three-microsecond one-shot flipflop **6147**, which is triggered at TIME–**3** by line **C31**, so that the prime output thereof goes FALSE at TIME–**3** and remains FALSE for a period of three microseconds, after which time the prime output thereof returns to a TRUE condition. The prime output of flipflop **6147** is connected to the reference input of a three-microsecond one-shot flipflop **6149**, so that, after a time lapse of three microseconds after TIME–**3**—i.e., when the prime output of flipflop **6147** goes TRUE—the reference output of flipflop **6149** goes TRUE and stays TRUE for a period of three microseconds, after which time the reference output thereof goes FALSE. Therefore, the lowermost input line to logical AND **1163** goes TRUE only after a time lapse of three microseconds after TIME–**3**, and stays TRUE for a period of only three microseconds.

As previously described, memory addresses "A" and "B" (FIG. 52B) have a common sense winding **891** separate from the main memory sense winding **871**. Memory addresses "A" and "B" have a common pair of Y-grounder lines (ABG)′ and ABG and separate Y-driver lines AAD and BBD (FIG. 60), which function in the same manner as the just-described Y-grounder and Y-driver lines for memory addresses ϕϕ through 99. Thus, it is possible to read and write in only one of memory addresses "A" or "B" at any given time, the same as for memory addresses ϕϕ through 99. However, as addresses "A" and "B" have a separate sense winding from addresses ϕϕ through 99, a word from either of addresses ϕϕ through 99 may be read out simultaneously with the word being read out of address "A" or "B." Because of the common X-driver and X-grounder lines, it is not possible to write separate words in both a memory address and either of addresses "A" or "B." However, the same word may be written in both addresses. As shown in FIG. 67A, the output of "A" and "B" sense amplifier **4400** is strobed by single-shot flipflops **6148** and **6150** in exactly the same manner as memory sense amplifier **4399**, as was just described in detail. The strobed signals from sense amplifier **4400** operates a transistor logical AND **1164**, which triggers flipflop **6056** in response to a negative-going impulse from "A" and "B" sense winding **891**. As the complete read-write cycle of operation of addresses "A" and "B" is, for all practical purposes, exactly the same as the just-described read-write cycle for memory addresses ϕϕ through 99, a description thereof would only result in an undue repetition, which is not deemed necessary for a full understanding of the mode of operation thereof in view of the just-described read-write cycle for memory addresses ϕϕ through 99.

27. *Format of Instruction Words Generally*

In order to perform the functions of a "general purpose" type computer, the present computer is provided with eighteen different types of general purpose "instructions" which it, in a sense, "obeys" in order to allow the programmer to solve almost any type of mathematical problem. For reasons to become more apparent hereinafter, the just-described magnetic core memory stores both "instruction" words and data words, which appear as one and the same as far as the computer memory is concerned—i.e., a ten-digit binary-coded decimal number—and, to alleviate the necessity of providing "plus" or "minus" indications to identify positive and negative numbers, a negative number is stored in the computer memory as its complement. For example, the minimum negative number that is effectively stored is 0000000001, whose complement is 9999999999; the maximum negative number that is effectively store is 1000000000, whose complement is 9000000000. Thus, the maximum positive number stored is 8999999999. Both data and instruction words may essentially be stored in the computer memory either by indexing the words in the accounting machine keyboard portion of the computer, by totaling or subtotaling selected totalizers of the accounting machine, by reading the words from the magnetic strip portion of the ledger cards, or by reading the words from punched cards or paper tapes via conventional, commercially-available paper tape and card readers.

Programming the computer consists essentially of writing two related programs, the first program of which is written for the accounting machine portion of the computer, which programming consists of designing mechanical stops which are placed on the form-bar portion of the accounting machine, in a manner fully described in the before-referred-to Patent No. 2,626,749. In programming the accounting machine, the programmer decides at what carriage columnar position the program is to start and the columnar position to which the carriage is to be moved by subsequent machine cycles. The stops on the form bar may be thought of as "instructions" to the accounting machine which are physically located in unique places on the form bar, a group of stops (i.e., instructions) being termed a "program."

The program is effectively carried out by the accounting machine, by "sampling" each stop, starting with the first stop, and performing the sequence of actions called for by each stop as the carriage sequentially moves from stop to stop. Finally, the carriage is returned to the first stop to permit the program to be repeated if desired. It is to be noted, however, that only one instruction is carried out at any given time. The mere placing of the form bar on the accounting machine, or the placing of particular stops on the form bar, does not, in and of itself, cause the accounting machine to carry out the operation specified by either of the stops.

The second program essentially to be written is in the form of a sequential group of instruction words, that are to be stored in the computer memory, comprising a group of words ten decimal digits in length. The computer "samples" only one of these words at any one time, and, after carrying out that particular instruction, the computer then samples the next instruction word. The mere placement of the instruction words in the computer memory does not cause any instruction to be carried out, in the same manner that the mere placement of a form bar on the accounting machine does not necessarily cause any of the stops to be sampled.

To start the accounting machine program, a machine cycle is first initiated, which allows the accounting machine to "sample" the first stop in the program and carry out that particular instruction. After the computer carries out its instruction, it effectively "moves on" to the next instruction in the program, just as the accounting machine "moves on" to the next stop on the form bar. If desired, the computer program may be so written that the computer returns to the first instruction after it has finished the last instruction, to permit the program to be repeated.

As previously mentioned, there are eighteen basic instructions which the computer executes. Five of these basic instructions concern the ledger card and the accounting machine; one concerns reading paper tape; and twelve are concerned with arithmetic and control functions within the computer. Of these eighteen different types of instruction words, there are three types of instruction word formats. The first type of instruction word format is that of a "modified three" or "four address" instruction word. Nine of the basic instruction words have a second type of format, which takes advantage of the ability of the computer to increment addresses in an instruction word, as will be more fully described hereinafter. The remainder of the basic instruction words have a third type of format, which is peculiar to the function effected by that particular instruction words.

The structure of every instruction word is effectively "divided" into five groups of two decimal digits each, with each group hereinafter being called a "section." The first section is the two highest-order decimal digits of the instruction word and represents the code number identifying that particular instruction, the code numbers ranging from 00 through 17; the second section normally identifies either the memory address location of the first data word or the address of an instruction modifier; the third section normally identifies the address location of either the first or the second data words; the fourth section generally identifies the address of the result; and the fifth section usually identifies the address of the next instruction word in the program.

As an example of a format of a particular instruction word, an "ADD" instruction is illustratively utilized. Suppose that two amounts are to be added, one amount being located in memory address 42, the other amount being located in memory address 5ϕ, the sum of these two amounts is to be stored in memory address 52, and the next instruction word is stored in memory address 32. As the code number for an ADD instruction is 08, the instruction word appears as illustrated below:

Instruction Word Format				
Sect. 1	Sect. 2	Sect. 3	Sect. 4	Sect. 5
08	42	50	52	32

After the addition operation is completed, the computer, in a sense, "looks" in memory address 32 for the next instruction. Also, following the addition, the words in memory addresses 42 and 5ϕ remain the same as before, even though each was involved in the addition operation. However, regardless of the word originally stored in memory address 52 before the addition, memory address 52 now contains the sum word derived from the addition. The reason for this is that a memory address is automatically cleared prior to storage of another word in that particular address. Consequently, only the sum word is stored in the sum address.

A characteristic of nine of the eighteen instructions is the ability to effectively increment section 3 or decrement section 4 of any instruction word. That is, if one operation is called for which is to be sequentially performed on several words that are stored in memory locations having sequential addresses, it is not necessary to utilize a separate instruction for each operation performed, or to build an address-incrementing loop.

An example of an instruction in which address incrementing is utilized is the "enter-keyboard-words" instruction, abbreviated as EKW, having a code number designation "00," and interpreted by the computer as: "The words subsequently indexed in the accounting machine keyboard are to be sequentially stored in memory address locations X through Y, where X and Y are specified in section 2 and section 4, respectively, of the instruction word." Thus, if several sequential entries are to be indexed in the accounting machine keyboard, it is, of course, obvious that these successive entries must be stored in adjacent memory address locations. If it were necessary to utilize a separate instruction word for storage of each of the keyboard entries in a designated memory address, ten instruction words would then be necessary to store ten entries in memory. However, with an address incrementing instruction, all ten entries are sequentially stored by the use of only one instruction word. This incrementing feature not only simplifies programming, but also reduces the required number of instruction words necessary for completion of a given program. When utilizing an instruction with an address-incrementing format, the programmer specifies, in section 3 of the instruction word, the memory address of the first word on which an operation is required, and specifies, in section 4 of the instruction word, the memory address of the last word on which an operation is required. The only restriction is that the address specified in section 4 of the instruction word must be equal to, or greater than, the address specified in section 3. Thereafter, the computer sequentially generates each of the address locations between those two specified, even if the original instruction word is no longer stored in memory. However, after the instruction has been executed, the original instruction word may remain in memory unchanged, if desired.

28. *Format of Enter-Keyboard-Words Instruction Word (EKW—00)*

The EKW instruction is utilized when amounts are to be entered, and thus stored, in the computer memory, either by the words being indexed in the keyboard or by a preselected totalizer within the accounting machine being totalized. With only a single EKW instruction, from one to one hundred different amounts may selectively be stored in sequentially-ascending memory addresses. However, in order to enter the amount indexed in the accounting machine keyboard, a machine cycle of operation must first be initiated. The particular motor bar of the machine to be depressed to initiate a cycle of operation thereof is specified in the instruction word by the programmer. In this way, the programmer is assured that the correct motor bar is automatically depressed for each amount entered. Hence, there is prevented the possibility of the accounting machine carriage becoming out of synchronism with the program in the computer due to the operator's depressing the wrong motor bar. When the computer is "ready" to carry out on EKW instruction, the "enter-keyboard-words" lamp, illustrated in FIG. 1 as EK, is illuminated. The operator thereafter indexes the amount into the accounting machine keyboard and then depresses the "resume-program-bar." Immediately thereafter, the computer automatically effects depression of the particular motor bar specified in the instruction word. Consequently, a cycle of operation of the accounting machine is thereby initiated, and the amount just indexed in the keyboard thereof is entered into the selected memory address.

In order for the programmer to prepare an instruction word which instructs the computer to enter words into memory, from either the accounting machine keyboard or the totalizers therein, the two-digit decimal number for each of the five sections of the instruction word must be specified, the composite ten-digit decimal number thereafter representing the desired instruction word, as before stated. As the code number designation for an EKW command is "00," the two decimal digits in section 1 of the instruction word are, likewise, "00." In section 2 of the instruction word, the high-order decimal digit therein designates the motor bar code, and the low-order decimal digit therein designates the decimal point lamp code; section 3 designates the memory address in which the first word entered from the accounting machine is to be stored; section 4 designates the memory address in which the last word entered is to be stored; and section 5 designates the memory address location of the next instruction word in the program. Listed below, in chart form, are the various codes specified by section 2 of the instruction word and to which the computer is responsive:

Section 2 of the Instruction Word	
High-Order Digit of Section 2 Motor-Bar Code	Low-Order Digit of Section 2 Decimal-Point Lamp Code
0*x* Touch Upper Motor Bar 1*x* Hold Upper Motor Bar 2*x* Touch Middle Motor Bar 3*x* Hold Middle Motor Bar 4*x* Touch Lower Motor Bar 5*x* Hold Lower Motor Bar 6*x* No Motor Bar (ECW only)	*x*0 Between Columns 2 and 3 *x*1 Between Columns 5 and 6 *x*2 Between columns 8 and 9 *x*3 Automatic-Resume-Program with decimal point between columns 2 and 3

If desired, the programmer may instruct the computer to illuminate a preselected one of the three decimal point lamps to give a visual indication on the accounting machine keyboard as to the decimal point location for the amount to be indexed, decimal point lamp Pϕ being located on the keyboard between amount rows **2** and **3**, decimal point lamp P1 being located between amount rows 5 and 6, and decimal point lamp P2 being located between amount rows 8 and 9.

When the decimal point is desired between rows 5 and 6, the computer illuminates lamp P1 between these rows even though the amount 0.465,85 is printed as 0,465.85, and, when a decimal point is desired between rows 8 and 9, the computer illuminates lamp P2 between these rows even though the amount 0.465,855,55 is printed as 0,465,855.55.

As an example, suppose that three amounts are to be stored in memory. The first amount is desired to be stored in memory address 45, the second amount is desired to be stored in memory address 46, and the third amount is desired to be stored in memory address 47. Consequently, the desired address of the first word to be entered is placed in section 3 of the instruction word, and the desired address of the last word to be entered is placed in section 4 thereof. It is also assumed that, when the resume-program bar is depressed, it is desired that the computer automatically initiate depression of the middle motor bar **27** for a "touch" operation (code 2x), thereby effecting a cycle of operation of the accounting machine and thus enter the first amount into memory. Further assumed is that the decimal point lamp located between rows **5** and **6** is to be illuminated (code x1) and the next instruction is located in memory address ϕ2. Therefore, the instruction word appears as shown below:

Sect. 1	Sect. 2	Sect. 3	Sect. 4	Sect. 5
00	21	45	47	02

If the first amount to be indexed is 0.025,00 and enter-keyboard-words lamp **P1** is illuminated, amount key #2 of row **4** and amount key #5 of row **3** are manually depressed first. Thereafter, the resume-program-bar is manually depressed, and, consequently, the computer automatically effects depression of the middle motor bar **27** for a "touch" operation. The amount thereafter printed appears as 0,025.00. If the printing of cents ".00" is not desired, a particularly designed stop is placed on the form bar to suppress printing thereof.

When a negative amount is to be entered into the computer memory, reverse key "REV" is initially depressed. This not only causes the amount to be printed with a "diamond" symbol printed to the right thereof, but, in addition, causes the complement of the number to be stored in memory as a "credit balance." Negative amounts previously stored in the totalizers of the accounting machine are automatically stored in the computer memory as complements thereof.

There are three ways to total or sub-total the amount in the computer memory by means of the EKW instruction. If the carriage of the machine is stationary and the non-select key "NON SELECT" is depressed, both the desired totalizer and the particular type of arithmetic operation are selected by the totalizer control keys, so that, when the resume-program-bar is depressed, the motor bar specified in the instruction word is automatically depressed by the computer, and the amount in the selected totalizer is transferred into the computer memory; if the carriage is stationary on a stop and the resume-program-bar is depressed, the computer automatically effects depression of the desired motor bar, and the amount in the totalizer is entered into the computer memory, the totalizer in which the amount is initially stored being determined by the stop on the form bar; and, if the carriage is moving in either direction and is stopped over a non-automatic stop, depression of the resume-program-bar effects depression of the desired motor bar to transfer the amount into the computer memory, the amount being transferred coming from the stop-selected totalizer. However, if an automatic-resume-program instruction (code-x3) is placed in the decimal point lamp code position in the instruction word, the preselected motor bar is automatically operated by the computer as soon as the carriage is stopped, even though the resume-program-bar was not depressed. The automatic-resume-program instruction, in effect, depresses the resume-program-bar automatically, and the word printed out has its decimal point in the normal position. For example, suppose that a totalizer is to be subtotaled and the contents thereof stored in memory address 62. Assuming that a middle rotor bar "touch" operation is desired, along with an automatic-resume-program operation, the instruction word appears as "00 23 62 62 48," the space between each pair of decimal digits indicating the five sections of the instruction word. As soon as the carriage comes to rest on a stop, the middle motor bar 27 is depressed, regardless of which direction the carriage was moving when the stop was engaged. Thus, the automatic-resume-program instruction eliminates the need for automatic stops for the carriage. The numeral "6" in the motor bar code specifies that a motor bar is not to be operated and is used only for an "enter-card-word" instruction (ECW), which is to be more fully described hereinafter. Since a machine cycle is required to enter the indexed amount from the keyboard into the computer memory during an EKW instruction, the numeral "6" placed in the motor bar code has no significance in this instance.

29. *Format of Print-Out-Words Instruction Word (POW—01)*

By utilizing but a single "print-out-words" instruction, it is possible to selectively initiate the printing out of memory from one to one hundred words. If the memory address locations of the words are in sequential order, the words are printed out sequentially, and, thus, only the addresses of the first and last words to be printed out need be specified. The first word printed is the word located in the lowest-numbered memory address, whereas the last word printed is the word located in the highest-numbered address. As with the "enter-keyboard-words" instruction, just described, a choice of any one of three decimal point positions is possible, with a comma being printed in the decimal point location, the actual "dot" designation being located in exactly the same position for every word printed out, regardless of the actual decimal point location.

In order to compose an instruction word which effectively instructs the computer to print out selected words stored in memory, a two-digit number is, in a sense, "written" for each of the five sections of the instruction word, as before. First, the code for POW (ie., 01) is placed in section 1 thereof; the motor bar code and decimal point code are placed in section 2, the same as for the EKW instruction; in section 3 is placed the address of the first word to be printed out; in section 4 is placed the address of the last word to be printed out; and in section 5 is placed the address of the next instruction in the program.

When the computer is in the process of carrying out a POW instruction, depression of the preselected motor bar is effected only while the carriage is positioned on a stop, or the "non-select" key is depressed. However, as soon as a POW instruction is presented to the computer during any part of the program thereof, depression of the preselected motor bar is automatically effected by the computer when the carriage arrives at the next stop, or if the carriage is already positioned on a stop. Hence, a numeral "3" in the low-order digit position of section 2 of the instruction word has no significance in a POW instruction for initiating an "automatic-resume-program" operation.

If it is desired that the computer automatically print out the three words that are stored in memory addresses 44 through 46, for example, with the decimal point located between amount columns #2 and #3, with an upper motor bar "touch" operation, and with the next instruction coming from address **88**, an instruction word capable of initiating this sequence of operations 01 00 44 46 88. In response to the POW instruction, the computer automatically prints out the words from memory addresses 44 through 46, one below the other in a single column, and at a speed of approximately one hundred words per minute. It is to be noted, however, that, when a word is read out of memory during a printing operation, the word nevertheless remains in that address after the printing operation, somewhat similar to sub-totaling a totalizer in the accounting machine portion.

When negative words are effectively printed out of memory, the computer utilizes the "X" totalizer to obtain the absolute value of the word prior to printing, just as the accounting machine utilizes the "X" totalizer to complement "negative" words stored in the accounting machine totalizers, as previously described. The negative words are printed out in red, with a "CR" symbol to the right thereof. Therefore, it is obvious that the "X" totalizer must contain a "zero" count at that point in the program when negative amounts are to be printed out. When printing out a negative word, the accounting machine automatically cycles twice, and the required motor bar need only be depressed once. Also to be noted is that, when a negative word is printed out of memory, all "hold" motor bar operations produce the same effect as a corresponding "touch" motor bar operation. It is to be further appreciated that, while a word is being printed out, the word also may be simultaenously added to or subtracted from any preselected one, or ones, of the totalizers in the accounting machine. There is, in a sense, no limit to the number of totalizers to which the word is added; however, in one cycle of operation, the word is subtracted from a maximum of only two totalizers.

30. *Format of Enter-Card-Words Instruction Word (ECW—02)*

As previously mentioned, the magnetic ledger cards utilized by the present computer are provided with a strip of saturable ferromagnetic material positioned along the leftmost side of the front face of the card and having a width of approximately one inch. A plurality of words are selectively and sequentially recorded from memory on the magnetic strip, under control of the computer, with the use of but a single instruction word, and, when the ledger card is thereafter mechanically fed into the card-handling portion of the computer, these words are effectively reproduced by the sensing mechanism thereof and again stored in memory.

The ECW instruction is similar to the EKW instruction in that it has what is known as a "string-of-addresses" format. Consequently, all the words recorded on the magnetic strip of a card are effectively stored in memory in sequential addresses with only one ECW instruction. It is to be noted that the computer is not responsive to an ECW instruction unless the carriage is in "home" position. However, if the previous instruction was a "record-on-card" instruction, which is to be described hereinafter, the carriage is thereafter in home position; if the carriage is not in home position, it is placed there either by program control or by manual manipulation thereof. When the computer starts to execute an ECW instruction, the carriage first opens, and the "end-card" lamp (EC, FIG. 1) is illuminated and thus indicates that the card-handling portion of the computer is ready to properly receive a ledger card.

After the ledger card has been aligned on the card loading table **232** (FIG. 1), the card is captivated, and the data magnetically recorded thereon is thereafter reproduced or "read" by the reading mechanism of the computer. During the process of reading the magnetically-recorded data, the card is physically translated by the card-handling mechanism past the read-write head assembly **343** (FIG. 19) and into the carriage of the accounting machine, as previously described. During the time the ledger is thereafter being driven outwardly from the carriage, the computer has the selective ability to either stop the card on the next posting line, or permit the card to be ejected.

If it is assumed that a "manual" mode of operation is specified, when the card is positioned on the next posting line, the carriage thereafter remains open to permit insertion thereinto of other record-keeping forms, if desired. After the form or forms have been inserted into the carriage, the resume-program-bar must be depressed to initiate the next cycle of operation. Consequently, the carriage closes, and the computer automatically operates the desired motor bar, if specified by the high-order digit located in section 2 of the instruction word.

If the desired sequence of operations does not necessitate the insertion of additional forms into the carriage, the computer is instructed to close the carriage and to operate the preselected motor bar immediately after the card has been automatically placed on the correct posting line. To accomplish this, an "automatic-resume-program line-find" code is placed in the low-order digit position of section 2 of the instruction word. Consequently, it is not necessary to thereafter depress the resume-program-bar to initiate a cycle of operation of the accounting machine. After the ledger card has been either placed on a posting line, or ejected, it is necessary for the carriage to be closed. If the card is positioned on a posting line, the carriage is closed either by depression of the resume-program-bar or by specifying an "automatic-resume-program" in the instruction word; if an "eject card" code ($x8$, Chart III) is specified and the string-of-addresses is filled, the card is ejected, and the carriage is closed automatically. After the carriage is closed, the motor bar code instruction, specified by the high-order digit of section 2 of the instruction word, is carried out, thereby completing the ECW operation.

The code for ECW—i.e., 02—is in section 1 of the instruction word; the motor bar and linefind codes are in section 2 thereof; in section 3 is the memory address into which the first word read from the card is to be stored; in section 4 is the memory address into which the last word read from a card is to be stored; and in section 5 is the memory address of the next instruction word.

The "line-find" code for an ECW instruction is as illustrated below and also as subsequently shown in Chart III.

$x0$—Enter new card with manual resume-program.
$x1$—Stop on first linefind with manual resume-program.
$x2$—Stop on second linefind with manual resume-program.
$x3$—Stop on third linefind with manual resume-program.
$x5$—Stop on first linefind with automatic resume-program.
$x6$—Stop on second linefind with automatic resume-program.
$x7$—Stop on third linefind with automatic resume-program.
$x8$—Eject card.

During normal operation of the computer, there may be as many as three areas on the face of the ledger card onto which posting is to selectively be made. By use of the proper line-find code in the low-order digit position of section 2 of the instruction word, the card is selectively positioned on the next posting line in any one of the three areas thereof. An "$x1$" or "$x5$" in the low-order digit position of section 2 causes the card to be positioned on the next posting line in the topmost area of the card; an "$x2$" or "$x6$" causes the card to be positioned for posting in the middle area; and an "$x3$" or "$x7$" causes the card to be positioned for posting in the lowermost area. This permits the words on the card to be grouped for simplicity of programming and mathematical operations. For example, a ledger card designed for payroll application usually has two posting areas whereby relatively fixed data, such as employee's clock number, number of dependents, earning rate, deductions, etc., are posted in an area at the top of the card, whereas employee's earnings, taxes, and deductions for each pay period are posted in the area of the main body of the card. In addition to being printed on the face of the card, this information is also magnetically recorded on the magnetic strip of the ledger card.

For payroll posting, for example, an "x2" or "x6" is placed in the low-order digit position of section 2 of the instruction word. However, if it is desired to change the fixed data at the top of the card, an "x1" or "x5" is placed in the low-order digit position of section 2 of the instruction word.

In a payroll application in which the words magnetically recorded on the ledger cards are indicative of an employee's earning record, the cards are sequentially fed into the card-handling portion of the computer and are posted one at a time. After the information recorded on each card is magnetically reproduced and stored in memory, the carriage of the accounting machine remains open to permit a pay check to be inserted, as before stated; when the resume-program-bar is depressed, the carriage closes. If it is desired that the middle motor bar is to be automatically actuated for a "touch" operation, if there are fifteen words to be magnetically read from the ledger card and stored in memory addresses 3ϕ through 44, and the next instruction word is stored in address 88, an instruction word capable of initiating this particular sequence of events is 02 22 30 44 88.

If it is assumed in the foregoing example that, when the carriage opened, there were no other forms to be inserted therein and it was desired for the carriage to close immediately after the card was automatically positioned with the next line thereof ready to be posted, an "x6" code is placed in the low-order digit position of section 2 of the instruction word, instead of an "x2," to effect automatic closing of the carriage and automatic actuation of the desired motor bar. An instruction word capable of initiating this particular sequence of events is 02 26 30 44 88.

An "x8" code in the low-order digit position of section 2 of an ECW instruction word causes the computer to magnetically reproduce the required number of words magnetically recorded on the ledger card strip and then immediately eject the card. This type of instruction word is used when it is desired to enter the information magnetically recorded on the ledger card into memory, but, where it is not desired to post the card. Thus the ledger card is ejected immediately after the reading operation is completed. If there were ten words previously magnetically recorded on the card which are to be stored in memory addresses 55 through 64, an instruction word capable of initiating reading and ejection of the card without a motor bar being actuated is 02 68 55 64 88. In this instance, the card is first magnetically read and ejected, and then the carriage immediately closes and the computer proceeds to the next instruction without a motor bar being depressed.

After a complete program is composed, a maximum number of ten of the instruction words, as composed, may be magnetically recorded on each of the ledger cards, if desired, and subsequently the program may then be stored in preselected memory addresses by sequentially reading and ejecting each of the ledger cards, in the manner previously described. A desirable feature of the present computer is that, by utilization of but a single ECW instruction word, an entire program may be stored in the computer memory, which program consists of any number of instruction words recorded on any number of ledger cards, assuming, of course, that the capacity of the memory is not exceeded. This is accomplished due to the fact that, if a card does not have a sufficient number of words magnetically recorded thereon to fill the remaining string of memory addresses as specified in the instruction word, the card is automatically ejected after it is read and the contents thereof stored in memory. Thus, by specifying in the instruction word an exact number of addresses to accommodate each and every instruction word in the program, the storage in memory of all of the instruction words is thereafter effected by only one ECW instruction word.

For an example, it is assumed that a particular accounting program consists of forty-five instruction words which are magnetically recorded on five ledger cards, four cards having ten instruction words magnetically recorded thereon, and one card having magentically recorded thereon the remaining five instruction words. If it is further assumed that the instruction words are to be stored in memory addresses ϕϕ through 44, an instruction word capable of initiating this particular sequence of events is 02 68 00 44 88.

In carrying out this instruction, the ten instruction words magnetically read from the first ledger card are stored in memory addresses ϕϕ through ϕ9. Thereafter, the ledger card is automatically ejected, as the ten instruction words magnetically recorded thereon were of insufficient number to fill the entire string of forty-five addresses. After the first ledger card is ejected, the information magnetically reproduced from the second ledger card is stored in addresses 1ϕ through 19, and so on for the third and fourth cards, until the five words magnetically recorded on the fifth card are stored in the addresses 4ϕ through 44, thus filling the entire string of addresses. Even though the contents of the fifth card fills the remaining string of addresses as specified in the instruction word, the last card is ejected also because of the "eject-card" instruction code (x8) which appears in the low-order digit position of section 2 of the instruction word. After the fifth card is ejected, if the computer is operating in an "automatic" mode, the carriage of the accounting machine closes, and the computer automatically proceeds to carry out the next instruction as specified by the instruction word stored in memory address 88 without a motor bar having been depressed. Due to the fact that the computer reads and stores the information magnetically recorded on the ledger card and then automatically ejects the card if the number of words recorded thereon is insufficient to fill the remaining string of addresses, the exact number of words recorded on the card must be known beforehand in order for an ECW instruction to properly initiate the desired sequence of operations. If there are a greater number of words magnetically recorded on the card than there are addresses remaining in the string of addresses, all the words are ignored that are magnetically read after the string of addresses is filled.

For some applications, it may be desired to utilize a ledger card which does not have magnetically recorded thereon any information whatever, possibly not even a "linefind" information, which is to be described in detail hereinafter. In this instance, a "new-card" instruction, having a code of "x0," is placed in the low-order digit position of section 2 of the instruction word to cause the instruction word to cause the ledger card to be magnetically "scanned," and to immediately be stopped after being scanned. Thereafter, the carriage is closed by depression of the resume-program-bar, and the accounting machine platen is then rotated manually to position the card on the correct posting line. Consequently, the addresses specified in section 3 and section 4 of the instruction word have no significance in this instance.

31. *Format of Record-on-Card Instruction Word (ROC—03)*

In the just-described payroll application, after each ECW operation is completed, during which all the amounts magnetically recorded on an employee's earning ledger card are automatically stored in the computer memory, the information is then used to compute the employee's taxes, deductions, and net pay. Thus, before the card is ejected, all year-to-date amounts are up-dated, and new year-to-date amounts and fixed data are recorded on the magnetic strip, thereby erasing the data previously recorded thereon. In addition, the new up-dated amounts are normally printed on the face of the ledger card.

The "record-on-card" instruction is quite similar to the "print-out-words" (POW) instructions, in that the amounts recorded on the magnetic strip of the ledger card are previously stored in sequential memory addresses. In an ROC instruction word, section 1 thereof contains the code for the ROC instruction, i.e., 03; section 2 contains the modifier code for the basic "record-on-card" operation; section 3 contains the address in memory of the first word in the string of addresses to be recorded on the ledger card; section 4 contains the address in memory of the last word of the string of addresses to be recorded on the ledger card; and section 5 contains the address in memory of the next instruction word.

For example, if ten amounts, or words, located in memory addresses 4 through 13, are to be magnetically recorded on the ledger card and the next instruction is located in memory address 88, an instruction word capable of initiating this particular sequence of operations is 03 00 04 13 88.

When the amounts are recorded on the magnetic strip of the ledger card, all non-significant zeros of the word are suppressed. Consequently, all zeros to the left of the most significant digit are replaced by one "end-of-word" symbol "*". Thus, the amount 1.95 is stored in memory as 0000000195 and yet is magnetically recorded on the ledger card as *195.

To be more fully described hereinafter, in accordance with the present invention, the magnetic strip on a ledger card thirteen inches in length is capable of having magnetically recorded thereon a maximum of approximately one hundred decimal digits, with an end-of-word symbol "*" occupying the same space as one decimal digit. Thus, if all non-significant zeros were not suppressed, it would not be possible to record more than ten words thereon if each word length was ten decimal digits. However, by suppressing all non-significant zeros, space is saved thereby and thus permits the possibility of more than ten amounts being recorded thereon. As a negative number is stored as its complement, there are no non-significant zeros in this instance. Consequently, it is necessary at all times to record all ten decimal digits thereof. It is to be noted, however, that, if an amount, whether positive or negative, is ten decimal digits in length, an end-of-word symbol is not recorded.

Before the computer is capable of initiating a recording operation on the ledger card, the carriage of the accounting machine portion thereof must be in "home" position, which, as previously mentioned, is approximately two inches from the extreme right-hand carriage position. Either a previous instruction in the program effects the moving of the carriage to home position upon completion thereof, or the carriage is manually translated thereto.

After a posting operation is completed, the ledger card is usually spaced upwardly by a distance of one posting line and thereafter magnetically recorded upon while being ejected. In addition, before the ledger card is ejected, the line on which the card was subsequently positioned is also recorded on the magnetic strip thereof. Consequently, when the ledger card is again utilized, it is automatically placed on this line for the next posting operation. This automatic placement of the card on the next posting line is herein called "linefinding," which is to be described in detail hereinafter.

There are occasions, however, when it is not desired to alter the linefinding information on the card. To prevent the linefinding information from being altered subsequently, the number "01" is placed in section 2 of the instruction word; when normal recording of the linefind information is desired on a single linefind card, the number "0" is placed in section 2 of the instruction word; and, with a multiple linefind card, "02" is placed in section 2 of the instruction word.

After the entire recording operation on the ledger card is completed and the card is ejected, the carriage remains in home position, with the paper guide open. However, if the next step in the program necessitates the carriage to be moved away from homeposition, the paper guide is first closed, either by an instruction involving a motor bar, or by manual depression of the carriage open-close key.

32. *Format of SHIFT Instruction Word (SHF—04)*

The "shift" instruction is utilized to effect the shifting of a word stored in memory by a predetermined number of decimal digits in either direction. Thereafter, the shifted word is selectively stored either in the same address or in another selected address in memory. The code for the SHF instruction—i.e., 04—is in section 1 of the instruction word; in section 2 of the instruction word is placed the "shift" and "direction" code; in section 3 is the address of the word to be shifted; in section 4 is the address in which the result is to be stored; and in section 5 is the address of the next instruction word. The "shift" and "direction" code for an SHF instruction is shown below, where "n" specifies the number of shifts from zero to nine.

00—No shift.
$2n$—Shift left.
$3n$—Shift right and round.
$4n$—Shift right.
$5n$—Shift right without preserving sign.

As an example, an instruction word capable of initiating the copying of a word from memory address 5ϕ into memory address 79, with the next instruction word located in memory address 81, is 04 00 50 79 81. Since a "00" code is placed in section 2 of the instruction word, the word stored in address 5ϕ is simply stored in address 79. After the instruction has been carried out, the same word appears in both of addresses 5ϕ and 79. The above instruction word printed out appears as 4,005,079.81. Thus, it is seen, an instruction word actually appears as a data word when printed out, with both types of words being stored in memory in exactly the same manner as previously mentioned. In fact, it will become more obvious hereinafter that an instruction word is capable of being arithmetically modified in the same manner as a data word.

In the previous portion of the description, each of the instruction words has been illustrated as consisting of five distinct groups of numbers of two decimal digits each. This was done in order to emphasize the separate functions of each of the five sections thereof. However, hereinafter, all instruction words will be designated in exactly the same manner as they appear while stored in memory.

In the above example, suppose that the word in address 5ϕ is 0000000155 and is to be shifted three places to the left, which, in effect, multiplies the word by one thousand, and the result is to be stored in address 79. In this instance, an instruction word capable of initiating this particular sequence of events is 0423507981. After being shifted three places to the left, the word subsequently stored in address 79 is 0000155000. However, suppose instead that the word in address 5ϕ is to be shifted one place to the right, which, in effect, multiplies the word by one tenth. In this instance, the instruction word is 0441507981. After being shifted one place to the right, the word subsequently stored in address 79 is, in this instance, 0000000015.

By placing the code designation "$3n$" in section 2 of the instruction word, the word being operated upon is first shifted to the right "$n-1$" times, and thereafter the word 0000000005 is added thereto to round off the low-order digit thereof. Following the addition, the word is shifted one more digit to the right. In the previous example, if the word initially stored in address

5φ (i.e., 0000000155) is to be shifted right two places and then rounded, an instruction word capable of initiating this sequence of events is 0432507981. The word in address 79, after being shifted and rounded, is now 0000000002.

When a negative word is shifted to the right in response to a "3*n*" or a "4*n*" code in section 2 of the instruction word, 9000000000 is added to the word each time it is shifted, in order to preserve its sign. However, with a "5*n*" code in section 2 of the instruction word, the word is shifted to the right without its sign being preserved, and, whether the word is positive or negative, a zero is effectively filled in as the tenth-order digit following each shift.

33. *Format of CLEAR-MEMORY-ADDRESSES Instruction Word (CMA—05)*

In a CMA instruction, use is again made of the ability of the computer to increment addresses. As a result, any number of addresses are cleared with the use of only one instruction word; when an address is cleared, a word of all zeros is stored therein. For a CMA instruction, the code therefor—i.e., 05—is in section 1 of the instruction word; two zeros are in section 2 thereof; in section 3 is the memory address of the first word in the string of addresses to be cleared; in section 4 is the address of the last word in the string of addresses to be cleared; and in section 5 is placed the address of the next instruction word to be acted upon.

As an example, an instruction word capable of initiating the clearing of memory addresses 1φ through 2φ, with the next instruction word being stored in memory address 9φ, is 0500102090.

34. *Format of MOTOR BAR Instruction Word (MB—06)*

The motor bar instruction effects the desired depression of a preselected motor bar to initiate a preselected cycle of operation of the accounting machine portion of the computer. In section 1 of a motor bar instruction word is the code number "06"; in section 2 are the motor bar and decimal point codes, the same as previously described with respect to the EKW instruction; zeros are in both section 3 and section 4; and in section 5 is the address of the next instruction word. In a MB instruction, an "*x*3" code in section 2 of the instruction word has no significance, as, in this instance, an "6*x*" code is placed therein to effect an "automatic-resume-program" operation.

If a normal "dollar" decimal point and a middle motor bar "touch" operation is desired, with the next instruction word being stored in memory address 66, an instruction word capable of initiating this particular sequence of events is 0620000066. Consequently, as soon as the carriage reaches a stop, either by tabulation, carriage return, or depression of the "non-select" key, the designated motor bar is operated, and the computer proceeds onward to carry out the dictates of the next instruction word.

35. *Format of STOP Instruction Word (STP—07)*

When a "stop" instruction is given to the computer, all operations therein effectively come to a standstill, and lamp HA (FIG. 1) is illuminated. However, when the resume-program-bar is depressed, lamp HA is turned off, and the computer proceeds to carry out the next instruction. The code for a "stop" operation—i.e., 07—is in section 1 of the instruction word; in sections 2, 3, and 4 thereof are zeros; and in section 5 is the address of the next instruction word.

If the next instruction word is located in memory address 5φ, an instruction word of this type is 0700000050.

36. *Format of ADD Instruction Word (ADD—08)*

In carrying out an "ADD" instruction, two words taken from memory are added and the sum stored in any specified address thereof. The two words are added in accordance with the laws of algebra whereby the addition of two positive words results in a positive sum; the addition of two negative numbers results in a negative sum; and the addition of a positive and a negative word results in a difference between the two words, the difference word having an algebraic sign of the operand having the larger absolute value.

In section 1 of an ADD instruction word is the code designation therefor, i.e., 08; in section 2 thereof is the memory address of the "addend"; in section 3 thereof is the memory address of the "augend"; in section 4 is the address in memory of the sum; and in section 5 is the memory address of the next instruction word to be carried out.

Thus, if the addend data word stored in memory address 17 is to be added to the augend data word stored in address 19, the sum data word is to be stored in memory address 21, and the next instruction word is located in memory address **22**, an instruction word capable of initiating this particular sequence of events is 0817192122. Since negative words are stored in memory as complements thereof, the addition of a positive word and a negative word, or the addition of two negative words, results in the correct sum thereof. For example, if +222 is to be added to −111, the addition operation carried out within the computer is 9999999889 plus 0000000222 equals 0000000111. If −222 is to be added to −111, the addition operation carried out within the computer is 9999999889 plus 9999999778 equals 9999999667, which is the complement of −333.

37. *Format of SUBTRACT Instruction Word (SUB—09)*

Like the "add" instruction, the "subtract" instruction has a four-address format and obtains the difference between any two words in accordance with the laws of algebra.

In section 1 of a "subtract" instruction word is the code therefor, i.e., 0.9; in section 2 thereof is the memory address of the "minuend"; in section 3 is the memory address of the "subtrahend"; in section 4 is the memory address of the "remainder"; and in section 5 is the memory address of the next instruction word.

Suppose that the word in address 19 is to be subtracted from the word in address 17, the remainder stored in address 21, and the next instruction word stored in address 22. In this instance, an instruction word capable of initiating this particular sequence of events is 0917192122. According to the laws of algebra, when a negative word is subtracted from a positive word, the absolute values of the two words are added, and the sign of their sum is positive. Thus, if the word 0000000222 is stored in address 17 and the complement of −444 is stored in address 19, the computation carried out within the computer is 9999999556 subtracted from 0000000222 results in a remainder of 0000000666 to be stored in address 21. If a positive word is subtracted from a negative word, the remainder is a negative word whose absolute value is the sum of the absolute values of the two operands. Thus, if the complement of −222 is stored in address 17 and the word 0000000444 is stored in address 19, the computation carried out is 0000000444 subtracted from 9999999778 gives a remainder of 9999999334 to be stored in address 21, 9999999334 being the complement of −666; if the complement of −222 is stored in address 17 and the complement of −444 is stored in address 19, the computation carried is 9999999556 subtracted from 9999999778 gives a remainder of 0000000222 to be stored in address 21; and, if the last-mentioned minuend and subtrahend are interchanged, the computation carried out is 9999999778 subtracted from 9999999556 gives a remainder of 9999999778 to be stored in address 21, 9999999778 being the complement of −222.

38. *Format of SUM Instruction Word (SUM—10)*

The "sum" instruction is another of the instructions which take advantage of the address-incrementing ability of the computer and is utilized to effect algebraic addition of a plurality of words located in a corresponding plurality of sequential addresses; thereafter, the total is stored in any preselected address in memory.

The code number for the SUM instruction—i.e.,10—is in section 1 of the instruction word; the memory address into which the total is to be stored is in section 2 thereof; the memory address of the first word in the series to be added is in section 3; the memory address of the last word in the series to be added is in section 4; and the memory address of the next instruction word is in section 5 thereof.

Suppose that all of the words located in addresses 8ϕ through 84 are to be added, the sum is to be stored in address 7ϕ and the next instruction word is stored in address 85. In this instance, an instruction word capable of initiating this particular sequence of events is 1070808485.

39. *Format of ADD-PAIRS-OF-NUMBERS Instruction Word (APN—11)*

By using the "add-pairs-of-numbers" instruction, from one to nine pairs of numbers are added together, and the sum resulting from each addition is returned to the address of the operand of the highest address in each pair.

In this instance, the code number "11" for the APN instruction is in section 1 of the instruction word; the low-order digit of the number in section 2 thereof specifies the number of additions to be performed and the high-order digit thereof is a zero; the memory address of the first word of the pair is in section 3 thereof, and, after each addition, this particular address is incremented by "one" to obtain the memory address of the first word of the following pair to be used in the subsequent addition; the memory address of the second word of the pair, which also corresponds to the memory address of the sum, is in section 4 of the instruction word, and, following the addition, the address in section 4 is decremented by "one" to obtain the address of the second word of the following pair of words, which address also corresponds to the memory address into which the sum derived from the following addition is to be stored; and in section 5 of the instruction word is the address of the next instruction word.

Suppose that "five" pairs of words are to be added, the first word of the first pair being stored in address 15, the second word of the first pair being stored in address 72, and the next instruction word being stored in address 88. An instruction word capable of initiating this particular sequence of events is 1105157288.

In carrying out the APN instruction, the sequence of events is as follows: The word stored in memory address 15 is added to the word stored in memory address 72, and the sum resulting from the addition is stored in memory address 72; thereafter, the address specified in section 3 of the instruction word is incremented by "one" and thus becomes "address 16," instead of address 15, and, simultaneously therewith, the address specified in section 4 of the instruction word is decremented by "one" and thus becomes "address 71" instead of address 72; thereafter, the word in address 16 is added to the word in address 71, and the sum resulting from this second addition is stored in address 71; thereafter, the addresses specified in section 3 and section 4 of the instruction word are respectively incremented and decremented by "one," and, thereafter, the word stored in address 17 is added to the word stored in address 7ϕ, and so on, until the word stored in address 19 is added to the word stored in address 68 and the sum resulting from the last addition is stored in address 68.

It is to be noted that, upon completion of the carrying out of the above-described APN instruction, the five sums, resulting from the addition of the five pairs of numbers, are respectively stored in memory addresses 68 through 72, while the first word of each of the five pairs of words respectively remain in a corresponding one of memory addresses 15 through 19.

The APN instruction may be utilized, for example, in a payroll application where it is desired to add an employee's deductions, such as "Community Chest," "Union Dues," "FICA," etc., to each of the respective weekly accumulations thereof. All of such additions are automatically performed with only one instruction word, due to the fact that the addresses in section 3 and section 4 of the instruction word are automatically modified by the computer during the execution of the instruction. Also, as will be seen later, due to the fact that an instruction word is stored in a "register" while the computer is executing the instruction, the original instruction word normally remains in memory even though the instruction specified by the instruction word has been previously carried out to completion.

40. *Format of MULTIPLY-DOLLAR-DECIMAL Instruction Word (MDD—12)*

There are two multiply instructions which are utilized by the present computer. The first multiply instruction, "multiply-and-store-dollar-decimal" (MDD), is a "fixed decimal point" multiplication. That is, two words having a maximum of ten decimal digits each are multiplied together to obtain a product thereof having a maximum of twenty decimal digits. After the multiplication, the product is shifted one decimal position to the right, and, effectively, "five cents" is then added thereto in order to round off the low-order digit thereof. Thereafter, the product is shifted one more decimal position to the right, and the ten low-order digits of the product are stored in the "product" address.

The numerical code for MDD—i.e., "12"—is in section 1 of the instruction word; the address of the multiplicand is in section 2 of the instruction word; in section 3 thereof is the address of the multiplier; in section 4 is the address of the product; and in section 5 is the address of the next instruction word.

If it is desired for the word in address 4ϕ to be multiplied by the word in address 5ϕ, the product is to be stored in address 95, and the next instruction word is stored in address 6ϕ. An instruction word capable of initiating this particular sequence of events is 1240509560.

According to the laws of algebra, the algebraic sign of the product depends on the sign of the two words multiplied. That is, the algebraic sign of the product of two positive or two negative words is positive, while the algebraic sign of the product of a positive word and a negative word is negative. The MDD instruction is a special form of the "multiply-and-shift-product" instruction (MUS), to be described next, due to the fact that it is possible to obtain essentially the same result with either instruction. However, upon completion of a MDD instruction, both factors are retained in their original addresses in memory. It is to be noted that in a MDD instruction, the address in section 3 of the instruction word should not also be the address into which the product is to be stored; otherwise the product is incorrect in this instance.

41. *Format of MULTIPLY-AND-SHIFT Instruction Word (MUS—13)*

The second multiply instruction, "multiply-and-shift-product," in contrast to the MDD instruction, initiates a variable-decimal-point type of multiplication. That is, the two words are first multiplied together to obtain the product thereof, which is stored in a twenty-digit register. Thereafter, while in the register, the product is shifted either to the left or to the right a predetermined number of decimal digit positions. After being shifted, the ten low-order digits of the shifted product are placed in the product address in memory.

It is to be noted that the absolute values of the two words, without regard to any decimal point locations, are multiplied together to obtain the product. Thereafter, the product is shifted either to the left or to the right by a maximum of nine decimal digit positions and then "rounded" if desired. As only the ten low-order digits of the shifted product are stored in memory, all digits above the ten low-order digits are essentially ignored.

In section 1 of an instruction word for initiating a MUS command is the code therefor, i.e., 13; in the high-low-order digital position of section 2 is the product "number-of-shifts" code; in the high-order digital position of section 2 is the product "direction-of-shift" code; in section 3 is the memory address of the multiplicand; in section 4 are the memory addresses of the multiplier and the product; and in section 5 is the memory address of the next instruction word.

Suppose that the word in address 78 (0222200000) is to be multiplied by the word in address 89 (0000000004), the product is to be shifted five places to the right, and the next instruction word is stored in address 99. Thus, an instruction word capable of effecting the carrying out of this particular command is 1345897899. The product, before being shifted in the direction of and by the number of places specified, is 00000000000888800000. After being shifted five places to the right, the product stored in address 89 is 0000008888.

Again, according to the laws of algebra, the algebraic sign of the product depends upon the sign of the two words being multiplied; the product of two positive or two negative words is positive, while the product of a positive word and a negative word is negative. It is to be noted that the positive or negative product stored in memory after the shifting operation is completed should not exceed the capacity of the computer memory. Even though the multiplicand may have essentially any value, the absolute value of the multiplier, however, should not be greater than 999999999. It is also to be noted that the multiplier is effectively "lost" following a MUS command. Therefore, to retain the multiplier, it is necessary that a "shift" instruction precede the MUS instruction in the program.

42. *Format of DIVIDE Instruction Word (DIV—14)*

The "divide" instruction is utilized to effect the dividing of one word by another in order to obtain a quotient thereof, with both the dividend and the divisor being treated as whole numbers during the division.

In section 1 of an instruction word for effecting a DIV mathematical operation is the code therefor, i.e., 14; in section 2 thereof is the memory address of the dividend; in section 3 is the memory address of the divisor; the memory address of the quotient is in section 4; and in section 5 is the memory address of the next instruction word.

As an example, if the word in address 43 is to be divided by the word in address 57, the quotient is to be stored in address ϕ2, and the next instruction word is stored in address 88. An instruction word capable of effecting the carrying out of this particular command is 1443570288. As before, the sign of the quotient is algebraically determined by the algebraic signs of the two operands; if the two operands are of the same algebraic sign, the quotient is positive, and, if the two operands are of dissimilar algebraic signs, the quotient is negative. It is to be noted that, essentially, there is no limit to the maximum value of the dividend (x). However, the high-order digit of the absolute value of the divisor (y) should be a "zero," and the absolute value thereof is 999999999. Also, in this instruction, the address of the dividend (x) should not be the same as that in which the quotient is to be stored; otherwise, the quotient is thereafter equal to 0010000000.

43. *Format of TAKE-ALTERNATE-INSTRUCTION-IF X≧Y Instruction Word (CFM—15)*

There are certain instances when it is desired for the computer to carry out an alternate instruction if a certain word in memory is equal to or greater than other particular words in memory. With the use of a CFM instruction, it is possible for the computer to algebraically compare the absolute magnitude of words located in two predetermined memory addresses, and to use either the memory address specified in section 4 or the memory address specified in section 5 of the instruction word for determining the next instruction word in the program, depending upon the result of the comparison.

In section 1 of the instruction word for effecting a CMF operation is the numerical code therefor, i.e., "15"; in section 2 thereof is the memory address of the first word (x) to be compared; in section 3 is the memory address of the second word (y) to be compared; in section 4 is the memory address of the next instruction word if the word in the memory address specified by section 2 is equal to or greater than the memory address specified by section 3; and in section 5 is the memory address of the next instruction word if the word in the memory address specified by section 2 is less than the word in the memory address specified by section 3 of the instruction word.

Thus, if the first comparison word (x) is in address 46, the second comparison word (y) is in address 5ϕ, the next regular instruction is in address 1ϕ, and the alternate instruction is in address 75. An instruction word capable of initiating such a CFM operation is 1546507510.

44. *Format of TAKE-ALTERNATE-INSTRUCTION-IF X≠Y Instruction Word (CFE—16)*

The CFE instruction is somewhat similar to the just-described CFM instruction, with the exception that the CFE instruction compares the two words only for equality.

In section 1 of an instruction word capable of initiating a CFE operation is the code "16"; in section 2 and section 3 thereof are the memory addresses of the two words to be compared; in section 4 is the memory address of the next instruction word if the two words are of unequal magnitude; and in section 5 of the instruction word is the memory address of the next instruction word if the two words are of equal magnitude. Thus, if the two words to be compared are located in memory addresses ϕ5 and ϕ6, the regular instruction word is located in memory address 1ϕ, and the alternate instruction word is located in memory address 78. An instruction word capable of initiating this type of operation is 1605067810.

45. *Format of ENTER-PUNCHED-TAPE Instruction Word (EPT—17)*

The "enter-punched-tape" instruction utilizes the ability of the computer to increment memory address designations in an instruction word. Consequently, by the use of but a single instruction word, any portion or even the entire capacity of the memory is selectively stored with information read from a punched tape. The particular type of tape reader utilized by the present computer is a well-known commercially-available photoelectric type such as that designated as Model 903 and at present manufactured by Potter Instrument Company, Inc. As the construction and mode of operation of such a tape reader are well known to those skilled in the art, a detailed description thereof is not deemed necessary for a full and complete understanding of the present invention. However, a brief description of the format of the decimal-coded digital information on the punched tape is given herein.

With reference to FIG. 88C, the paper tape used in conjunction with such a tape reader is effectively divided into eight equally-spaced "channels," exclusive of "clock" channel (CLK), which are disposed parallel with respect to each other and with respect to the length of the tape. A combination of perforations punched in predetermined ones of the channel positions, and in a line perpendicular to the length of the tape, collectively represent a particular decimal digit or symbol. For example, an absence of a perforation in each of channels **1** through **4** represents the decimal digit "zero," assuming, of course, that a perforation appears in the clock channel "CLK"; a single perforation in channel **1** represents the decimal digit "one"; a single perforation in channel **2** represents the decimal digit "two"; a single perforation in channel **3** represents the decimal digit "four"; a perforation in channel **1** and channel **2** collectively represent the decimal digit "three"; and so on, so that a perforation in channel **1** and a perforation in channel **4** collectively represent the decimal digit "nine." It is also to be noted that a perforation in channel **6** and a perforation in channel **8** collectively represent an "end-of-frame" symbol (EOF), and a perforation in channel **6** and a perforation in channel **7** collectively represent an "alternate-instruction" symbol (AI).

In the operation of some of the various types of conventional paper tape punches, an odd number of holes, exclusive of clock channel (CLK), is required to be perforated in the tape to collectively represent a data digit or a symbol digit. Thus, if a data digit or a symbol digit is represented by an even number of holes, as shown, a hole is additionally perforated in channel **5** in order that an odd number of holes collectively represent that particular digit; if the number of holes is already an odd number, an additional hole is not perforated in channel **5**. Thus, in the above tape, a hole is additionally perforated in channel **5** whenever the holes perforated in data channels **1** through **4** collectively represent either a "zero," a "three," a "five," a "six," or a "nine." Also, a hole is additionally perforated in channel **5** whenever the holes perforated in symbol channels **6** through **8** collectively represent either of the end-of-frame or alternate-instruction symbols.

Therefore, whenever a hole is perforated in channel **6**, as for symbols (EOF) and (AI), the digit is not considered as "data." Consequently, data punching errors are cancelled simply by punching a perforation in channel **6** along with each incorrectly punched data digit, and the computer is thereby instructed not to consider these data digits. A perforation in both channel **6** and channel **7** effectively causes the computer to immediately stop the tape when the tape reader "senses" the "alternate-instruction" symbol (AI) and to carry out the "alternate" instruction specified in the instruction word. A perforation in both channel **6** and channel **8** causes the computer to immediately stop the tape when the tape reader senses the end-of-frame symbol (EOF) and to carry out the next "regular" instruction specified in the instruction word. A "frame" may comprise from one to one hundred words ten decimal digits in length, with each word normally punched one digit at a time, starting with the high-order digit and ending with the low-order digit. However, each word is thereafter sequentially read one digit at a time, normally starting with the low-order digit and ending with the high-order digit.

In an instruction word capable of initiating an "enter-punched-tape" operation, the code designation therefor (i.e., "17") is in section 1 thereof; in section 2 is the "direction code"; in section 3 is the memory address into which is stored the first word read; in section 4 is the memory address of the alternate instruction; and in section 5 is the memory address of the next regular instruction.

The "direction code" in section 2 of an EPT instruction word determines the direction in which the tape is to be translated past the reading station. That is, a "00" causes the tape to be translated in a forward direction to permit the reading of both data and symbol digits, whereas a "01" causes the tape to be translated in a reverse direction to permit the reading of symbol digits only. Thus, the operation of an EPT command not only is dependent upon the "direction code" in section 2 of the instruction word but, in addition, depends upon the particular type of digits to be read from the tape; i.e., data or symbol. However, when an EPT instruction is first initiated, the particular digit which is located directly over the tape-reader head, as represented by a particular combination of perforations, as heretofore described, may be either a decimal digit from "0" through "9", an end-of-frame symbol (EOF), or an alternate-instruction symbol (AI).

Thus, suppose that the number "00" is in section 2 of the instruction word and a combination of perforations representing a decimal digit is located directly over the reading head at the beginning of the cycle of operation. In this instance, the tape is translated in a forward direction past the reading station, and the first ten decimal digits read are stored in memory at the address specified in section 3 of the instruction word; the address in section 3 being incremented each time a ten-digit word is stored in memory. Data words are sequentially read from the tape and stored in memory until an end-of-frame symbol (EOF) is read, after which the tape is stopped and thereafter remains in position for the next recorded digit to be read therefrom. In the event the number of digits in a given frame does not equal an integral number of ten-digit words, "zeros" are effectively "filled in" in the remaining high-order digital positions of the data word stored in the last one of the memory addresses used, thus completing the word. Thereafter, the computer carries out the dictates of the next instruction word stored in the memory address specified in section 5 of the instruction word. It is to be noted that the particular number of words, thus stored, is not determined by the instruction word. The number of words stored in memory in response to an "enter-punched-tape" instruction is determined solely by the number of words recorded on the tape between end-of-frame symbols.

If, in the preceding example, an alternate-instruction symbol digit (AI) was positioned over the reading head instead of a data digit, the tape is then indexed forward by a distance of one digit, and the computer thereafter carries out the dictates of the alternate instruction word which is stored in memory at the address specified in section 4 of the instruction word. In the preceding example, an end-of-frame symbol (EOF) represents an "illegal" instruction for the computer. Consequently, in this instance, the computer ignores that particular instruction and does not stop the tape.

If an "enter-punched-tape" command (EPT) is initiated with the number "01" in section 2 of the instruction word and with a data digit initially positioned over the reading head, the tape first starts to rewind—i. e., is translated in a reverse direction—and, consequently, the data information recorded on the tape is not read; thus no data is stored in memory from the tape. The tape continues to rewind until an end-of-frame symbol (EOF) has been read. When the end-of-frame symbol is read, the computer immediately begins to carry out the dictates of the next instruction word stored in memory at the address specified by section 5 of the instruction word. Thereafter, the tape continues to rewind until an alternate-instruction symbol is read, at which time the tape is automatically stopped.

Should an end-of-frame symbol digit be positioned over the reading head in the just-preceding example instead of a data digit, the tape starts to rewind, and the computer immediately begins to carry out the dictates of the next instruction word which is stored in memory at the address specified in section 5 of the EPT instruction word. When an alternate-instruction symbol (AI) is thereafter first read, the movement of the tape is automatically stopped. However, if an alternate-instruction symbol has been positioned over the reading head, the tape starts to rewind and is stopped only when the next alternate-instruction symbol is read subsequent to an end-of-frame symbol. When the "end-of-frame" symbol is read, the computer starts to carry out the dictates of the next instruction word which is stored in memory at the address specified in section 5 of the EPT instruction word, even though the tape continues to rewind under the control of the tape unit only, and is automatically stopped when the next alternate-instruction symbol (AI) is sensed, all end-of-frame symbols (EOF) essentially being ignored. It is to be noted that successive "enter-punched-tape" instructions should be separated by a minimum time lapse of approximately 3.5 milliseconds.

In the preceding example, in which the code number "01" is in section 2 of the instruction register instead of "00" and the tape is in the process of being rewound, if a subsequent "enter-punched-tape" (EPT) command is initiated before the rewinding operation is completed and the code number "00" is subsequently in section 2 of the instruction register, the computer essentially waits until the winding operation is completed before attempting to carry out the next succeeding "enter-punched-tape" command.

46. *Format Code Designations of Instruction Words*

As just described, the code designations for the eighteen instruction words are numbered from "00" through "17." As will be seen later, the computer is properly responsive to any one of the code designations "00" through "09" even though the high-order digit thereof is any one of the even-numbered digits 0, 2, 4, 6, or 8. Thus, for example, the code designation for an "add" instruction (i.e., "08") is effective in initiating the proper sequence of operations even though stored as "08," "28," "48," "68," or "88." Also, the computer is properly responsive to either of the remaining code designations "10" through "17" even though the high-order digit thereof is any one of the odd-numbered digits 1, 3, 5, 7, or 9. For example, the code designation for an "add-pairs-of-numbers" instruction—i.e., "11"—is effective in initiating the proper sequence of operations even though stored as "11," "31," "51," "71," or "91."

Following are three charts, respectively labeled I, II, and III, which give an abbreviated listing and description of the various instruction words and formats thereof as previously described.

CHART I

Instruction Word Format

Instruction	Sect. 1	Sect. 2		Sect. 3	Sect. 4	Sect. 5
EKW	00	(MB)c	(DP)c	X	Y	*
POW	01	(MB)c	(DP)c	X	Y	*
ECW	02	(MB)c	(ECW)c	X	Y	*
ROC	03	(ROC)c		X	Y	*
SHF	04	(SHF)c		X	Y	*
CMA	05	O	O	X	Y	*
MB	06	(MB)c	(DP)c	OO	OO	*
STP	07	O	O	OO	OO	*
ADD	08	X		Y	Z	*
SUB	09	X		Y	Z	*
SUM	10	Z		X	Y	*
APN	11	O	N	X	Y	*
MDD	12	X		Y	Z	*
MUS	13	(SHF)c		Y	X	*
DIV	14	X		Y	Z	*
CFM	15	X		Y	Z	*
CFE	16	X		Y	Z	*
EPT	17	(EPT)c		X	Z	*

Where the asterisk represents the address of the next regular instruction and ( )c represents the operational code designation for that particular command as given by Chart II.

CHART II

Instruction	Description of the Instruction
EKW	Enter the keyboard words into addresses X through Y.
POW	Print out the words from addresses X through Y.
ECW	Enter the words from the ledger card into addresses X through Y.
ROC	Record on the ledger card the words in addresses X through Y.
SHF	Shift the word in address X to address Y.
CMA	Clear memory addresses X through Y.
MB	Upper, middle, or lower motor bar touch or hold operation.
STP	Stop.
ADD	Add the word in address X to the word in address Y and store the sum in address Z.
SUB	Subtract the word in address Y from the word in address X and store the remainder in address Z.
SUM	Obtain the sum of the words in addresses X through Y and store the sum in address Z.
APN	Add the word in address X to the word in address Y and store the sum in address Y. Thereafter, decrement address Y, increment address X, and repeat the operation N number of times.
MDD	Multiply the word in address Y by the word in address X and store the product in address Z with a "dollar" decimal point location.
MUS	Multiply the word in address Y by the word in address X. Thereafter, shift the product as specified by SHFc and store the final product in address X.
DIV	Divide the word in address X by the word in address Y and store the quotient in address Z.
CFM	Carry out the alternate instruction stored in address Z if the word in address X is equal to or greater than the word in address Y.
CFE	Carry out the alternate instruction stored in address Z if the word in address X is not equal to the word in address Y.
EPT	Enter the words from the punched paper tape into memory, starting with address X, and take alternate-instruction stored in the address in section Z when an AI symbol is read from the tape.

CHART III

Motor Bar Operational Code	
0x	Upper Motor Bar "Touch."
1x	Upper Motor Bar "Hold."
2x	Middle Motor Bar "Touch."
3x	Middle Motor Bar "Hold."
4x	Lower Motor Bar "Touch."
5x	Lower Motor Bar "Hold."
6x	No Motor Bar (ECW Only).

Decimal Point Location Code	
x0	Between Columns 2 and 3.
x1	Between Columns 5 and 6.
x2	Between Columns 8 and 9.
x3	Automatic-Resume-Program.

Enter-Punched-Tape Operational Code	
00	Read Tape.
01	Reverse and Rewind Tape.

Enter-Card-Words Line-Find Operational Code	
x0	Enter New Card-Manual Resume Program.
x1	Stop on first Linefind-Manual Resume Program.
x2	Stop on second Line-find-Manual Resume Program.
x3	Stop on third Linefind-Manual Resume Program.
x5	Stop on first Linefind-Automatic Resume Program.
x6	Stop on second Linefind-Automatic Resume Program.
x7	Stop on third Linefind-Automatic Resume Program.
x8	Eject Card—Automatic Resume Program.

Record-On-Card Line-Find Operational Code	
00	Record Single Linefind.
01	Do Not Record Linefind.
02	Record Multiple Linefind.

Shift and Directional Operational Code	
00	No Shift (Copy).
2n	Shift Left.
3n	Shift Right and Round.
4n	Shift Right.
5n	Shift Right and Do Not Preserve Sign (SHF Only).

Where "n" is any number from 0 to 9 specifying the number of shifts.

47. *Instruction Execution Time*

The following formulas give the first-order approximations of the time required for the execution of each of the previously-described eighteen instructions:

(00—EKW) The computer starts to execute the next instruction 0.250 second before the end of the last machine cycle initiated by the EKW instruction, the machine cycle time being approximately 0.6 second. The total time (t) in milliseconds is given by the formula $t=350+600(Y-X)+(Y-X+1)t_h$, where X is the address in section 3 of the instruction word, Y is the address in section 4 of the instruction word, and t_h is the average time interval between depressions of the resume-program-bar.

(01—POW) The time required for POW is the same as that for EKW, except that t_h in this instance is the amount of time that the accounting machine printer is delayed while the carriage is moving from one stop to another.

(02—ECW) The total time required to read and "line-find" a ledger card thirteen inches in length varies from 1.5 seconds for a card on which posting is to take place on the topmost posting line thereof to 2.5 seconds for a fully posted card; the total time required to read and eject a ledger card thirteen inches in length is 2.8 seconds. For each inch decrease in length, the time requirement is decreased by approximately 0.2 second.

(03—ROC) The total time required to record on and then eject a ledger card thirteen inches in length varies from 1.5 seconds for a new card to 2.5 seconds for a fully posted card. The total time required for reading and recording on the ledger card varies from 3.0 seconds for a new card to 5.0 seconds for either a partially or a fully posted card.

(04—SHF) The time required for a shift instruction depends on the particular digits placed in section 2 of the instruction register; i.e., the direction of shift and the number of shifts required. If the digits in section 2 are ($2n$), ($4n$), or ($5n$), the total time (t) in milliseconds is given by the formula $t=5.12+8.08n$; if the digits are ($3n$), then $t=9.16+8.08n$; and, if the digits are (00), then t is equal to 5.12 milliseconds.

(05—CMA) The time in milliseconds (t) required for a CMA instruction depends upon the number of memory addresses to be cleared and is given by the formula $t=3.44+1.68(Y-X)$, where X is the address in section 3 of the instruction word and Y is the address in section 4 thereof.

(06—MB) The total time (t) in milliseconds required for an MB instruction is given by the formula $t=350+t_h$, where t_h is the time elapsed before the start of the machine cycle.

(07—STP) The time (t) in milliseconds required for a STP instruction is given by the formula $t=1.76+t_h$, where t_h is the time elapsed before the resume-program-bar is depressed.

(08—ADD) and (09—SUB) The time required for either an ADD or a SUB instruction is 9.16 milliseconds.

(10—SUM) The time (t) in milliseconds required for a SUM instruction depends upon the number of words to be totaled and is given by the formula

$$t=9.08+(Y-X)4.04$$

where X is the address in section 3 of the instruction word and Y is the address in section 4 thereof.

(11—APN) The time (t) in milliseconds required for an APN instruction depends upon the number of word-pairs to be added and is given by the formula

$$t=9.16+(n-1)7.40$$

where n is the number of pairs of words to be added.

(12—MDD) The time required for an MDD operation is calculated from the following equations for an MUS operation, except that, in an MDD operation, m is equal to "3" and n is equal to "2."

(13—MUS) The time required for an MUS instruction depends on the "shift" and "direction" operational code in section 2 of the instruction word, the algebraic signs of the multiplier X and the multiplicand Y, and the sum of the values of the digits of the multiplier. If the high- and low-order digits in section 2 of the instruction word are (m) and (n), respectively, the particular formula to be applied to obtain the time in milliseconds (t) required to carry out a MUS instruction depends upon the algebraic signs of the two operands in the following manner:

(a) If the signs of X and Y are both positive, and m is either a "2," a "4," or a "5," then

$$t=104.72+8.08n+4.04\sum_{i=0}^{9}X_i$$

$$\left(\sum_{i=0}^{9}X_i\right)$$

represents the sum of the digit values of the multiplier, X_0 being the lower-order digit of the multiplier, and X_9 being the high-order digit thereof. For example, if the multiplier is equal to (12.45), then

$$\left(\sum_{i=0}^{9}X_i\right)$$

is equal to (5+4+2+1+0+0+0+0+0+0), or (12). However, in the above example, if m was a "3," then

$$t=112.90+8.08n+4.04\sum_{i=0}^{9}X_i$$

(b) If the sign of X is negative and the sign of Y is positive, 12.12 milliseconds is to be added to the above time.

(c) If the sign of Y is negative and the sign of X is positive, 8.08 milliseconds is to be added to the above time.

(d) If the signs of X and Y are both negative, and m is either a "2," a "4," or a "5," then

$$t=112.90+8.08n+4.04\sum_{i=0}^{9}\left\|X_i\right\|$$

where $|X_i|$ represents a general digit of the number, regardless of whether the number appears in normal or complemental form. However, when (m) is a "3," then

$$t=120.98+8.08n+4.04\sum_{i=0}^{9}X_i$$

The maximum time occurs when X is equal to 89,999,999.99, the sign of Y is negative, and a shift of nine places to the right with "round-off" is desired. In this instance (t_{max}) is equal to .55 second. The minimum time occurs when (m) is a "2" or a "4," and (X) and (n) are both equal to zero, the minimum time being equal to 0.1 second.

(14—DIV) The time in milliseconds (t) required for a DIV operation depends upon the algebraic sign of the dividend and the divisor, and on the magnitude of the quotient. When the signs of X and Y are both positive, $t=48.28+t_d$; however, when the signs of X and Y are both negative, or of opposite sign, $t=56.28+t_d$, where t_d in each instance is equal to

$$\left(100Q_9+10Q_8+\sum_{i=0}^{7}Q_i\right)8.08$$

where (Q_i) is the individual digits of the quotient. For example, if the quotient Q is 000.4351000, then Q_0 through Q_2 are equal to "0," Q_3 is equal to "1," Q_4 is equal to "5," Q_5 is equal to "3," Q_6 is equal to "4," and Q_7 is equal to "0." Consequently,

$$\left(\sum_{i=0}^{7}Q_i\right)$$

in this instance is equal to (1+5+3+4), or (13). The maximum time required for a DIV operation is approximately six seconds.

(15—CFM) The time required for a CFM operation depends only on the algebraic signs of the two words to be compared. If X and Y have the same sign, t is equal to 9.16 milliseconds; however, if X and Y have opposite signs, t is equal to 5.12 milliseconds.

(16—CFE) The time required for a CFE operation is 7.48 milliseconds.

(17—EPT) The time in milliseconds (t) required for an EPT instruction depends upon the frame of data read and is expressed by the formula $t=5+1.66N$, where N is the number of digits per frame.

48. *Instruction Register Generally*

While the computer is in the process of carrying out the instruction specified by one of the just-described instruction words, that particular instruction word is effectively stored in an "instruction register." With reference to FIGS. 45A and 45B, the instruction register (i) is effectively divided into five sections, 1 through 5, where section 1 is capable of storing the two highest-order digits and section 5 is capable of storing the two lowest-order digits of the ten-digit instruction word. Each of the five sections is effectively divided into a high-order and a low-order digit-register, each of which is utilized to store a particular-order decimal digit of the instruction word in binary-coded-decimal form. For example, digit-register (19) effectively stores the highest or tenth-order decimal digit of the instruction word; digit-register (18) effectively stores the ninth-order decimal digit of the instructions word; and so on, so that digit-register (1ϕ) stores the lowest or first-order decimal digit of the instruction word.

49. *Section 1 of Instruction Register*

With reference to FIG. 55, section 1 of the instruction register is logically illustrated as including five flipflops, 6001 through 6005, whose states of energization collectively represent, in binary-coded decimal form, each of the previously-described instruction code numbers "$\phi\phi$" (EKW) through "17" (EPT). As the high-order digit of each of the instruction code numbers is either a "one" or a "zero," only one flipflop (6005) is necessary to store the high-order digit of the code. However, as the low-order digit of the code may be any of the digits "zero" through "nine," four flipflops (6001 through 6004) are necessary to represent the value of the low-order digit thereof.

As previously described with respect to FIGS. 67A and 86, output line MSA from the memory sense amplifier flipflop 6055 is either TRUE or FALSE, depending upon whether a binary "1" or a binary "0" had just previously been read out of memory during a read-write cycle of operation. If a binary "1" had just been read out of memory, the state of line MSA goes from FALSE to TRUE approximately 3 microseconds after TIME–3, and stays TRUE until the next TIME–2, at which time the state of line MSA returns FALSE. If a binary "0" had just been read out of memory, the state of line MSA remains unchanged; i.e., FALSE.

Thus, after presetting section 1 of the instruction register to "00" by effecting a TRUE-to-FALSE change of state of line $(PI\phi)'$ suppose that an ADD instruction word was originally stored in a particular address in memory and it is desired to take this word from memory and store it in the instruction register. As the code designation for an ADD instruction is "08," the digit to be stored in the high-order digit register (19) of section 1 of the instruction register is a "zero," and the digit to be stored in the low-order digit register (18) thereof is an "eight." Consequently, as binary-bit "a" of the low-order digit of the code designation is first to be read out of memory, line Ba is TRUE, and, as a word is to be transferred from memory to the instruction register, line MI is likewise TRUE, as will be seen later. However, in this example, as the binary value of bit "a" of the low-order digit of the code designation is "zero," line MSA remains FALSE, and therefore output lines $(18a)'$ and 18a are TRUE and FALSE, respectively. When bit "b" is to be read out of memory, only the line Bb from the bit counter is TRUE, while lines Ba, Bc, and Bd are FALSE. As the binary

value of bit "b" is likewise "zero," output lines $(18b)'$ and 18b remain TRUE and FALSE, respectively. When bit "c" is to be read out of memory, only line Bc from the bit counter is TRUE. As the binary value of bit "c" is also "zero," output lines $(18c)'$ and 18c remain unchanged. When the high-order bit "d" is to be read out of memory, only line Bd from the bit counter is TRUE. As the binary value of bit "d" is now a "one," when line C41 goes TRUE at TIME–4, all the inputs to logical AND 963 are TRUE, and, consequently, line BdL likewise goes TRUE at TIME–4. As the high-order digit of the code designation is next to be read in this particular example, only line D8 from the digit counter is TRUE at this particular instance. Therefore, the reference input to flipflop 6004 goes TRUE at TIME–4. At TIME–1, the reference input thereto goes FALSE, so that the states of output lines $(18d)'$ and 18d are reversed. That is, line $(18d)'$ goes FALSE, and line 18d goes TRUE.

It is seen, therefore, that the states of energization of flipflops 6001 through 6004, collectively represent the binary number "1000," which, in turn, represents the decimal digit "eight." As the value of bit "a" of the tenth-order digit is "0," the state of energization of flipflop 6005 represents the "zero" value of the tenth-order digit of the instruction word.

With reference to FIG. 56, all of the output lines from section 1 of the instruction register are connected as inputs to an "instruction-code readout" having a multiplicity of output lines which are selectively energized by logical combinations of the five flipflops of section 1 of the instruction register and individually correspond to a selected one of instruction words EKW(00) through EPT(17). Ignoring, for the moment, those output lines bearing a "prime" notation, only one of the remaining output lines is TRUE at any given instance; the particular line that is TRUE is dependent upon, and corresponds to, the particular code number stored in section 1 of the instruction register. There are, however, certain exceptions, which are to be noted hereinafter. Thus, in the preceding example, an "08" being stored in section 1 of the instruction register causes the state of output line ADD to be TRUE, all the remaining output lines being FALSE. It is, therefore, to be appreciated that, for an "enter-keyboard-words" command (code 00), only line EKW is TRUE; for a "print-out-words" command (code 01), line POW is TRUE; and so on, so that output line EPT is TRUE, indicative of the "enter-punched-tape" command (code 17). The simplified logical equations for the eighteen readout lines of the decoder are as follows:

Abb. of Command	Code No.	Logical Equation
EKW	00	=(19a)' (18d)' (18c)' (18b)' (18a)' ICR
POW	01	=(19a)' (18d)' (18c)' (18b)' 18a' ICR
EOW	02	=(19a)' (18c)' 18b (18a)' ICR
ROC	03	=(19a)' (18c)' 18b 18a ICR
SHF	04	=(19a)' 18c (18b)' (18a)' ICR
CMA	05	=(19a)' 18c (18b)' 18a ICR
M[illegible]	06	=(19a)' 18c 18b (18a)' ICR
STP	07	=(19a)' 18c 18b 18a ICR
ADD	08	=(19a)' 18d ICR
SUB	09	= 18a ADD
S[illegible]	10	= 19a (18d)' (18c)' (18b)' (18a)' ICR
APN	11	= 19a (18d)' (18c)' (18b)' 18a ICR
MDD	12	= (18a)' MUS
MUS	13	= 19a (18c)' 18b ICR
DIV	14	= 19a 18c (18b)' (18a)' ICR
CFM	15	= 19a 18c (18b)' 18a ICR
CFE	16	= 19a 18c 18b (18a)' ICR
EPT	17	= 19a 18c 18b 18a ICR

For reasons to be set forth hereinafter, it is to be noted that line ADD is TRUE whenever either an "08" or an "09" is stored in section 1 of the instruction register, i.e., for either an "add" or a "subtract" command;

however, line SUB is TRUE only for a "subtract" command, i.e., "09." In addition, line MUS is TRUE when either a "12" or a "13" is stored in section 1 of the instruction register, i.e., for either a "multiply-dollar-decimal" or a "multiply-and-shift" instruction; however, line MDD is TRUE only for a "multiply-dollar-decimal" command, i.e., code "12." An additional readout is provided by line MOD, which is essentially a logical OR of lines MUS and DIV and which signifies that either a multiplication or a division operation is being carried out.

50. *Section 2 of Instruction Register*

With reference to FIG. 57, section 2 of the instruction register includes eight flipflops, **6006** through **6013**, which store the seventh and eighth-order digits of the instruction word. As previously mentioned with respect to that part of the foregoing description relating to the "Instruction Word Format," section 2 of the instruction register is utilized to store the address of either the first or the second operands, the destination address, or a special code, depending upon the particular instruction. A decoder for the high-order digit-register of section 2 is provided with seven readout lines Hϕ through H6 to respectively indicate a digit having a value "zero" through "six" being stored in the high-order digit-register; the decoder for the low-order digit-register is provided with readout lines Lϕ through L3, which respectively indicate a digit having a value of "zero" through "three" being stored in the low-order digit-register.

For example, in an instruction calling for a particular motor bar operation of the accounting machine portion of the computer, readout lines Hϕ through H6 indicate which motor bar should be depressed and what type of depression is desired—i.e., "touch" or "hold"—in carrying out a "shift" instruction (SHF—**04**), and readout lines H2 through H5 indicate the direction and type of shift desired. The simplified logical equations for each of readouts Hϕ through H6 are as listed below. It is to be noted, however, as shown in the logical circuit diagram of FIG. 57, that readout lines H2 and H4 cannot go TRUE during a "multiply-dollar-decimal" instruction (MDD—**12**), while readout line H3 is permitted to go TRUE unconditionally.

$$\begin{array}{llllll}
H\phi = & (17d)' & (17c)' & (17b)' & (17a)' & \\
H1 = & (17d)' & (17c)' & (17b)' & 17a & \\
H2 = & & (17c)' & 17b & (17a)' & (MDD)' \\
H3 = & & (17c)' & 17b & 17a & +MDD \\
H4 = & & 17c & (17b)' & (17a)' & (MDD)' \\
H5 = & & 17c & (17b)' & 17a & \\
H6 = & & 17c & 17b & (17a)' &
\end{array}$$

Readout lines Lϕ through L3 not only are utilized in an "enter-card-words" instruction (ECW—**02**) but also are utilized in any of the instructions which call for a particular motor bar operation, such as "enter-keyboard-words" (EKW—**00**) and "print-out-words" (POW—**01**). For an EKW and a POW command, readout lines Lϕ through L3 selectively indicate the particular decimal point lamp on the keyboard that is to be illuminated and/or the particular "order-hook" solenoid that is to be energized. The simplified logical equations for readout lines Lϕ through L3 are as given below:

$$\begin{array}{lllll}
L\phi = & (16d)' & (16c)' & (16b)' & (16a)' \\
L1 = & (16d)' & (16c)' & (16b)' & 16a \\
L2 = & & (16c)' & 16b & (16a)' \\
L3 = & & (16c)' & 16b & 16a
\end{array}$$

It is to be noted that output lines **17***a* through **17***d* of the high-order digit-register of section 2 of the instruction register are respectively connected as an input to one of flipflops **6047** through **6050** of the high-order section of the word-selecting register previously described with respect to FIG. 64. It is also to be noted that output lines **16***a* through **16***d* of the low-order digit-register of section 2 of the instruction register are respectively connected as an input to flipflops **6051** through **6054** of the low-order section of the word-selecting register previously described with respect to FIG. 65, all of which may more clearly be seen by reference to the block diagram shown in FIG. 45. Thus, as lines **17***a* through **17***d* of FIG. 64 and lines **16***a* through **16***d* of FIG. 65 are each logically ANDED with line TI2, which originates in FIG. 61, after the word-selecting register is preset to "00" by effecting a TRUE-to-FALSE reversal of the state of line (PWϕ)', the high- and low-order digits stored in section 2 of the instruction register are permitted to respectively be simultaneously transferred directly into the high- and low-order digit positions of the word-selecting register by effecting selective energization of line TI2, selective energization of line TI2, therefore, effecting the transfer of the contents of section 2 of the instruction register into the word-selection register.

51. *Section 3 of Instruction Register*

As previously mentioned, section 3 of the instruction register normally stores either a destination address or the address of one of the operands. With reference to the left-hand portion of FIG. 58, section 3 of the instruction register comprises eight flipflops, **6014** through **6021**, flipflops **6014** through **6017** being a part of the high-order digit-register (I5) and flipflops **6018** through **6021** being a part of the low-order digit-register (I4) thereof, as illustrated in the block diagram of FIG. 45. The two digit-registers (I4) and (I5) are interconnected in such a manner that, when so combined, they function essentially as a binary-coded-decimal counter which is capable of effectively counting from 00 to 99 in units of "one."

To accomplish this, flipflops **6018** through **6021** of digit-register (I4) are connected together to collectively function as a "powers-of-two" binary counter that increments one unit each time the state of line **IN4** changes from TRUE to FALSE. After a count of "nine" (binary 1001) is reached, digit-register (I4) recycles back to "zero" (binary 0000) on the next change of state of line **IN4** from TRUE to FALSE. Flipflops **6014** through **6017** of digit-register (I5) are connected together to also collectively function as a "powers-of-two" binary counter that effectively increments by one unit each time digit-register (I4) recycles from "nine" back to "zero." Thus, it can be said, a four flip-flop powers-of-two forward-counting binary counter is one in which a change of state of the reference output of any or all of the four flipflops from a TRUE state to a FALSE state causes a corresponding reversal of the state of the next succeeding high-order flipflops. Therefore, each flipflop effectively "divides" the reference output of the succeeding low-order flipflop by a factor of "two."

With all four flipflops of each of digit-registers (I4) and (I5) connected in a conventional manner, each of the registers would normally count from "zero" (binary 0000) to "fifteen" (binary 1111) and then return to "zero" to again start the count cycle. However, in the instant case, digit-register (I4) includes a logical AND gate **1067** (FIG. 59), which detects a count of "nine" being held by digit-register (I4), a count of "nine" (binary 1001) being indicated by a TRUE state of each of the amplified reference outputs of flipflops **6018** and **6021**. Thus, it is evident that, when reference output lines **I4***a* and **I4***d* are both simultaneously TRUE, output line **I49** from gate **1067** (FIG. 59) is likewise TRUE each time the counter reaches a count of "nine," line (I49)', of course, being FALSE at that particular moment. As line (I49)' is FALSE only during a count of "nine," the state of the prime input to flipflop **6019** (FIG. 58) is prevented from being changed from TRUE to FALSE during the next succeeding TRUE-to-FALSE change of state of line **I4***a*. Consequently, lines **I4***b* and **I4***c* remain FALSE. At the same time, a logical AND (**1029**) of lines **I49** and **IN4** effectively returns the state of line **I4***d* back to FALSE, thus returning digit-register (I4) to "zero," whereby output lines I4a through I4d are FALSE, indicative of binary 0000. Each time the state of line I4d goes from TRUE to FALSE—i.e., when digit-register (I4) recycles—the state of flipflop **6014**, thence the state of output line I5a, is reversed thereby. It is to be noted, however, that digit-register (I5) is not recycled when a count of "nine" is reached therein, as in the case of digit-register (I4).

With reference to FIGS. 64 and 65, lines I5a through I5d and I4a through I4d of section 3 of the instruction register are each logically ANDED with line TI3 in such a manner that, after the word-selecting register has previously been preset to "00," the contents of section 3 of the instruction register is transferred to the word-selecting register by effecting a TRUE-to-FALSE change of state of line TI3.

52. *Section 4 of Instruction Register*

Section 4 of the instruction register, as previously mentioned, is utilized to store either the address of one of the operands, the address of the result, or the address of an alternate instruction. As shown in FIG. 58, and also in the block diagram of FIG. 45, the fourth section of the instruction register includes a high-order digit-register (I3) comprising four electrically-interconnected flipflops **6022** through **6025**, and a low-order digit-register (I2) likewise comprising four electrically-interconnected flipflops **6026** through **6029** Each of digit-registers (I2) and (I3) independently operates essentially as a power-of-two binary counter However, instead of being capable of counting in a forward direction as digit-registers (I4) and (I5) of section 3, digit-registers (I2) and (I3) are connected in such a manner as to collectively count in a reverse direction, in units of "one," from 99 to 00 for each succeeding TRUE-to-FALSE change of state of line DE2. Consequently, it can be said that each flipflop in such a forward-counting register is essentially a scale-of-two binary counter which effectively "divides" by two the reference output binary notation of a successive low-order flipflop, whereas each flipflop in such a reverse counting register effectively "divides" by two the prime output binary notation of a successive low-order flipflop.

With flipflops **6026** through **6029** of digit-register (I2) interconnected in a conventional manner to function as a reverse counter, digit-register (I2) would then normally count backwards in units of "one" until a "zero" count (binary 0000) is reached, and then would return to a count of "fifteen" (binary 1111) to start the count cycle again. However, in the instant case, digit-register (I2) includes a logical AND gate **1066** (FIG. 59), which detects a count of "zero" condition of digit-register (I2), a count of "zero" (binary 0000) being indicated by a FALSE state of each of output lines I2a through I2d from flipflops **6026** through **6029**, respectively, or by a TRUE state of each of prime outputs (I2a)' through (I2d)'. Thus, when all of lines (I2a)' through (I2d)' are simultaneously TRUE, output line I2φ from gate **1066** (FIG. 59) is likewise true each time the digit-register reaches a count of "zero," line (I2φ)' of course, being FALSE at that particular moment. As line (I2φ)' is FALSE only during the count of "zero," the state of the prime input to flipflop **6027** is prevented from being changed from TRUE to FALSE during the next succeeding TRUE-to-FALSE change of state of line (I2a)'. Consequently, the states of lines (I2b)' and (I2c)' remain TRUE. Simultaneously therewith, a logical AND of lines DE2 and I2φ (**1041**) effectively returns the state of line I2d back to TRUE, thus returning digit-register (I2) to a count of "nine," whereby output lines I2a and I2d are TRUE and lines I2b and I2c are FALSE, indicative of binary 1001. Each time the state of line (I2d)' changes from TRUE to FALSE—i.e., when digit-register (I2) recycles back to a count of "nine"—the state of flipflop 6022, thence the state of line I3a, is reversed thereby. It is to be noted that digit-register (I3) is not recycled when a count of "zero" is reached, as in the case of digit-register (I2).

With reference to FIGS. 64 and 65, the reference output lines from section 4 of the instruction register are each logically ANDED with line TI4 in such a manner that the contents of section 4 of the instruction register is transferred to the word-selecting register by effecting selective energization of line TI4, assuming, of course, that the word-selecting register is initially preset to "00."

53. *Section 5 of Instruction Register*

As previously mentioned, section 5 of the instruction register stores the address of the next regular instruction and, as shown in FIG. 59, comprises eight flipflops **6030** through **6037**, flipflops **6030** through **6033** being included in high-order digit-register (I1) and flipflops **6034** through **6037** being included in lower-order digit-register (Iφ). Digit-registers (I1) and (Iφ) are of conventional circuit configuration and indicate, in binary-coded decimal form, the value of the decimal digit stored therein, and, consequently, a more detailed description thereof is not deemed necessary.

With reference to FIGS. 64 and 65, output lines I1a through I1d and Iφa through Iφd from section 5 of the instruction register are each logically ANDED with line TI5 in such a manner that the contents of section 5 of the instruction register is transferred to the word-selecting register simply by effecting a TRUE-to-FALSE change of state of line TI5, assuming, of course, that the word-selecting register is initially preset to "00," as before stated with respect to sections 2 through 4 of the instruction register.

54. *Instruction Register Presets*

There are two "preset" lines for the instruction register, which are utilized to preset each section thereof to thereafter be representative of a specific two-digit numerical designation. For example, section 1 of the instruction register (FIG. 55) is preset to "00" by effecting a TRUE-to-FALSE change of state of line (PIφ)', which presets the state of output lines I9a and I8a through I8d FALSE, indicative of "00"; section 2 of the instruction register (FIG. 57) is simultaneously preset to "00" by line (PIφ)', which presets a FALSE state of output lines I7a through I7d, and I6a through I6d, also indicative of "00"; section 3 (FIG. 58) is also simultaneously preset to "00" by line (PIφ)', which presets the states of output lines I5a through I5d, and I4a through I4d, FALSE, indicative of "00"; each of the digit-registers of section 4 (FIG. 58) is, however, simultaneously preset by binary 1111 by a TRUE-to-FALSE change of state of line PIφ)', which presets the states of output lines I3a through I3d TRUE, indicative of binary 1111, and presets the states of output lines I2a through I2d TRUE, also indicative of binary 1111; and section 5 (FIG. 59) is simultaneously preset to "00" by line (PIφ)', which presets the states of output lines I1a through I1d and Iφa through Iφd, FALSE, indicative of "00."

From the foregoing, it is seen that selective energization of preset line (PIφ)' causes each section of the instruction register to be preset to "00," with the exception of section 4, each section of which is preset thereby to binary 1111. In other words, the reference outputs of all the flipflops in sections 1, 2, 3, and 5 of the instruction register are preset FALSE, whereas the reference outputs of the flipflops of section 4 thereof are preset TRUE by (PIφ)' prior to the "loading" of a word from memory into the instruction register. The reason for this type of preset is that, due to the previously-described decrementing ability of section 4, it has been found expedient, circuitwise, to first preset the reference output of each of the flipflops of section 4 to a TRUE representation and then reverse the state of each particular reference output whenever its corresponding binary bit from memory is "zero," contra to presetting the reference outputs FALSE and then reversing the state of each when the corresponding bit from memory is a binary "one." However, to avoid confusion, and for the purpose of convenience in the following description, the above presetting technique of the instruction register by line (PIϕ)′ will, nevertheless, be hereinafter referred to as presetting the instruction register to "zero."

With reference to FIG. 58, the second preset line PRB is a logical AND (1045) of lines SP and FG and is utilized to preset section 4 of the instruction register to "99" after the instruction register has previously been preset by line (PIϕ)′. The purpose for such a preset is to be covered hereinafter.

As previously described with respect to the memory sense amplifiers shown in FIG. 67 and the wave-form chart of FIG. 86, each time a binary "one" is read out of memory, the state of line MSA is changed from FALSE to TRUE at a time approximately three microseconds after TIME–3, and then its state returns to FALSE at the next succeeding TIME–2. Thus, with reference to FIG. 55, it is seen that from TIME–4 to the next succeeding TIME–1 during a "memory-to-instruction-register" word-cycle in which line MI is TRUE for approximately 1600 microseconds, line B*a*L is TRUE if the binary bit "*a*" just read out of memory is a "one," otherwise line B*a*L remains FALSE; during the next succeeding time interval of ten microseconds from TIME–4 to TIME–1, line B*b*L is TRUE if the binary bit "*b*" just read out of memory is a "one," otherwise line B*b*L remains FALSE; during the next succeeding time interval, line B*c*L is TRUE if binary bit "*c*" just read out of memory is a "one," otherwise line B*c*L remains FALSE; and, during the next succeeding ten-microsecond time interval from TIME–4 to TIME–1, line B*d*L is TRUE if binary bit "*d*" just read out of memory is a "one," otherwise line B*d*L remains FALSE.

As shown in FIG. 59, lines B*a*L through B*d*L are each individually ANDED with the low-order read-out line Dϕ of the previously-described digit counter (FIG. 63) and then respectively connected to the reference inputs of flipflops **6037**—**6034** of the low-order digit-register (Iϕ) of section 5 of the instruction register; lines B*a*L through B*d*L are also individually ANDED with the second-order read-out line D1 of the digit-counter and then respectively connected to the reference inputs of flipflops **6033**—**6030** of the high-order digit-register (I1) of the fifth section of the instruction register. Thus, if it is assumed that the instruction word stored in memory address ϕϕ is to be transferred into the instruction register, the state of flipflop **6037** is first "set" in a manner to be indicative of the magnetic state of core **885** (FIG. 52A), forty microseconds later the state of flipflop **6036** is "set" indicative of the magnetic state of core **886**, forty microseconds later flipflop **6035** is "set" indicative of the state of core **887**, and so on, until the first eight binary bits of the word in address ϕϕ are effectively stored in section 5 of the instruction register, the first four bits, of course, representing the low-order digit and the second set of four bits representing the second-order digit of the word in address ϕϕ.

Input lines B*a*L through B*d*L are similarly ANDED with the remaining output lines D4 through D9 from the digit counter to effect transfer of the corresponding digits of the word in memory and subsequent storage thereof in sections 1 through 3 of the instruction register. For example, in section 3 (FIG. 58) are stored the fifth- and sixth-order digits, in section 2 (FIG. 57) are stored the seventh- and eighth-order digits, and in section 1 (FIG. 55) are stored the ninth- and tenth-order digits of the word located in address ϕϕ. As previously described, the reference output from each of flipflops **6022** through **6029** (FIG. 58) of section 4 of the instruction register is initially preset TRUE just prior to a memory-to-instruction-register word transfer. Thus, if the binary bit just read out of memory is a "zero," line (MSA)′ remains TRUE, and, consequently, output line S4L is rendered TRUE from TIME–4 to TIME–1. Therefore, assuming that low-order bit "*a*" of the third-order digit is a "zero," the prime input to flipflop **6026** goes from TRUE to FALSE at TIME–1 approximately forty microseconds after the state of flipflop **6030** (FIG. 59) is "set" indicative of bit "*d*" of the second-order digit of the word in address ϕϕ; a TRUE-to-FALSE reversal of the prime input to flipflop **6026**, of course, "sets" the state thereof so that output line I2*a* is rendered FALSE, indicative of binary "zero." The remaining bits of the third and fourth order digits are sequentially stored in section 4 in the same manner as just described.

55. *Summary of Instruction Register*

In summary, the instruction register is provided with a binary "bit" serializer at the input of each digit-register thereof and also is provided with a serializer on a digit-by-digit basis for sequentially selecting each of the ten digit-registers thereof, starting with the lowest-order digit-register and ending with the highest-order digit-register thereof. Consequently, by means of a "memory-to-instruction-register" transfer instruction, the word stored in memory is transferred into the instruction register, digit by digit, starting with the low-order digit thereof, the low-order digit being stored in digit-register (Iϕ) of section 5 and the high-order digit being stored in digit-register (I9) of section 1 of the instruction register.

As previously described, and as illustrated in the block diagram of FIGS. 45A and 45B, the digit stored in a selected one of low-order digit-registers (Iϕ), (I2), (I4), or (I6) is selectively stored in the low-order digit-register (Wϕ) of the word-selecting register, and, simultaneously therewith, the digit stored in a selected one of high-order digit-registers (I1), (I3), (I5), or (I7) is selectively stored in the high-order digit-register (W1) of the word-selecting register. Additionally, the low-order digit of section 2 of the instruction register is selectively stored in an "R-counter," which is next to be described in detail.

56. *R-Counter*

With reference to FIG. 68, there is logically illustrated a reverse-counting counter, hereinafter termed "R-counter," which includes four interconnected flipflops **6057** through **6060**. The R-counter operates essentially in the same manner as low-order digit-register (I2) of section 4 of the instruction register, previously described in detail, in that it is a single digit counter capable of counting in a binary-coded-decimal code in a reverse digital order. For example, if the R-counter is initially at a count of "nine," on each occurrence of a TRUE-to-FALSE change of state of line DR, the R-counter is effectively decremented by one decimal digit until a count of "zero" is reached. However, as no recycling of the R-counter is provided, on the next subsequent change of state of line DR the R-counter is returned to a count of binary 1111, instead of binary 1001 as in digit-register (I2). The two presets provided for the R-counter are preset line PRϕ, which presets the counter to "zero" by causing the states of flipflops **6057** through **6060** to be such that each of the reference outputs thereof is FALSE (hereinafter known as setting a flipflop "FALSE"), and preset line PR8, which presets the counter to "eight" by setting flipflop **6060** TRUE and setting flipflops **6057** through **6059** FALSE.

Any single decimal digit, from "zero" to "nine," is selectively loaded into the R-counter from two different sources. For example, the digit stored in the low-order digit-register (I6) of section 2 of the instruction register is transferred into the R-counter by means of selective energization of line RLR. However, prior to loading, all four flipflops **6057** through **6060** of the R-counter are first preset FALSE by means of preset line PRϕ. Thereafter, the state of each flipflop is selectively set by line RLR to correspond to the state of the corresponding flipflop in digit-register (I6) of the instruction register. Thus, if digit-register (I6) were storing the numeral "six," so that flipflops **6011** and **6012** (FIG. 57) were TRUE (flipflops **6010** and **6013** being FALSE), when the state of line RLR is rendered FALSE after being TRUE for at least twenty microseconds, both of flipflops **6058** and **6059** (FIG. 68) are simultaneously set TRUE. Therefore, it is evident that any digit from "zero" to "nine" is selectively transferred from digit-register (I6) to the R-counter.

After the R-counter has previously been preset to "zero" by line PRϕ, any digit being read out of memory is selectively stored in the R-counter by effecting sequential energization of line RLM simultaneously with one of output lines B*a* through B*d* from the bit-counter (FIG. 62). Thus, the bit-counter operates as a so-called "serializer" at the input of the R-counter to effect selective setting of the states of the four flipflops therein to correspond to the four binary bits being sequentially read out of memory and corresponding to the particular memory digit to be stored in the R-counter.

The R-counter is used in various of the instructions for different purposes. For example, in a "shift" instruction, the R-counter is first loaded with the low-order digit stored in section 2 of the instruction register, which digit specifies the number of places the word in memory is to be shifted. After the entire word has been shifted one digital order, the R-counter is effectively decremented by one decimal digit by line DR and then is essentially "checked" to determine if the R-counter is at a count of "zero." In that way, the R-counter is used to serve as an indication as to the remaining number of digital orders the word is to be shifted. In a division operation, suppose that it is necessary to perform a series of subtraction operations during which eight shift operations of both the quotient and the remainder is required. In this instance, the R-counter is initially preset to a count of "eight" and thereafter decremented by one decimal digit after each shifting operation is completed, until a count of "zero" is reached. In other words, the R-counter counts downwardly from "eight" and thus keeps a record of the number of shifts remaining. In a "multiply-and-shift" instruction, the R-counter is sequentially loaded with each one of the digits of the multiplier word in memory. The R-counter then controls the number of times that the multiplicand is added into an accumulating register, and signifies when the required numbers of additions have been performed. In a "multiply-and-shift" or a "multiply-dollar-decimal" instruction, the R-counter is utilized to record the number of times that the product has been shifted, and, as a result, determines the number of shifting operations remaining to be completed.

The R-counter is capable of providing three logically-derived readout signals from output lines Rϕ, R1, and R8. Among other purposes, the states of readout lines Rϕ and R1 are respectively utilized during a "shift" operation to indicate when the R-counter is at a count of "zero" or "one"; the state of readout line Rϕ is utilized during both multiply instructions to indicate a "zero" count of the R-counter; and readout line R8 is utilized during all multiply and divide instructions, during which time the R-counter is effectively recording the progress of the instruction as it is being carried out.

57. *F-Counter*

With reference to FIG. 72, there is logically illustrated a single-digit forward-counting counter, hereinafter termed "F-counter," which includes four electrically-interconnected flipflops **6073** through **6076**. The specific mode of operation of the F-counter is essentially the same as that of the previously-described high-order digit-register (I5) of section 3 of the instruction register (FIG. 58), and, consequently, a detailed description is not deemed necessary for a full and complete understanding thereof. Like digit-register (I5), the F-counter is capable of counting from "zero" to "fifteen" and then recycling to a count of "zero."

Prior to a mathematical computation, the F-counter is generally preset to "zero" by means of a FALSE-to-TRUE change of state of preset line PFϕ. Thereafter, on each successive TRUE-to-FALSE change of state of line AF, the F-counter is incremented from a "zero" count and advanced by one decimal digit for each change of state of line AF. It is to be noted, however, that selective energization of line PFϕ presents the F-counter back to "zero" regardless of the count held therein. The F-counter is provided with four readout lines, Fϕ, F8, F9, and F1ϕ, which respectively indicate a count of "zero," "eight," "nine," and "ten" thereof.

58. *Compare F-Counter and Digit-Counter*

With reference to the upper right-hand portion of FIG. 72, the states of output lines F*a* through F*d* and (F*a*)′ through (F*d*)′ from the F-counter are logically compared with respect to the states of output lines D*a* through D*d* and (D*a*)′ through (D*d*)′ from the digit-counter, previously described with respect to FIG. 63. Thus, the state of output line (IFD)′ is TRUE whenever the digital count of the F-counter is equal to the digital count of the digit-counter; however, the state of output line (IFD)′ is FALSE as long as there is inequality between the contents of the F-counter and the digit-counter.

59. *"J" Digit-Register*

With reference to the lower right-hand portion of FIG. 71, there is logically illustrated a digit-register, hereinafter termed "J" digit-register, which includes four electrically-interconnected flipflops **6067** through **6070** collectively capable of storage of any number from binary 0000 through and including binary 1111. The "J" digit-register is selectively preset to "zero" by effecting a TRUE-to-FALSE change of state of input line PJϕ. Thereafter, the states of flipflops **6067** through **6070** are selectively conditioned to collectively represent either of decimal digits "one" or "five" by effecting a TRUE-to-FALSE change of state of a corresponding one of input lines PJ1 and PJ5.

The "J" digit-register is capable of being loaded, via line JL, with any digit previously stored in any one of memory addresses ϕϕ through 99, or addresses A or B, or, as will be seen later, with a digit magnetically read from a ledger card. However, prior to loading the "J" digit-register, each of the flipflops therein is effectively preset to "zero." Thereafter, the states of flipflops **6070**, **6069**, **6068**, and **6067** are sequentially conditioned to respectively correspond to a different one of the four binary bits which are sequentially read from memory or the ledger card in the following manner: As illustrated in FIG. 71, output line MSA from the memory sense amplifier flipflop **6055** (FIG. 67) for addresses ϕϕ through 99, output line ASA from the memory sense amplifier flipflop **6056** for addresses A and B (FIG. 67), and output line CSA from the ledger card sense amplifier flipflop **6144** (FIG. 85), essentially are each logically ANDED, via line JL, with each of output lines B*a* through B*d* from the bit-counter (FIG. 62). Consequently, if bit *"a"* of the digit is a binary "one," the state of the reference output of flipflop **6070** is rendered TRUE; if bit *"b"* is a binary "zero," the state of the reference output of flipflop **6069** remains FALSE, and so on.

Logically illustrated directly above flipflops **6067** through **6070** are three additional readout lines Jϕ, J9, and EOW, whose individual state of energization is indicative of a particular number stored in the "J" digit-register. For example, the state of output line Jϕ is rendered TRUE whenever a binary 0000 is stored in the register; the state of output line J9 is rendered TRUE whenever a binary 1001 is stored therein; and a TRUE state of output line EOW is indicative of a binary 1111 being stored therein. In addition to the availability of a parallel type of read-out, the output of the "J" digit-register is also effectively serialized, in that output lines Ja through Jd thereof are logically ANDED with the bit-counter output lines Ba through Bd via logical AND gates **1432** through **1435**, line Ja being ANDED with line Ba, line Jb being ANDED with line Bb, and so on. The outputs of logical AND gates **1432** through **1435** are logically ORED together and terminate at output line JS. Output line JS, for example, is logically connected as an input line to the "write-control" circuitry (FIG. 60), so that the digit in the "J" digit-register is capable of being selectively stored in any one of memory addresses φφ through 99, or addresses A or B.

60. *Compare F-Counter and "J" Digit-Register*

With reference to FIG. 72, a compare circuit is illustrated as comprising eight logical AND gates **1462** through **1469**, which are logically ORED together and terminate at output lines IFJ and (IFJ)′. The output lines from the "J" digit-register (FIG. 71) and the output lines from the F-counter (FIG. 72) are logically connected to selected ones of AND gates **1462** through **1469** in such a manner that the value of the digit stored in the F-counter is effectively compared with the value of the digit stored in the "J" digit-register. When the two digits are of unequal magnitude, the state of output line IFJ is rendered TRUE. However, when the value of the two digits is of equal magnitude, the state of output line (IFJ)′ is rendered TRUE, indicative of equality.

61. *"K" Digit-Register*

In FIG. 70 there is logically illustrated a second digit-register, hereinafter termed "K" digit-register, comprising four electrically-interconnected flipflops **6062** through **6065**, which are collectively capable of storage of any number from binary 0000 through and including binary 1111. The "K" digit-register is preset to "zero" by effecting selective energization of line PKφ. Thereafter, the register is selectively preset to binary 1001 by effecting a TRUE-to-FALSE change of state of line PK9.

Like the previously-described "J" digit-register, the "K" digit-register is capable of being loaded with any digit previously stored in any one of memory addresses φφ through 99, or one of addresses A or B. However, prior to loading, the register is preset to "zero." Thereafter, the states of flipflops **6065**, **6064**, **6063**, and **6062** are sequentially conditioned in the following manner to respectively correspond to a different one of the four binary bits which are sequentially read from memory: Output line MSA from the memory sense amplifier flipflop **6055** for addresses φφ through 99 (FIG. 67) and output line ASA from the memory sense amplifier flipflop **6056** for addresses A and B (FIG. 67) are each essentially logically ANDED, via line KL, with each of the output lines Ba through Bd from the bit-counter (FIG. 62). Consequently, if bit "*a*" of the digit just read out is a binary "one," the state of the reference output of flipflop **6065** is rendered TRUE by a change of state of the flipflop; if bit "*b*" is a binary "zero," the state of the reference output of flipflop **6064** remains FALSE, and so on.

Logically illustrated directly above flipflops **6062** through **6065** is an additional readout line Kφ, whose TRUE state is indicative of binary 0000 being stored in the "K" digit-register. In addition to the availability of a parallel type of read-out, the output of the "K" digit-register is also effectively serialized, in that output lines Ka through Kd thereof are each logically ANDED with a corresponding one of bit-counter output lines Ba through Bd, via logical AND gates **1393** through **1396**, line Ka being ANDED with line Ba, line Kb with Bb, etc. The outputs of AND gates **1393** through **1396** are logically ORED together and terminate in output line KS, which, in turn, is logically connected as an input line to the "write-control" circuitry (FIG. 60) in such a manner that the digit stored in the "K" digit-register is permitted to be selectively stored in any preselected one of the memory addresses.

62. *Adder-Subtracter*

Before proceeding with a detailed description of the adder-subtracter portion of the computer logically illustrated in FIGS. 53 and 54, a brief description will first be given, setting forth the various salient operations which are sequentially executed in the performance of a simple addition and subtraction mathematical computation; such a description is deemed desirable in expediting and assuring a clear understanding of the various principles involved in the construction and mode of operation of the adder-subtracter unit.

Preceding an "add" instruction (ADD—**08**), for example, both the addend and augend data words are first stored in memory. To begin the "add" instruction, one of the data words is transferred to address-A, with the second data word remaining in one of memory addresses φφ through 99. Thereafter, the low-order decimal digit of the word in address-A is read out and stored in the "J" digit-register via line ASA, as illustrated in the block diagram of FIG. 45. Simultaneously therewith, the low-order decimal digit of the word remaining in memory is read out and stored in the "K" digit-register via line MSA. The output lines from the "J" and "K" digit-registers are connected as input lines to the adder-subtracter unit in a manner such that the two decimal digits stored therein are effectively added together by the adder-subtracter unit, so that, if the sum is equal to "nine" or less, a "sum" digit and a zero "carry" digit are derived therein. However, if the sum is greater than "nine," a "sum" digit and a "carry" one digit are derived.

Following the addition of the two low-order digits, the sum digit is stored in the low-order digital position of address-A, and the carry digit is stored in a carry flipflop. Next, the second-order digit of the word in address-A is stored in the "J" digit-register, and, simultaneously therewith, the second-order digit of the word remaining in memory is stored in the "K" digit-register. Thereafter, both digits are, in a sense, simultaneously transferred to the adder-subtracter unit, wherein they are added together with the carry digit previously stored in the carry flipflop. If the sum of the two second-order digits plus the carry digit is greater than "nine," a sum digit is obtained and stored in the second-order digital position of address-A, and a carry "one" digit is obtained and stored in the carry flipflop. This sequence of operations is repeated for each digital order of the words until each digit of one of the data words is added to the corresponding digit of the other data word. Following the addition, the sum thereof is located in address-A. Consequently, the final step in the "add" instruction is to store the sum data word in the particular address in memory as previously specified.

A "subtract" instruction (SUB—**09**) is carried out in essentially the same manner, with the exception that the data word remaining in memory is subtracted from the data word in address-A, rather than being added thereto. More specifically, the first step to be carried out for a "subtract" instruction is to transfer to address-A the minuend which is previously stored in one of memory addresses φφ through 99. As before, the first-order digit of the word in address-A is stored in the "J" digit-register, and, simultaneously therewith, the first-order digit of the data word remaining in memory is stored in the "K" digit-register. Thereafter, the digit in the "K" digit-register is subtracted from the digit in the "J" digit-register, a difference digit is derived therefrom and stored in the low-order digital position of address-A, and the carry digit derived therefrom is stored in the carry flipflop, as before.

Due to the fact that there is no "borrowing" of digits in the present type of computation, the carry digit is added to the next higher-order digit stored in the "K" digit-register rather than being subtracted from the corresponding next higher-order digit in the "J" digit-register. In other words, any carry "one" digit effectively increments the next succeeding digit in the "K" digit-register rather than decrementing the next succeeding digit in the "J" digit-register.

The just-described subtraction operation is sequentially repeated for each of the next successively higher-order digits until the remainder derived from the subtraction operation is in address-A. As in the "add" instruction, the final step in the "subtract" instruction is to copy the remainder data word into the particular address in memory, as previously specified.

From the foregoing, it is evident that the adder-subtracter unit is capable of adding two decimal digits and a carry digit, either "zero" or "one," and to obtain therefrom a decimal sum digit and a new decimal carry digit. The adder-subtracter is also capable of subtracting one decimal digit, plus a carry digit of either "zero" or "one," from a second decimal digit and of obtaining therefrom a decimal remainder digit and a new decimal carry digit.

Essentially, the adder-subtracter unit is an "adder" and a "subtracter" combined into one circuit-sharing unit, the adder portion being selectively capable of individually performing the necessary arithmetic computation whenever an addition operation is required, and the subtracter portion being selectively capable of individually performing the necessary arithmetic computation whenever a subtraction operation is required.

As previously mentioned, the adder-subtracter unit is capable of sequentially performing an addition operation with respect to two binary-coded decimal digits and a decimal carry digit of "0" or "1" magnitude. The magnitude of the sum, derived from such an addition operation, may be as large as "19" (binary 10011). For example, if the digit "9" is added to the digit "9" plus a carry of "1," the sum thereof is "19." Consequently, upon completion of each addition operation, the binary form of the sum is essentially "checked" to determine if the magnitude thereof is equal to decimal "10" or more. If the sum has a magnitude of decimal "10" or more, the amount binary 1010 is effectively subtracted therefrom to obtain a binary-coded sum digit, plus a decimal "1" carry digit. However, if the sum is less than decimal "10," the single sum digit represents the actual sum, and, thus, a carry "0" digit is generated.

In order to add the four binary bits of a digit to the four binary bits of a second digit, corresponding bits of each digit are added together with the previous carry bit, and the carry bit derived from such addition is added to the next successively high-order bit of that particular digital group. The carry bit resulting from the highest-order bit addition of that digital group is added to the low-order bit of the next succeeding digital group. For example, suppose that the digit "5" (binary 0101) is to be added to the digit "4" (binary 0100), the rules for binary addition, of course, being: binary "0" plus binary "0" equals binary "0"; binary "1" plus binary "0" equals binary "1"; binary "0" plus binary "1" equals binary "1"; and binary "1" plus binary "1" equals binary "0" plus a binary "1" carry. Applying the just-mentioned rules for binary addition, binary 0101 plus binary 0100 equals binary 1001, or "9." By the same token, the decimal digit "5" plus the decimal digit "5" equals binary 1010. However, if decimal "10" (binary 1010) is subtracted therefrom, "5" plus "5" also equals binary 0000 plus a binary "1" carry.

Listed below, in "truth-table" form, are representations of input bits "*a*" through "*d*" for each of the digits previously stored in the "J" and "K" digit-registers, together with the carry bit, and the desired output bit for each additive combination thereof, TRUE being denoted by "1" and FALSE being denoted by "0," as before. In this instance, "J*a*" and "K*a*" are representative of the "*a*" bits of the digits in "J" and "K" digit-registers, respectively; "J*b*" and "K*b*" are representative of the "*b*" bits; "J*c*" and "K*c*" are representative of the "*c*" bits; and "J*d*" and "K*d*" are, respectively, representative of the "*d*" bits of the digits in the "J" and "K" digit-registers. "AN*a*" is representative of the binary sum of the "*a*" bits plus the previous digit carry; "PCI" is representative of the previous digit carry; "*pc*" is representative of the bit carry derived from the binary sum of the "*a*" bits plus the previous "*t*" digit carry; "*q*" is representative of the binary sum of the "*b*" bits plus the previous bit carry; "*qc*" is representative of the bit carry derived from the binary sum of the "*b*" bits plus the previous carry bit; "*r*" is representative of the binary sum of the "*c*" bits plus the previous carry bit; "*rc*" is representative of the bit carry derived from the binary sum of the "*c*" bits plus the previous carry bit; "*s*" is representative of the binary sum of the "*d*" bits plus the previous carry bit; and "*t*" is representative of the bit carry derived from the binary sum of the "*d*" bits plus the previous carry bit.

(*a*) Bit Addition					(*b*) Bit Addition					(*c*) Bit Addition					(*d*) Bit Addition				
Input			Output		Input			Output		Input			Output		Input			Output	
J*a*	K*a*	PCI	AN*a*	*pc*	J*b*	K*b*	*pc*	*q*	*qc*	J*c*	K*c*	*qc*	*r*	*rc*	J*d*	K*d*	*rc*	*s*	*t*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0
1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0
1	1	0	0	1	0	0	1	1	0	1	1	0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0	0	1
0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1
1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Reducing the above truth tables to the "canonical" form of the logical equations thereof in a well-known manner:

$$(1)\quad ANa=(Ja)'Ka(PCI)'+Ja(Ka)'(PCI)'+(Ja)'(Ka)'PCI+JaKaPCI$$

$$(2)\quad pc=JaKa(PCI)'+(Ja)'KaPCI+Ja(Ka)'PCI+JaKaPCI$$

$$(3)\quad q=(Jb)'Kb(pc)'+Jb(Kb)'(pc)'+(Jb)'(Kb)'pc+JbKbpc$$

$$(4)\quad qc=JbKb(pc)'+(Jb)'Kbpc+Jb(Kb)'pc+JbKbpc$$

$$(5)\quad r=(Jc)'Kc(qc)'+Jc(Kc)'(qc)'+(Jc)'(Kc)'qc+JcKcqc$$

$$(6)\quad rc=JcKc(qc)'+(Jc)'Kcqc+Jc(Kc)'qc+JcKcqc$$

$$(7)\quad s=(Jd)'Kd(rc)'+Jd(Kd)'(rc)'+(Jd)'(Kd)'rc+JdKdrc$$

$$(8)\quad t=JdKd(rc)'+(Jd)'Kdrc+Jd(Kd)'rc+JdKdrc$$

Listed below, in "truth table" form, are input bit representations of the sum of the two digits at present stored in the "J" and "K" digit-registers and the carry derived from the previous addition. Also listed are the desired binary coded decimal output plus any output carry, where "AN*a*" through "AN*d*" represent the binary coded output of the sum, and "EAS" represents the decimal carry digit:

Decimal Numbers	Binary Code of Decimal Numbers (0-19) Decoder Inputs					Binary Code of Decimal Numbers (0-19) with Decimal Carry Decoder Outputs				
	t	s	r	q	ANa	ANd	ANc	ANb	ANa	EAS
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	0
2	0	0	0	1	0	0	0	1	0	0
3	0	0	0	1	1	0	0	1	1	0
4	0	0	1	0	0	0	1	0	0	0
5	0	0	1	0	1	0	1	0	1	0
6	0	0	1	1	0	0	1	1	0	0
7	0	0	1	1	1	0	1	1	1	0
8	0	1	0	0	0	0	0	0	0	0
9	0	1	0	0	1	1	0	0	1	0
10	0	1	0	1	0	0	0	0	0	1
11	0	1	0	1	1	0	0	0	1	1
12	0	1	1	0	0	0	0	1	0	1
13	0	1	1	0	1	0	0	1	1	1
14	0	1	1	1	0	0	1	0	0	1
15	0	1	1	1	1	0	1	0	1	1
16	1	0	0	0	0	0	1	1	0	1
17	1	0	0	0	1	0	1	1	1	1
18	1	0	0	1	0	1	0	0	0	1
19	1	0	0	1	1	1	0	0	1	1

Reducing the above truth table to the "canonical" form of the logical equations thereof, and thereafter simplifying, it is seen that:

$$(9)\quad ANa=(Ja)'Ka(PCI)'+Ja(Ka)'(PCI)'+(Ja)'(Ka)'PCI+JaKaPCI$$

$$(10)\quad ANb=q(EAS)'+q'EAS$$

$$(11)\quad ANc=r(EAS)'+qrs+tq$$

$$(12)\quad ANd=s(EAS)'+tq$$

$$(13)\quad EAS=sq+sr+t$$

As previously mentioned, the subtracter portion of the added-subtracter is utilized to perform a subtraction arithmetic operation of two binary-coded decimal data digits and a carry digit, and obtain therefrom a binary remainder digit plus a carry digit. As long as the minuend data digit is larger than or equal to the subtrahend data digit plus the previous carry digit, the value of the remainder digit is between binary 0000 and binary 1001 with no carry digit output. However, should the minuend be less than the subtrahend plus the previous carry, the value of the remainder digit will be between binary 0111 and binary 1111 plus an output carry digit. Therefore, in order to obtain the decimal remainder digit, the binary remainder is subtracted from binary 10000, commonly termed "taking the sixteens complement and ignoring the carries beyond the four low-order bits."

Using the laws of binary subtraction, whereby binary "0" subtracted from binary "0" results in a remainder of binary "0", binary "0" subtracted from binary "1" results in a remainder of binary "1," binary "1" subtracted from binary "1" results in a remainder of binary "0," and binary "1" subtracted from binary "0" results in a remainder of binary "1" plus binary "1" carry, suppose that the decimal digit "5" (binary 0101) is subtracted from decimal digit "9" (binary 1001). In this instance, the remainder is binary 0100 or "4" with no carry.

However, decimal digit "8" (binary 1000) subtracted from decimal digit "3" (0011) results in a binary remainder of 1011. Thus, to obtain the decimal remainder digit, the binary remainder of 1011 is subtracted from binary 10000, resulting in a new binary remainder of 0101, representative of the decimal digit "3," and a carry of "1," denoting a negative remainder.

Listed below, in truth table form, are representations of input bits "a" through "d" for each of the digits stored in the "J" and "K" digit-registers, together with the carry bit and the desired output bit for each combination thereof. As before, "Ja" and "Ka" are representative of the "a" bits of the digit stored in the "J" and "K' digit-registers, respectively; "Jb" and "Kb" are representations of the "b" bits; "Jc" and "Kc" are representative of the "c" bits; and "Jd" and Kd" are representative of the "d" bits of the digits respectively stored in the "J" and "K" digit-registers. However, in this instance, "ANa" is representative of the remainder obtained from a subtraction of the "a" bits; "PCI" is representative of the previous bit-carry; "pc" is representative of the bit-carry derived from subtraction of the "a" bits; "q" is representative of the remainder derived from subtraction of the "b" bits; "qc" is representative of the bit-carry derived from the "b" bit subtraction; "r" is representative of the remainder derived from "c" bit subtraction; "rc" is representative of the bit-carry derived from "c" bit subtraction; "s" is representative of the remainder derived from "d" bit subtraction; and "t" is representative of the bit-carry derived from "d" bit subtraction.

(a) Bit Subtraction					(b) Bit Subtraction					(c) Bit Subtraction					(d) Bit Subtraction				
Inputs			Outputs		Inputs			Outputs		Inputs			Outputs		Inputs			Outputs	
Ja	Ka	PCI	ANa	pc	Jb	Kb	pc	q	qc	Jc	Kc	qc	r	rc	Jd	Kd	rc	s	t
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1
1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0
1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0
0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1
0	1	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	1
1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Reducing the above truth tables to the canonical form of the logical equations thereof:

$$(14)\quad ANa=(Ja)'Ka(PCI)'+Ja(Ka)'(Pci)'+(Ja)'(Ka)'PCI+JaKaPCI$$

$$(15)\quad pc=(Ja)'Ka(PCI)+(Ja)'(Ka)'PCI+(Ja)'KaPCI+JaKaPCI$$

$$(16)\quad q=(Jb)'Kb(pc)'+Jb(Kb)'(pc)'(Jb)'(Kb)'pc+JbKbpc$$

$$(17)\quad qc=(Jb)'Kb(pc)'+(Jb)'(Kb)'pc+(Jb)'Kbpc+JbKbpc$$

$$(18)\quad r=(Jc)'Kc(qc)'+Jc(Kc)'(qc)'+(Jc)'(Kc)'qc+JcKcqc$$

$$(19)\quad rc=(Jc)'Kc(qc)'+(Jc)'(Kc)'qc+(Jc)'Kcqc+JcKcqc$$

$$(20)\quad s=(Jd)'Kd(rc)'+Jd(Kd)'(rc)'+(Jd)'(Kd)'rc+JdKdrc$$

$$(21)\quad t=(Jd)'Kd(rc)'+(Jd)'(Kd)'rc+(Jd)'Kdrc+JdKdrc$$

Listed below, in truth table form, are input bit representations of the remainder plus carry which are obtained from subtracting the digit stored in the "K" digit-register from the digit stored in the "J" digit-register. Also listed are the desired binary coded decimal output plus any output carry, where "ANa" through "ANd" collectively represent, in this instance, the binary coded output of the remainder, and "EAS" represents the decimal carry digit obtained from the subtraction operation.

Decimal Digits	Binary Code of Decimal Numbers "9" through "−10" Inputs					Binary Code of Decimal Numbers "9" through "−10" with Decimal Carry Outputs				
	t	s	r	q	ANa	ANd	ANc	ANb	ANa	EAS
9	0	1	0	0	1	1	0	0	1	0
8	0	1	0	0	1	1	0	0	0	0
7	0	0	1	1	1	0	1	1	1	0
6	0	0	1	1	0	0	1	1	0	0
5	0	0	1	0	1	0	1	0	1	0
4	0	0	1	0	0	0	1	0	0	0
3	0	0	0	1	1	0	0	1	1	0
2	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0
−1	1	1	1	1	1	1	0	0	1	1
−2	1	1	1	1	0	1	0	0	0	1
−3	1	1	1	0	1	0	1	1	1	1
−4	1	1	1	0	0	0	1	1	0	1
−5	1	1	0	1	1	0	1	0	1	1
−6	1	1	0	1	0	0	1	0	0	1
−7	1	1	0	0	1	0	0	1	1	1
−8	1	1	0	0	0	0	0	1	0	1
−9	1	0	1	1	1	0	0	0	1	1
−10	1	0	1	1	0	0	0	0	0	1

Reducing the above truth table to the canonical form of the logical equations thereof and then simplifying to remove redundancies therein, the four bit line and decimal carry representations of the decimal difference are:

$$(22)\quad ANa=(Ja)'Ka(PCI)'+Ja(Ka)'(PCI)'+(Ja)'(Ka)'PCI+JaKaPCI$$

$$(23)\quad ANb=qt'+q't \text{ or } q(EAS)'+q'EAS$$

$$(24)\quad ANc=rt'+t(rq'+r'q)'$$

$$(25)\quad ANd=st'+qrs$$

$$(26)\quad EAS=t$$

Having essentially derived the logical equations for an adder and a subtracter separately, the next step is to logically combine the two sets of equations thereof and obtain therefrom a circuit sharing adder-subtracter unit that is capable of performing an addition mathematical operation when the state of line (SAS)′ is TRUE, and, additionally, is capable of performing a subtraction mathematical operation when line SAS is TRUE.

By comparing the previously-derived non-simplified forms of the logical equations for lines "ANa," "q," "r," and "s" for an addition operation (Equations 1, 3, 5, and 7) with (Equations 14, 16, 18, and 20) for a subtraction operation, it is observed that the logical equation for each line is the same in each instance. Thus, whether the adder-subtracter unit is to carry out an addition or a subtraction operation, the sum and the difference digits in the binary answer are also the same in each instance. By comparing the logical equation for line "pc" for an addition and a subtraction operation (Equations 2 and 15), it is observed that two terms "(Ja)′ Ka PCI" and "Ja Ka PCI" in each equation are identical in each operation and the remaining terms thereof are different, the two like terms obviously being reducible to the simplified term "Ka PCI." Hence, the logical equation for output line "pc" for either an addition or a subtraction mathematical operation is:

$$(27)\quad pc=KaPCI+(Ja)'Ka(PCI)'SAS+(Ja)'(Ka)'PCISAS+JaKa(PCI)'(SAS)'+Ja(Ka)'PCI(SAS)'$$

The logical equations for output lines "qc," "rc," "s," and "t," likewise modified in the same manner, are as below, where:

$$(28)\quad qc=Kbpc+(Jb)'Kb(pc)'SAS+(Jb)'(Kb)'pcSAS+JbKb(pc)'(SAS)'+Jb(Kb)'pc(SAS)'$$

$$(29)\quad rc=Kcqc+(Jc)'Kc(qc)'SAS+(Jc)'(Kc)'qcSAS+JcKc(qc)'(SAS)'+Jc(Kc)'qc(SAS)'$$

$$(30)\quad t=Kdrc+(Jd)'Kd(rc)'SAS+(Jd)'(Kd)'rcSAS+JdKd(rc)'(SAS)'$$

By likewise combining the logical equations for "ANa" through "ANd" and "EAS" for both an addition operation (Equations 9 through 13) and a subtraction mathematical operation (Equations 22 through 26), it is seen that the combined logical equation for each output line is:

$$(31)\quad EAS=sq+sr+t$$

$$(32)\quad ANa=(Ja)'Ka(PCI)'+Ja(Ka)'(PCI)'+(Ja)'(Ka)'PCI+JaKaPCI$$

$$(33)\quad ANb=q(EAS)'+q'EAS$$

$$(34)\quad ANc=r(EAS)'+trq'SAS+tr'qSAS+qrs(SAS)'+tq'(SAS)$$

$$(35)\quad ANd=s(EAS)'+qrsSAS+tq(SAS)'$$

Therefore, with reference to the logical diagrams of the adder-subtracter unit as illustrated in FIGS. 53 and 54, it is seen that the above-stated logical equations numbered 27 through 35 are satisfied thereby and thus correspond to the circuit configuration thereof.

As an illustrative example of an addition and a subtraction operation being carried out by the just-described adder-subtracter arithmetic unit, suppose that the first-order digit of the addend word located in memory address $\phi\phi$ is a "nine," and the first-order digit of the augend word in address $\phi 1$ is a "seven."

To begin an "add" instruction, as previously mentioned, the augend word is first read out of memory address $\phi 1$ and stored in address-A. Thereafter, the low-order digit of the augend is sequentially read out of address-A, bit by bit, starting with the low-order binary bit "a" thereof, and is sequentially stored, bit by bit, in the "J" digit-register (FIG. 71) via lines ASA and JL. Simultaneously therewith, the addend is sequentially read out of address $\phi\phi$, bit by bit, starting with the low-order binary bit "a" thereof, and is sequentially stored, bit by bit, in the "K" digit-register (FIG. 70) via lines MSA and KL. As the assumed value of the decimal digit stored in the "J" digit-register (FIG. 71) is a "seven" (binary 0111), the state of output line Jd through Jc is FALSE, whereas the states of output lines Ja through Jc are each TRUE, the prime output lines of flipflops 6067 through 6070, of course, being of opposite states from the corresponding reference output lines thereof, as previously described. As the assumed value of the decimal digit stored in the "K" digit-register (FIG. 70) is a "nine" (binary 1001), the states of output lines Kd and Ka thereof are TRUE, whereas the states of output lines Kb and Kc are FALSE.

Reference is now made to that portion of the adder-subtracter unit logically shown in FIG. 54, in addition to periodic reference to the simplified block diagram of the adder-subtracter shown in FIG. 45C. As the low-order bit of the digit stored in each of the "J" and "K" digit-registers is a binary "1," output lines Ja and Ka are both TRUE, as just mentioned. Also, as the previous decimal carry is a "0" in this instance, the state of line (PCI)′ is TRUE, and, as an "add" operation is to be carried out, the state of line (SAS)′ is previously rendered TRUE. Therefore, as none of AND gates 956 through 959 have all of the inputs thereto simultaneously TRUE, the state of output line ANa is FALSE, indicative of binary "0." Consequently, the sum of the "a" bits of the first-order digits in the "J" and "K" digit-registers is "0."

With reference to the "a" bit carry circuitry shown in FIG. 53, as the states of lines Ja, Ka, (PCI)′, and (SAS)′ and TRUE, as previously stated, the states of all of the inputs to AND gate 901 are simultaneously TRUE. Consequently, the state of output line pc is TRUE, indicative of a binary "1" carry resulting from addition of the "a" bits.

As the "b" bit in the "J" digit-register is a binary "1" and the "b" bit in the "K" digit-register is a binary "0," the state of line Jb is TRUE and the state of line Kb is FALSE, as previously noted. As none of AND gates 911 through 914 have all of the inputs thereto simultaneously TRUE, the state of output line q is FALSE, indicative of binary "0"; consequently, the sum of the "b" bits plus the "a" bit carry is "0." However, as the states of all of lines Jb, (Kb)′, pc, and (SAS)′ are TRUE, the states of the inputs to AND gate 917 are simultaneously

TRUE. As a result, the state of output line qc is TRUE, indicative of a binary "1" carry resulting from addition of the "b" bits plus the "a" bit carry.

The "c" bit in the "J" digit-register being a binary "1" and the "c" bit in the "K" digit-register being a binary "0," the state of line Jc is TRUE and the state of line Kc is FALSE. As none of AND gates **920** through **923** have all of the inputs thereto simultaneously TRUE, the state of output line r is FALSE; consequently, the sum of the "c" bits plus the "b" bit carry is "0."

As the respective states of lines Jc, (Kc)′ qc, and (SAS)′ are each TRUE, all of the inputs to AND gate **926** are simultaneously TRUE. Consequently, the state of output line rc is TRUE, indicative of a binary "1" carry resulting from addition of the "c" bits and the "b" bit carry.

The high-order or "d" bit in the "j" digit-register being a binary "0" and the "d" bit in the "K" digit-register being a binary "1," the state of line Jd is FALSE and the state of line Kd is TRUE. With reference back to FIG. 54, as none of AND gates **934** through **937** have all of the inputs thereto simultaneously TRUE, the state of output line s is FALSE, thus indicating that the sum of the "d" bits plus the "c" bit carry is "0."

Finally, as the states of both of lines Kd and rc are TRUE, all of the inputs to AND gate **929** are simultaneously TRUE. As a result, the state of output line t is TRUE, thus indicating a binary "1" carry resulting from addition of the "d" bits plus the "c" bit carry.

Therefore, as the state of output line t is TRUE and the state of each of output lines s, r, q, and ANa is FALSE, the sum of the digit "7" (binary 0111) and the digit "9" (binary 1001) is therefore indicated as being "16," or binary 10000. The binary sum—i.e., 10000—is converted to a binary-coded-decimal form in the following manner: As the state of the single input to AND gate **948** (FIG. 54) is TRUE, the state of output line EAS is TRUE, thus indicative of a decimal "1" carry. Therefore, as the state of each of lines EAS and $(q)'$ is TRUE, the state of output line ANb is likewise TRUE, thus indicating that the value of the "b" bit of the sum digit is a binary "1." As the state of each of lines $(q)'$, t, and (SAS)′ is TRUE, all the inputs to AND gate **942** are TRUE, thus rendering output line ANc TRUE and thereby indicating that the value of the "c" bit of the sum digit is a binary "1." As none of AND gates **945** through **947** have all of the inputs thereto simultaneously TRUE, the state of line ANd is FALSE and thereby indicates that the high-order bit "d" of the sum digit is a binary "0."

It is now evident that, upon completion of the just-described addition operation of the two decimal digits "7" and "9," the state of output line ANa is FALSE, the state of output line ANb is TRUE, the state of output line ANc is TRUE, and the state of output line ANd is FALSE. Thus, the states of output lines ANa through ANd collectively represent the sum digit "6" (binary 0110). As the state of output line EAS is simultaneously TRUE, as before stated, its state is indicative of the decimal "1" carry digit.

Following the just-described addition operation of the two first-order digits taken from the data words stored in address $\phi\phi$ and address A, there is derived thereby a sum digit which is written in the low-order digit position of address-A and a decimal "1" carry digit which is stored in carry flipflop **6000** (FIG. 53) via line EAS. Thereafter, the second-order digit of the word in address $\phi\phi$ is transferred to the "K" digit-register, and the second-order digit of the word in address-A is transferred to the "J" digit-register. As the output lines from the "J" and "K" digit-registers are connected as input lines to the adder-subtractor unit, as previously described, the two second-order digits are simultaneously "presented" to the adder-subtracter unit, wherein they are added together, along with the decimal "1" carry in flipflop **6000**, in exactly the same manner as just described with respect to the first-order digits of the two data words. The above sequence of operations is sequentially repeated for each higher-order digit until the tenth-order decimal digits have been added and the sum digit stored in the tenth-order digital position of address-A. As previously mentioned, the final step of an "add" instruction is to effect the copying of the sum data word from address-A into the particular memory address as previously specified.

A "subtract" instruction is carried out in essentially the same manner as the just-described "addition" operation, with the exception that prior thereto, the state of line (SAS)′, originating in FIG. 53, is selectively rendered TRUE, the state of line SAS, of course, being simultaneously rendered FALSE. Consequently, in the previous example, the adder-subtracter unit now effectively subtracts the word in address $\phi\phi$ from the word in address-A, after which the difference word is transferred from address-A to a particular address in memory as specified.

As the mode of operation of the adder-subtracter unit in carrying out an addition or a subtraction mathematical operation is essentially the same in both instances, in light of the just-described detailed mode of operation with respect to an addition operation, a further detailed mode of operation thereof with respect to a subtraction operation is not deemed necessary in order to insure a full and complete understanding and appreciation of the adder-subtracter portion of the instant computer.

63. *Control Counter Generally*

With reference to the upper left-hand portion of FIG. 45A, there is diagrammatically illustrated a "control counter" comprising a multitude of flipflops and a maze of other electronic control circuitry. As will become more obvious hereinafter, the control counter portion of the computer functions in many ways as a central regulatory means which electronically controls the proper sequence of all data handling and transfer within the computer, and, additionally, controls all communication between the control portion of the computer and the various input-output equipment utilized by the computer. Consequently, due to the inherent complex functional nature of the control counter, it is not readily susceptible of being described as a separate entity as such.

As an example of the foregoing, the first step to be carried out in the execution of each instruction is to read out from memory the next instruction word in the program and store the instruction word in the instruction register. After the instruction word is stored in the instruction register, the two decimal digits stored in section 1 thereof are "examined" to determine the particular type of instruction, or sequence of operations, to be carried out next. For example, as previously described, if the two-digit decimal number in section 1 is an "eleven," an "add-pairs-of-numbers" instruction is carried out, and all data thereafter transferred between data-handling sections of the computer is in accordance with a particular pattern dictated by this instruction.

Each instruction concludes by effecting the storage in the word-selecting register, the address in memory of the next instruction word; the memory address usually is specified in section 5 of the present instruction word, also as previously described. After the address of the next instruction word is stored in the word-selecting register, the computer may immediately read out the next instruction word from memory, or may pause for a predetermined period of time before reading out the instruction. Whether or not the computer immediately proceeds to carry out the next instruction in the program depends upon the particular mode of operation preselected by the operator. If the computer is operating in an "automatic" mode, the next instruction in the program is immediately carried out upon completion of each preceding instruction. However, if the computer is operating in a "manual" mode, all computation steps upon completion of each instruction and is not resumed until either the manual or the automatic-resume-program-push-button is depressed. Thereafter, the next instruction word, as specified by the word-selection register, is stored in the instruction register, the two decimal digits in section 1 of the instruction register are examined to determine the particular instruction to be carried out, and the computer then proceeds to carry out that particular instruction.

64. *Subinstructions Generally*

There have been previously mentioned, and briefly described herein, eighteen different types of general-purpose instructions to which the computer is responsive, and the specific mode of operation or sequence of data handling and transfer events which is initiated by the computer in carrying out each of the instructions given. For the purpose of convenience in describing in detail the inherently complex mode of operation which the computer executes in carrying out a particular instruction, each of the eighteen different instructions will hereinafter be considered a "subprogram" comprising a plurality of "subinstructions" which are sequentially given to the computer, the sequence of events initiated by the computer in carrying out each of the subinstructions being under the control of the "control counter," as previously mentioned.

Each of the just-mentioned subinstructions, hereinafter described, falls within one of six general categories, labeled "word-cycles," "subcommands," "variable-time-delays," "decisions," "incrementing and decrementing," and "miscellaneous," where a "word-cycle" type of subinstruction effects the transfer of a data or instruction word from one memory address to another, or from one memory address to the instruction register; where a "subcommand" type of subinstruction effects the modification and/or copying of an instruction or data word from one memory address to another; where a "variable-time-delay" type of subinstruction essentially indicates the degree of completion of a subprogram, somewhat like a "bookmark," while the computer, in a sense, is "waiting" for some outside action or event to take place (for example, a variable-time-delay subinstruction is effective to stop the cycle of operation called for by a subsequent subinstruction while the operator indexes an amount into the keyboard during an "enter-keyboard-words" instruction (EKW—**00**)); where "decision" type of subinstructions allows the cycle of operations called for by subsequent subinstructions to be skipped, depending upon certain prescribed conditions being met, instructions involving (CFM—**15**) and (CFE—**16**), for example; where "incrementing and decrementing" types of subinstructions effectively initiate an incrementing or decrementing cycle of operation with respect to the various counters within the computer; and where the subinstructions labeled "miscellaneous" are those types which do not come within a common classification.

65. *Functional Listing and Description of Subinstructions*

In the following chart is a functional listing and description of a substantial portion of the various subinstructions, with each being preceded by a code designation which identifies that particular subinstruction. It is to be noted that each subinstruction, in a sense, is related to a specific portion of the computer control circuitry even though each such specific portion does not always exist as a separate and distinct entity as such. Therefore, in an attempt to further simplify the following description, the code designation preceding the leftmost hyphen corresponds to a like-designated signal line, the state of energization of which is the prime instrumentality, either in initiating the required cycle of operation as called for by that particular subinstruction after certain prescribed conditions have been met, or, in some instances, as an indication that a particular cycle of operation has been completed. The fact that more than one code designation refers to a common signal line indicates that a corresponding number of different cycles of operation are selectively initiated or indicated by that particular signal line; i.e., the line has more than one function, as indicated by the numeral in parentheses. The location in the drawings of the origination of each of such like-designated signal lines is given by the figure number of the drawing in the column of the chart below labeled "FIG. NO." The particular type of each subinstruction is given by the notation in the column labeled "TYPE," wherein type "WC" designates a "word-cycle" type of subinstruction; "SC" designates a "subcommand" type subinstruction; "VTD" designates a "variable-time-delay" subinstruction; "D" designates a "decision" type subinstruction; "I–D" designates an "incrementing or decrementing" type subinstruction; and "M" designates a "miscellaneous" type subinstruction. The asterisk symbol designates the next cycle of operation in the subprogram which is to be carried out upon completion of the cycle of operation called for by that particular subinstruction.

SUBINSTRUCTIONS

Code	Type	Fig. No.	Description
AD(1)-O-*	I-D	61	Increment digit-counter.
AD(2)-O-*	I-D	61	Decrement digit-counter.
AF-O-*	I-D	72	Increment F-counter.
AI-X-Y	M	77	If the digit just read from the paper tape is an alternate-instruction symbol, go to Stop-Y; otherwise, go to Stop-X.
AM-n-*	WC	74	Copy word of address-A into memory at address specified by section-n of instruction register.
ARO-O-*	SC	74	Add word of address-A to a constant and store result in address-A.
BB-O-*	M	85	Record single linefind type of infomration on ledger card.
Bd-X-Y	D	62	If bit-counter is at bit "d," go to Step-Y; otherwise, go to Step-X.
BM-n-*	WC	74	Copy word of address-B into memory at address specified by section-n of instruction register.
CA(1)-O-*	WC	74	Clear address-A; i.e., write all "zeros" therein.
CA(2)-O-*	WC	74	Copy word indicated by accounting machine readout switches into address-A.
CB-O-*	WC	74	Clear address-B.
CBS-O-*	M	81	Energize "credit-balance" solenoid.
CCR-X-Y	D	85	Wait until clock pulse is magentically picked up from ledger card and, thereafter, go to Step-Y. When ledger card is all the way into carriage prior to reversal, go to Step-X if a clock pulse has not been picked up.
CLC-O-*	M	82	Close accounting machine carriage.
CM-n-*	WC	74	Clear memory address specified by section-n of instruction register.
CPA-O-*	SC	74	Complement word of address-A and store result in address-A.
CPM-n-*	SC	74	Complement word of memory address specified by section-n of instruction register and store result in same address.
CYC-O-*	I-D	61	Increment bit-counter.
Dϕ-X-Y	D	63	If digit-counter is at "0," go to Step-Y; otherwise, go to Step-X.
D1-X-Y	D	63	If digit-counter is at "1," go to Step-Y: otherwise, go to Step-X.
D2-X-Y	D	63	If digit-counter is at "2," to to Step-Y; otherwise, go to Step-X.
D3-X-Y	D	63	If digit-counter is at "3," go to Step-Y; otherwise, go to Step-X.
D4-X-Y	D	63	If digit counter is at "4," go to Step-Y; otherwise, go to Step-X.
D5-X-Y	D	63	If digit-counter is at "5," go to Step-Y; otherwise, go to Step-X.
D6-X-Y	D	63	If digit-counter is at "6," go to Step-Y; otherwise, go to Step-X.
D7-X-Y	D	63	If digit-counter is at "7," go to Step-Y; otherwise, go to Step-X.
D8-X-Y	D	63	If digit-counter is at "8," to to Step-Y; otherwise, go to Step-X.
D9-X-Y	D	63	If digit-counter is at "9," go to Step-Y; otherwise, go to Step-X.
DAD-O-*	SC	75	Add a constant to word in address-B and store sum in address-B.

SUBINSTRUCTIONS—Continued

Code	Type	Fig. No.	Description
DR-O-*	I-D	68	Decrement R-counter.
DE2-O-*	I-D	58	Decrement section-4 of instruction register.
EAS-X-Y	D	54	If the algebraic sign of the difference is positive, go to Step-Y; otherwise, go to Step-X.
EOF-X-Y	D	77	If the digit just read from paper tape is an end-of-frame symbol, go to Step-Y; otherwise, go to Step-X.
EOW-X-Y	D	71	If "J" digit-register is at "15", go to Step-Y; otherwise, go to Step-X.
F-9-X-Y	D	72	If F-counter is at "9," go to Step-Y; otherwise, go to Step-X.
F1ϕ-X-Y	D	72	If F-counter is at "10," go to Step-Y; otherwise, go to Step-X.
GC-O-*	VTD	75	Variable time-delay.
GO-O-*	M	82	Go to next step in subprogram whenever depression of a motor bar will initiate a cycle of operation of the accounting machine.
Hϕ-X-Y	D	57	If high-order digit of section-2 of instruction register is "0," go to Step-Y; otherwise ,go to Step-X.
H1-X-Y	D	57	If high-order digit of section-2 of instruction register is "1," go to Step-Y; otherwise, go to Step-X.
H2-X-Y	D	57	If high-order digit of section-2 of instruction register is "2," go to Step-Y; otherwise ,to to Step-X.
H3-X-Y	D	57	If high-order digit of section-2 of instruction register is "3,", go to Step-Y; otherwise, go to Step-X.
H4-X-Y	D	57	If high-order digit of section-2 of instruction register is "4," go to Step-Y; otherwise, go to Step-X.
H5-X-Y	D	57	If high-order digit of section-2 of instruction register, is "5" go to Step-Y; otherwise, go to Step-X.
H6-X-Y	D	57	If high-order digit of section-2 of instruction register is "6," go to Step-Y; otherwise, go to Step-X.
I34-X-Y	D	59	If memory addresses specified by sections 3 and 4 of instruction register are equal, go to Step-Y; otherwise, go to Step-X.
IFD-X-Y	D	72	If digits in F-counter and digit-counter are equal, go to Step-Y; otherwise, go to Step-X.
IFJ-X-Y	D	72	If digits in F-counter and "J" digit-register are equal, go to Step-Y; otherwise, go to Step-X.
IN4-O-*	I-D	58	Increment section-3 of instruction register.
IW-O-*	M	61	Transfer contents of particular section of instruction register into word-selecting register as indicated by the state of a corresponding one of lines TI2 through TI5.
J9-X-Y	D	71	If digit in "J" digit-register is a "9," go to Step-Y; otherwise, go to Step-X.
JHR-O-*	VTD	75	Variable time-delay.
JL(1)-O-*	M	71	Store in "J" digit-register the data bit read from ledger card, which bit corresponds to count in bit-counter.
JL(2)-O-*	M	71	Store in "J" digit-register the data bit read from memory, which bit corresponds to count in bit-counter.
JM-n-*	M	60	Copy digit of "J" digit-register into memory address at digital position indicated by digit-counter, the address in memory being specified by section-m of instruction register.
Kϕ-X-Y	D	70	If digit in "K" digit-register is a "0," go to Step-Y; otherwise, go to Step-X.
KEY-O-*	VTD	75	Variable time-delay.
L-O-*	M	85	Place ledger card on next posting line.
LϕX-Y	D	57	If low-order digit in section-2 of instruction register is a "0," go to Step-Y; otherwise, go to Step-X.
L1-X-Y	D	57	If low-order digit in section-2 of instruction register is a "1," go to Step-Y; otherwise, go to Step-X.
L2-X-Y	D	57	If low-order digit in section-2 of instruction register is a "2," go to Step-Y; otherwise, go to Step-X.
L3-X-Y	D	57	If low-order digit in section-2 of instruction register is a "3," go to Step-Y; otherwise go to Step-X.
LFA-O-*	M	85	Wait for next linefind impulse.
LK-O-*	VTD	75	Variable time-delay.
LL-O-*	M	85	Record multiple linefind type information on ledger card.
MA-n-*	WC	75	Copy word from memory into address-A, and, simultaneously, copy the word digit by digit into "J" digit-register, the address of the word in memory being specified by section-n of the instruction register.
MBϕ(1)-O-*	M	81	Depress upper motor bar for touch operation.
MBϕ(2)-O-*	M	81	Depress upper motor bar for hold operation.
MB1(1)-O-*	M	81	Depress middle motor bar for touch operation.
MB1(2)-O-*	M	81	Depress middle motor bar for hold operation.
MB2(1)-O-*	M	81	Depress lower motor bar for touch operation.
MB2(2)-O-*	M	81	Depress lower motor bar for hold operation.
MI-O-*	WC	76	Copy instruction word from memory into instruction register, the memory address of the instruction word being indicated by the contents of the word-selecting register.
MIC-O-*	VTD	76	Variable time-delay.
MJ-n-*	WC	71	Copy the word stored in the address in memory, as specified by section-n of the instruction register, digit by digit, into the "J" digit-register.
MK-n-*	M	69	Copy the digit of the word, as indicated by the digit-counter, from memory address into "K" digit-register, the address in memory being specified by section-n of the instruction register.
MOU(1)-O-*	VTD	76	Variable time-delay.
MOU(2)-O-*	M	76	Eject the ledger card.
OBM(1)-O-*	VTD	76	Variable time-delay.
OBM(2)-X-Y	D	76	If state of line OBM is TRUE, go to Step-Y; otherwise, go to Step-X
OBM(3)-O-*	M	76	Preset line OBM to a FALSE state.
OBM(4)-O-*	M	76	Preset state of line OBM to correspond to sign of word read from memory.
OBM(5)-O-*	M	76	Preset line OBM to a TRUE state.
OBN(1)-X-Y	D	77	If state of line OBN is TRUE, go to Step-Y; otherwise, go to Step-X.
OBN(2)-O-*	M	77	Preset line OBN to a TRUE state.
OBN(3)-O-*	M	77	Preset line OBN to a FALSE state.
OBN(4)-O-*	D	77	Set line OBN to a TRUE state if the sign of the word read from memory corresponds to the state of line OBM; otherwise, set line OBN to a FALSE state.
OH1-O-*	M	81	When printing occurs, place a comma between rows #5 and #6, and, if there is no significant digit in rows #6 through #10, print a "zero" preceding the comma.
OH2-O-*	M	81	When printing occurs, place a comma between rows #8 and #9. If there is no significant digit in rows #9 or #10, print a "zero" preceding the comma.
PCT-O-*	M	72	When amount racks are traveling in a "setting" direction, go to next step in subprogram each time the timing rack changes digital position and also when printing liner comes in.
PFϕ-O-*	M	72	Preset F-counter to "0."
PIϕ-O-*	M	55	Preset instruction register to "zero."
PJϕ-O-*	M	71	Preset "J" digit-register to "0."
PJ1-O-*	M	71	Preset "J" digit-register to "1."
PJ5-O-*	M	71	Preset "J" digit-register to "5."
PKϕ-O-*	M	70	Preset "K" digit-register to "0."
PK9-O-*	M	70	Preset "K" digit-register to "9."
PRϕ-O-*	M	68	Preset R-counter to "0."
PR2-O-*	M	68	Preset R-counter to "2."
RP8-O-*	M	68	Preset R-counter to "8."
PTF-O-*	M	76	Translate paper tape in a forward direction.
PTR-O-*	M	76	Translate paper tape in a reverse direction.
PTS-O-*	M	76	Stop paper tape.
PWϕ-O-*	M	64	Preset word-selecting register to "00."
Rϕ-X-Y	D	68	If R-counter is "0," go to Step-Y; otherwise, go to Step-X.
R1-X-Y	D	68	If R-counter is "1," go to Step-Y; otherwise, go to Step-X.

Code	Type	Fig. No.	Description
R8-X-Y	D	68	If R-counter is "8," then go to Step-Y; otherwise, go to Step-X.
RAϕ-O-*	M	80	Energize first-order rack-stopping solenoid.
RA1-O-*	M	80	Energize second-order rack-stopping solenoid.
RA2-O-*	M	80	Energize third-order rack-stopping solenoid.
RA3-O-*	M	80	Energize fourth-order rack-stopping solenoid.
RA4-O-*	M	80	Energize fifth-order rack-stopping solenoid.
RA5-O-*	M	80	Energize sixth-order rack-stopping solenoid.
RA6-O-*	M	80	Energize seventh-order rack-stoppong solenoid.
RA7-O-*	M	80	Energize eighth-order rack-stopping solenoid.
RA8-O-*	M	80	Energize ninth-order rack-stopping solenoid.
RA9-O-*	M	80	Energize tenth-order rack-stopping solenoid.
RAD-n-*	SC	77	Add the word in the address specified by section-n of the instruction register to the word in address-A, and store sum in address-A.
Rb-O-*	M	68	Preset R-counter to "2."
REV-X-Y	D	77	If accounting machine reverse key is depressed, go to Step-Y; if not, go to Step-X.
RLM-n-*	M	68	Load into R-counter a digit from word in memory address specified by section-n of the instruction register, which digit corresponds to count in digit-counter.
RLR-O-*	M	68	Preset the count of the R-counter to correspond to low-order digit of section-2 of instruction register.
ROS-O-*	VTD	77	Variable time-delay.
RPB-O-*	VTD	78	Upon depression of resume-program-bar, go to next step in subprogram.
RRS-O-*	M	81	Energize rack-release solenoid.
RSB-n-*	SC	78	Subtract word in memory from word in address-A and store remainder in address-A, the address in memory being specified by section-n of the instruction register.
SA(1)-O-*	SC	78	Shift word of address-A one place to left and store result in address-A.
SA(2)-O-*	SC	78	Shift word of address-A one place to right and store result in address-A.
SAM-O-*	VTD	79	Variable time-delay.
SB(1)-O-*	SC	79	Shift word of address-B one place to left and store result in address-B.
SB(2)-O-*	SC	79	Shift wore of address-B one place to right and store result in address-B.
SES-O-*	VTD	79	Variable time-delay.
STD-n-*	M	79	Copy contents of section-n of instruction register into word-selecting register.
TCL-O-*	M	75	Go to next step when the output of the paper tape clock is rendered TRUE.
TDA-X-Y	D	69	If the character just read from the tape is a data digit, go to Step-Y; otherwise, go to Step-X.
TDS-n-*	M	60	Store the data just read from the tape in memory address specified by contents of section-n of instruction register.

66. *Word-Cycle Subinstructions*

Each of the previously-described word-cycles is herein classified in one of three categories, depending upon the particular mode or sequence of operations initiated thereby. The first category comprises word-cycles (AM–*n*–*), (BM–*n*–*), and (MA–*n*–*), which are hereinafter respectively designated "AM," "BM," and "MA" word-cycles. As shown in the just-preceding chart, an "AM" word-cycle initiates the copying of the word of address-A into the memory address specified by a particular section of the instruction register; a "BM" word-cycle initiates the copying of the word of address-B into the memory address specified by a particular section of the instruction register; and a "MA" word-cycle effects the copying into address-A of the word located in the memory address specified by a particular section of the instruction register, the word also being simultaneously copied digit by digit into the "J" digit-register.

The second category comprises word-cycles (CA–O–*), (CB–O–*), and (CM–*n*–*), which are hereinafter respectively designated "CA," "CB," and "CM" word-cycles. Again, as shown in the just-preceding chart, a "CA" word-cycle is capable of selectively initiating two distinct modes of operation. The first mode of operation selectively initiated thereby is the clearing of address-A by the storage of "zeros" therein, whereas the second mode of operation selectively initiated thereby is the copying into address-A of the word indicated by the collective conditions of the accounting machine rack read-out switches. A "CB" and "CM" word-cycle respectively initiates the clearing of address-B and the memory address specified by a particular section of the instruction register.

The third category comprises word-cycles (MI–*n*–*) and (MJ–*n*–*), which are herein respectively designated "MI" and "MJ" word-cycles, where a "MI" word-cycle initiates the copying into the instruction register of the instruction word previously stored in a specified address in memory, and where a "MJ" word-cycle initiates the digit-by-digit copying into the "J" digit-register of the word stored in a particular address in memory.

Each word-cycle comprises forty read-write bit cycles of forty microseconds each, plus one bit time of forty microseconds preceding the first read-write bit cycle, and plus one bit time of forty microseconds following the completion of the last read-write bit cycle, for control purposes. Thus, the total time required for completion of a word-cycle operation is 1.68 milliseconds.

During the first bit time period preceding the first read-write bit cycle, transfer to the word-selecting register of the selected memory address previously stored in the instruction register is effected by selectively energizing one of lines TI2 through TI5 (FIGS. 64 and 64) in the manner previously described with respect to the mode of operation of instruction register. Also during the first bit time preceding the first read-write bit cycle, the states of the particular signal lines that initiate the selection of a particular one, or ones, of memory addresses ϕϕ through 99 and addresses "A" and "B" are selectively rendered TRUE. For example, as shown in FIG. 60, in order to initiate any one of word-cycles "AM," "BM," "CM," "MA," "MJ," or "MI," the state of a correspondingly-labeled input line to logical OR **3116** is rendered TRUE and thereby renders the state of output line MYW TRUE also. As previously described, the state of output line MYW is rendered TRUE for a period of 1680 microseconds to permit one of Y-driver output lines YDϕ through YD9 (FIG. 65) to be selectively energized for twenty-microsecond periods by selected ones of output lines Wϕϕ through Wϕ9 from the decoding portion of the low-order section of the word-selecting register. Likewise, in order to initiate any one of word-cycles "AM," "CA," or "MA," the state of a correspondingly-labeled input line to logical OR **3126** (FIG. 60) is rendered TRUE and thereby renders the state of line AYW TRUE also. The state of line AYW, in this instance, is also rendered TRUE for a period of 1680 microseconds to permit Y-driver output line AAD, for address-A, to be selectively energized (see also FIG. 52B); and, in order to initiate one of word-cycles "BM" or "CB," the state of a correspondingly-labeled input line to logical OR **3127** is rendered TRUE and thereby renders the state of line BYW TRUE also. The state of line BYW, in this instance, is also TRUE for a period of 1680 microseconds to permit Y-driver output line BBD for address-B to be selectively energized.

Also during the first bit time period preceding the first read-write bit cycle, in order to initiate an "AM" or "BM" word-cycle, logical AND **1088** is conditioned by the state of a correspondingly-labeled one of lines AM and BM being rendered TRUE, so that, when the state of line ASA from address-A and B sense amplifier flipflop **6056** (FIG. 67A) is rendered TRUE, the state of output line MXW also is rendered TRUE. In order to initiate a "MA," "MJ," or "MI" word-cycle, logical AND **3120** is conditioned by the state of a correspondingly-labeled one of lines MA, MJ, and MI being selectively rendered TRUE, so that, when the state of line MSA from the memory sense amplifier flipflop **6055** (FIG. 67A) is rendered TRUE, the state of output line MXW is thereby rendered TRUE. As previously described, when the state of line MXW is TRUE simultaneously with lines SMC and C**41**, all the inputs to logical AND **1098** are thus TRUE, and, as a result, the state of line XDW is thereby rendered TRUE, the states of line XDW and output lines B*a*M through B*d*M from the bit-counter being capable of effecting selective energization of X-driver output lines XD*a* through XD*d*.

The chart below illustrates the respective state which each of lines MYW, AYW, BYW, and MXW is selectively and conditionally rendered in response to the initiation of each of the eight word-cycles just described:

Word-Cycles	MYW	AYW	BYW	MXW
"AM"	True	True	False	True.
"BM"	True	False	True	True.
"CA" (1)	False	True	False	False.
"CA" (2)	False	True	False	True.
"CB"	False	False	True	False.
"CM"	True	False	False	False.
"MA"	True	True	False	True.
"MI"	True	False	False	True.
"MJ"	True	False	False	True.

As just mentioned, in order to effect the carrying out of each of the previously-described word-cycles, with the exception of line MXW, the state of a selected one, or ones, of lines MYW, AYW, and BYW is selectively rendered TRUE during the entire word-cycle initiated thereby; i.e., 1680 microseconds. With reference to FIGS. 64 and 65, with the exception of a "MI" word-cycle, the state of line $(PW\phi)'$ is TRUE at the beginning of each word-cycle, so that, at the first TIME–**2**, the state of line $(PW\phi)'$ is rendered FALSE and thereby causes the word-selecting register to be preset to "00" at TIME–**2**. With reference to FIG. 61, the state of line IW is rendered TRUE at TIME–**3** and remains TRUE for ten microseconds until TIME–**4**, at which time the state of line IW is rendered FALSE. As shown, line IW is logically ANDED with each of lines TS2 through TS5, so that the contents of a preselected one of sections 2 through 5 of the instruction register is transferred to and thus stored in the word-selecting register at TIME–**4** when the state of line IW is rendered FALSE, all in the manner previously described with respect to that portion of the description relating to the mode of operation of the instruction register. At the just-preceding TIME–**3**, the state of line WDF is rendered TRUE, and, consequently, the states of all the inputs to logical AND **1115** are simultaneously TRUE at TIME–**3**, so that, ten microseconds later at the just-mentioned TIME–**4**, the state of the reference input to flipflop **6039** changes from TRUE to FALSE, thus rendering the state of line CFF TRUE at TIME–**4**. Thereafter, the state of line CFF remains TRUE for a period of 1600 microseconds. As the states of both of the input lines to logical AND **1131** are simultaneously TRUE at each TIME-**4** during initiation of a word-cycle, it is evident that the state of line CYC is rendered TRUE at each TIME-**4** and is rendered FALSE at each TIME-**1**.

As previously described with respect to the bit-counter (FIG. 62), for each successive occurrence of a TRUE-to-FALSE change of state of line CYC, the bit-counter is selectively incremented or decremented by a count of one binary bit. Thus, at the second TIME–**1** at the beginning of each word-cycle, the bit-counter is effectively incremented or decremented, so that the state of output line B*a* therefrom is rendered TRUE instead of line B*d*, or vice versa. As shown in FIG. 60, the state of line MOO is rendered TRUE for the entire time duration of each of the word-cycles. Therefore, with reference to FIG. 61, the states of all of the input lines to logical AND **1123** are simultaneously TRUE at the first TIME–**4** of each word-cycle, the state of output line AD is rendered TRUE at the first TIME–**4** and, ten microseconds later, is rendered FALSE at the second TIME–**1**. As previously described with respect to the digit-counter (FIG. 63), for each successive TRUE-to-FALSE change of state of line AD, the digit-counter is selectively incremented or decremented by one digit, depending upon whether the digit-counter was previously conditioned to count in a forward or a reverse direction. Therefore, when the bit-counter effectively advances from a count of "*d*" to a count of "*a*," the state of line AD reverses from TRUE to FALSE, and, consequently, at the second TIME–**1** when the state of line B*d* is rendered FALSE, the digit-counter is effectively advanced from a count of "9" to a count of "0," as indicated by the state of digit-counter line Dϕ being rendered TRUE at TIME–**1**.

As previously described with respect to the word-selecting register (FIGS. 64 and 65) together with that portion of the previous description entitled "Detailed Read-Write Cycle," the bit and digit counters in combination determine which row of cores is to receive a half-select current impulse during a read-write cycle. That is, the output of the bit-counter effectively determines the binary bit to be read out, whereas the output of the digit-counter effectively determines the particular digit of the word to be read out of a particular address in memory.

With reference to FIG. 60, at the first TIME–**4** of each word-cycle when the state of line CFF is rendered TRUE, the states of all of the input lines to logical AND **1094** are simultaneously TRUE. Thus, at the second TIME–**1**, the state of flipflops **6038** is reversed, so that the state of line SMC is thereby rendered TRUE and remains TRUE for 1600 microseconds during the entire read-write cycle of operation during which the ten-digit word is read out of a particular address in memory, the word being read out bit by bit, beginning with low-order bit "*a*" of the first-order digit and ending with high-order bit "*d*" of the tenth-order digit of the word, all in the manner previously described in detail.

At TIME–**4** after bit "*d*" of the tenth-order digit of the word has been read out, the state of the prime input to flipflop **6039** (FIG. 61) is reversed from TRUE to FALSE, so that the state of flipflop **6039** is reversed and the state of output line CFF is thereby rendered FALSE at the TIME–**4** following the reading-out of the last bit of the word. Due to the fact that the state of line CFF is thus rendered FALSE, the bit and digit counters are prevented from being incremented at the following TIME–**1**, but, instead, respectively remain at counts of "*d*" and "9." At the following TIME–**1**, the state of the prime input to flipflop **6038** (FIG. 60) is reversed from TRUE to FALSE, so that the state of flipflop **6038** is reversed and the state of output line SMC is thereby rendered FALSE at the TIME–**1** preceding the last forty-microsecond interval of the word-cycle, thus terminating the read-write cycle of operation.

With reference to FIG. 61, when the state of output line CFF is rendered FALSE by flipflop **6039** at TIME–**4** during the reading and writing of the last bit of the word, the state of line (CFF)′, of course, being TRUE, the states of all of the input lines to logical AND **1119** are simultaneously TRUE at that particular TIME–**4**. Thus, at the TIME–**1** preceding the last forty-microsecond interval of the word-cycle, flip-flop **6040** changes state, so that the state of line PC is thereby rendered TRUE and remains TRUE for a period of forty microseconds. Consequently, the state of output line AN is rendered

TRUE from TIME–4 to TIME–1, and the state of output line EG is rendered TRUE from TIME–2 to TIME–3 of the last forty-microsecond period of the word-cycle. At the last TIME–1, the state of line AN is rendered FALSE and, in a sense, effectively "turns off" the particular flipflop which is responsible for initiating that particular word-cycle. For example, with reference to FIG. 74, in order to initiate an "AM" word-cycle, the state of output line AM is selectively rendered TRUE by flipflop **6077**; in order to initiate a "BM" word-cycle, the state of line BM is selectively rendered TRUE by flipflop **6079**; in order to initiate a "CA" word-cycle, the state of line CA is selectively rendered TRUE by flipflop **6080**; and, so on, for word-cycles "CB" (flipflop **6081**, FIG. 74), "CM" (flipflop **6082**, FIG. 74), "MA" (flipflop **6091**, FIG. 75), "MI" (flipflop **6092**, FIG. 76), and "MJ" (flipflop **6066**, FIG. 71). As shown, line AN is logically connected to the prime input of each of the just-mentioned word-cycle flipflops in a manner such that, when the state of line AN is rendered FALSE at TIME–1 following completion of each word-cycle, the state of the flipflop responsible for initiating that particular word-cycle is reversed, thus bringing to an end the word-cycle initiated thereby.

With the exception of an "MI" word-cycle, each of the remaining word-cycles essentially conforms to the same time pattern thus described in detail. However, with respect to an "MI" word-cycle during which the state of line MI (FIG. 76) is rendered TRUE, the word-selecting register (FIGS. 64 and 65) is not preset to "00" as before, and, in addition, the state of line IW (FIG. 61) remains FALSE during the first forty-microsecond period of the word-cycle. Consequently, the individual contents of sections 2 through 5 of the instruction register are not transferred to the word-selecting register. Instead, each of the sections of the instruction register is preset to "00" prior to the transfer thereto of the word from a particular memory address. The remaining portion of an "MI" word-cycle, however, is the same as just described.

During a "MA" and a "MJ" word-cycle, the digits of the word in memory are sequentially stored in the "J" digit-register, starting with the first-order digit and ending with the tenth-order digit thereof. Thus, as shown in FIG. 71, the "J" digit-register is preset to "0" by a change of state of line PJϕ at TIME–3 during the "*a*" bit time interval of each digit during which the state of line B*a* is TRUE. Thereafter, bits "*a*" through "*d*" of the digit are sequentially stored in the "J" digit-register.

67. *Digit-Cycles*

There are occasions when it is desirable to read a word out of memory, or, conversely, to store a word in memory on a digit-by-digit basis, hereinafter called "digit-cycle," whereby the time lapse between each digit-cycle is possibly as long as three milliseconds, as, for example, during the carrying-out of "record-on-card" (ROC—**03**), "enter-card-words" (ECW—**02**), and "enter-punched-tape" (EPT—**17**) instructions.

With reference to FIG. 69, during an ROC, ECW, or EPT operation, the state of flipflop **6061** is selectively reversed, so that the state of output line TH is thereby rendered TRUE. Output line TH, when TRUE, renders the state of output line MK TRUE during an ROC operation, and renders the state of output line JM (FIG. 6Q) TRUE during an ECW or an EPT operation.

Referring to FIG. 60, when the state of output line MK is rendered TRUE during a ROC operation, the state of output line SOO is thereby rendered TRUE.

As will later be seen, the time required for completion of a digit-cycle is five binary bits of forty microseconds each; i.e., 200 μsec. Thus, at the beginning of the first forty-microsecond bit period for each digit-cycle, the state of line SOO is rendered TRUE and remains TRUE for the entire two-hundred-microsecond period until the digit-cycle operation is completed. With reference to FIG. 64, the states of all of the input lines to logical AND **1198** are simultaneously TRUE at the beginning of each digit-cycle. At the following TIME–2, output line (PWϕ)′ experiences a TRUE-to-FALSE reversal of state and, consequently, presets the word-selecting register to "00." As shown in FIG. 61, the state of line DDF is rendered TRUE by AND gate **1114** at the first TIME–3 and FALSE at the first TIME–4 of the digit-cycle. As line IW is logically ANDED with each of lines TS2 through TS5, as previously described, and the state thereof is rendered TRUE at TIME–3 and FALSE at TIME–4, the contents of a selected one of sections 2 through 5 of the instruction register is transferred to the word-selecting register at the first TIME–4 in the manner previously described. Also, at the first TIME–4, the states of all of the input lines to logical AND **1128** are simultaneously TRUE, so that the state of line AD is rendered TRUE at the first TIME–4 and FALSE at the following TIME–1. Thus, when line AD experiences a TRUE-to-FALSE reversal of state, the digit-counter is advanced at the second TIME–1 from a count of "9" to a count of "0."

At the first TIME–3 when the state of line DDF is rendered TRUE, the states of all of the input lines to logical AND **1116** are TRUE. Consequently, at the first TIME–4, the reference input to flipflop **6039** reverses from a TRUE to a FALSE state, so that the state of output line CFF is rendered TRUE at the first TIME–4 by a reversal of state of flipflop **6039**. With reference to FIG. 60, the states of all of the input lines to logical AND **1094** are simultaneously TRUE at the first TIME–4. Consequently, at the second TIME–1, the state of flipflop **6038** is reversed, so that the state of output line SMC is thereby rendered TRUE. With reference to FIG. 70, the states of all of the input lines to logical AND **1391** are TRUE at the first TIME–3. Consequently, when the state of line AD is rendered FALSE at the second TIME–1, line PKϕ experiences a TRUE-to-FALSE reversal of state and thereby presets the "K" digit-register to "zero," all in the manner previously described. Thus, it is evident that, each time the digit-counter is incremented by a change of state of line AD during a digit-cycle operation, the "K" digit-register is unconditionally preset to "zero."

Due to the fact that the state of line CFF (FIG. 61) is rendered TRUE at the first TIME–4 of each digit-cycle and remains TRUE for 160 microseconds, the state of line CYC is thereafter rendered TRUE at each TIME–4 and FALSE at each TIME–1. As previously described, when the state of line CYC is reversed from TRUE to FALSE, the bit-counter (FIG. 62) is incremented by a count of "one," assuming, of course, that the state of line (DBD)′ is previously rendered TRUE, as before stated.

Thus, in the present example, when the state of line CYC is rendered FALSE at the second TIME–1, the bit-counter is effectively advanced thereby from a count of "*d*" to a count of "*a*." As the state of output line MYW (FIG. 60) is rendered TRUE by line MK, and as the state of output line MXW is selectively rendered TRUE by lines MK and MSA, the memory "X" and "Y" drivers and grounders are properly energized at the second TIME–1 in a manner previously described, so that a memory read-write cycle of operation is carried out. Thus, the first-order digit of the word stored in the selected memory address is read out bit by bit and simultaneously stored bit by bit in the "K" digit-register (FIG. 70) via line KL.

At the last TIME-4 after bit "*d*" of the first-order digit of the word is read out, the state of flipflop **6039** is reversed, so that the state of output line CFF is thereby rendered FALSE. Thus, with reference to FIG. 60, as the state of line (CFF)′ is rendered TRUE at the last TIME-4, the state of flipflop **6038** is reversed at the following TIME-1, so that the state of output line SMC is rendered FALSE thereby to effectively de-energize the "X" and "Y" grounders and thereby terminate the read-write cycle after the first-order digit is read. With reference to FIG. 69, the state of the prime input to flipflop **6061** is rendered TRUE at the last TIME-4 when the state of line (CFF)′ is rendered TRUE. Consequently, at the following TIME-1, the state of flipflop **6061** is reversed, so that the state of line TH is thereby rendered FALSE, thus terminating the digit-cycle cycle of operation at the last TIME-1.

68. *Subcommand Subinstructions*

As shown in the preceding chart entitled "SUBINSTRUCTIONS," an "ARO-O-*" subcommand, hereinafter termed "ARO" subcommand, initiates the addition of the word of address-A to a constant and subsequent storage of the result in address-A; a "CPA-O-*" subcommand, hereinafter termed "CPA" subcommand, initiates the complementing of the word of address-A and subsequent storage of the result in address-A; a "CPM-*n*-*" subcommand, hereinafter termed "CPM" subcommand, initiates the complementing of the word stored in memory at the address specified by a particular section of the instruction register, and subsequent storage of the result in the same address in memory; a "DAD-O-*" subcommand, hereinafter termed "DAD" subcommand, initiates the addition of the word of address-B to a constant and subsequent storage of the result in address-B; a "RAD-*n*-*" subcommand, hereinafter called "RAD" subcommand, effects the addition of the word of address-A to a particular word in memory and subsequent storage of the result in address-A; a "RSB-*n*-*" subcommand, hereinafter termed "RSB" subcommand, initiates the subtraction of a word in memory from the word of address-A and subsequent storage of the result in address-A; a "SA(1)-O-*" and a "SA(2)-O-*" subcommand, hereinafter termed "SA" commands, selectively initiates the shifting of the word of address-A to the right or to the left by one digital-order position and subsequent storage of the result in address-A; and a "SB(1)-O-*" and a "SB(2)-O-*" subcommand, hereinafter termed "SB" subcommands, selectively initiates the shifting of the word of address-B to the right or to the left by one digital-order position and subsequent storage of the result in address-B.

As will later be seen, due to the fact that a "RAD" subcommand initiates the addition of a particular word in memory to the word in address-A and subsequent storage of the sum word in address-A, the entire operation essentially comprises ten "double-digit" operations, in that eight read-write cycles of operation are required for each digital-order of the word, rather than four read-write cycles for each digital order, as in a word-cycle mode of operation previously described.

With reference to FIG. 77, when a "RAD" subcommand is to be initiated, the state of flipflop **6097** is properly conditioned, so that the state of output line RAD is rendered TRUE at TIME-1, as a result of line AN being rendered FALSE also at TIME-1 indicating that the previously-initiated cycle of operation has been completed to the extent that a "RAD" subcommand is permitted to be initiated at this time. As shown in the following timing chart, during the first bit time of forty microseconds after line RAD is rendered TRUE, several operations are performed in order to prepare the memory for a read-write cycle of operation beginning at the second TIME-1. For example, at the beginning of the cycle, the state of line $(PW\phi)'$ (FIG. 64) is TRUE. Consequently, when line $(PW\phi)'$ is rendered FALSE at TIME-2, the word-selecting register (FIGS. 64 and 65) is preset to "00." With reference to FIG. 69, line RAD being rendered TRUE at the first TIME-1 causes line FO2 to be rendered TRUE also at TIME-1. With reference to FIG. 60, when line FO2 is rendered TRUE at TIME-1, the state of both the inputs to logical AND **1105** are simultaneously TRUE, so that output line ZA is also rendered TRUE at TIME-1, which, in turn, renders line SOO TRUE at TIME-1.

With reference to FIG. 61, twenty microseconds after line SOO is rendered TRUE, line DDF is rendered TRUE at the first TIME-3 and is thereafter rendered FALSE at the following TIME-4. Consequently, output line IW is rendered TRUE at TIME-3 and FALSE at TIME-4. When line IW is rendered FALSE at TIME-4, the contents of a selected one of sections 2 through 5 of the instruction register is transferred to the word-selecting register in the manner previously described. As shown in FIGS. 69 and 71, the states of lines FOO, JJF, and KKF are rendered TRUE at the first TIME-1 when line RAD is rendered TRUE. Thus, with reference back to FIG. 61, line AD is rendered TRUE at TIME-3 and FALSE at TIME-4 by logical AND **1129**, the same as the state of line IW. When line AD is rendered FALSE at TIME-4, the digit-counter (FIG. 63) is effectively advanced from a count of "9" to a count of "0," also in a manner previously described, so as to permit the first eight read-write cycles to take place with respect to the first-order digits of the two operand words previously stored in memory.

As line DDF is rendered TRUE at the first TIME-3, the reference input to flipflop **6039** experiences a TRUE-to-FALSE reversal of state at TIME-4, so that output line CFF is rendered TRUE also at TIME-4. Consequently, as a result of line CFF being TRUE at TIME-4, output line CYC is thereafter rendered TRUE at each TIME-4 and FALSE at each TIME-1. As previously described with respect to the bit-counter (FIG. 62), each time the state of line CYC reverses from TRUE to FALSE, the count thereof is advanced by one binary bit. Therefore, at the second TIME-1, the bit-counter is essentially advanced from a count of "*d*" to a count of "*a*." As shown in FIGS. 70 and 71, lines JJF and KKF are both rendered TRUE at the first TIME-1 by line RAD, and preset lines $PJ\phi$ (FIG. 71) and $PK\phi$ (FIG. 70) experience a TRUE-to-FALSE reversal of state when line AD is rendered FALSE at the first TIME-4. Consequently, both the "J" and "K" digit-registers are respectively preset to "0" by lines $PJ\phi$ and $PK\phi$ at the first TIME-4 when line AD is rendered FALSE. With reference to FIG. 53, also at TIME-4 when line AD is rendered FALSE, the prime input to flipflop **6000** experiences a TRUE-to-FALSE reversal of state, so that line PCI is thereby rendered FALSE. Thus, in a sense, carry flipflop **6000** of the adder-subtracter unit is preset to "0." As shown in FIG. 79, when line DDF is rendered FALSE at the first TIME-4, the state of flipflop **6105** is reversed, so that output line SHA is rendered TRUE thereby. When line SHA is rendered TRUE at the first TIME-4, digit-cycle lines MK (FIG. 69) and AJ (FIG. 71) are both rendered TRUE thereby also at the first

TIME–4. With reference to FIG. 60, when line SHA is rendered TRUE at the first TIME–4, line (SHA)' being simultaneously rendered FALSE, output line ZA is rendered FALSE.

As shown in FIG. 60, as line MK is rendered TRUE at TIME–4, output line MYW is also rendered TRUE for the first time at TIME–4. It is to be noted that, at the first TIME–1 when line ZA was rendered TRUE, output line AYW is also rendered TRUE thereby. Consequently, due to the fact that line AJ is rendered TRUE at TIME–4, output line AYW remains TRUE even though line ZA is rendered FALSE at TIME–4. When line CFF is rendered TRUE at the first TIME–4, all the inputs to logical AND **1094** are simultaneously TRUE. Consequently, at the second TIME–1, the state of flipflop **6038** is reversed, so that output line SMC is rendered TRUE thereby. Immediately thereafter, a sequence of four read-write cycles of operation are carried out in essentially the same manner as previously described with respect to that portion of the description entitled "Detailed Read-Write Cycle."

During the first four read-write cycles, the four binary bits of the first-order digit of the word stored in address-A is sequentially read out and simultaneously stored bit by bit in the "J" digit-register (FIG. 71) via gate **1419** and line JL. Simultaneously therewith, the first four binary bits of the first-order digit of the word stored in a selected one of memory addresses $\phi\phi$ through 99 are sequentially read out and simultaneously stored bit by bit in the "K" digit-register (FIG. 70) via gate **1383** and line KL. As just seen, it is possible to simultaneously read from a selected one of memory addresses A or B and a selected one of memory addresses $\phi\phi$ through 99. However, as before mentioned, a different word cannot be simultaneously written in the two selected addresses. Since it is desired that the selected word in one of memory addresses $\phi\phi$ through 99 be preserved, the first-order digit just read from one of memory addresses $\phi\phi$ through 99 is simultaneously stored in the first-order digital positions of address-A and the selected one of addresses $\phi\phi$ through 99. Consequently, with reference to FIG. 60, the information to gate **1038** from line ASA is essentially suppressed due to the fact that the state of line (RAD)' is FALSE during the carrying-out of an "RAD" subcommand. This type of suppression is also necessary during the carrying-out of a "RSB" subcommand, which is to be described hereinafter.

During the last bit time of the fourth read-write cycle, the bit-counter (FIG. 62) is at a count of "*d*," so that output lines BC*a* and BC*b* from flipflops **6041** and **6042**, respectively, are TRUE. Consequently, with reference to FIG. 61, at TIME–4 of the last bit time, the state of the prime input to flipflop **6039** is reversed from TRUE to FALSE, and, as a result, flipflop **6039** reverses state, so that output line CFF is rendered FALSE at TIME–4 of the fourth read-write cycle and line CYC is thereafter prevented from subsequently effecting advancement of the count of the bit-counter. When line CFF is rendered FALSE, line (CFF)' being rendered TRUE, the state of the prime input to flipflop **6038** (FIG. 60) is reversed at TIME–1 following completion of the four read-write cycles, and, as a result, the state of flipflop **6038** is reversed, so that output line SMC is thereafter FALSE. Also at TIME–4 when line (CFF)' is rendered TRUE, lines DDF and IW (FIG. 61) are rendered TRUE at the following TIME–3 and FALSE ten microseconds later at TIME–4. With reference to FIG. 79, when the output of gate **1699** is rendered FALSE at TIME–4 by line DDF, the state of flipflop **6105** is reversed, so that output line SHA is rendered FALSE thereby. When SHA is rendered FALSE at TIME–4, line MK (FIG. 69) is thereby rendered FALSE, line AJ (FIG. 71) is thereby rendered FALSE, and line ZA (FIG. 60) is thereby rendered TRUE.

With reference to FIG. 60, it is to be noted that, when line MK is rendered FALSE, line MYW is rendered FALSE thereby, and, consequently, the "Y" drivers for memory addresses $\phi\phi$ through 99 are prevented from being energized. However, even though line AJ was rendered FALSE at TIME–4, line ZA was simultaneously rendered TRUE. Therefore, line AYW remains TRUE, and, consequently, the "Y" driver for address-A is permitted to be selectively energized.

With reference to FIG. 61, when the output of logical AND **1116** is rendered FALSE at TIME–4, the state of flipflop **6039** is reversed, so that output line CFF is rendered TRUE. Thus, at TIME–4 when line CFF is rendered TRUE, line CYC is likewise rendered TRUE and is rendered FALSE ten microseconds later at TIME–**1**. When line CYC experiences a TRUE-to-FALSE reversal of state at TIME–1, the bit-counter (FIG. 62) is effectively advanced from a count of "*d*" to a count of "*a*." Also at TIME–1, line SMC (FIG. 60) is rendered TRUE by a reversal of state of flipflop **6038**.

As is evident from the preceding portion of the description relating to the adder-subtracter portion of the computer (FIGS. 53 and 54), immediately following the simultaneous copying into the "J" and "K" digit-registers of bit "*d*" from each of the first-order digits of the word originally stored in one of memory addresses $\phi\phi$ through 99 and the word stored in one of addresses A and B, the sum and carry digits are immediately available at the output of the adder-subtracter unit and ready for respective storage into address-A and carry flipflop **6900** (FIG. 53). Consequently, immediately following the second forty-microsecond interval, four read-write cycles are sequentially initiated to effect the copying of the sum digit into address-A and the carry digit into the carry flipflop of the adder-subtracter unit.

As previously described with respect to FIG. 54, the states of output lines AN*a* through AN*d* of the decoder portion of the adder-subtracter unit collectively represent the binary value of the sum digit, whereas the state of output line EAS indicates whether or not there is a decimal carry digit. As shown in the upper right-hand corner of FIG. 54, output lines AN*a* through AN*d* of the adder-subtracter unit are each logically ANDED with a corresponding one of lines B*a* through B*d* from the bit-counter (FIG. 62) in a manner such that, during the first forty-microsecond read-write cycle while the bit-counter is at a count of "*a*," the state of line Z is TRUE if binary bit "*a*" of the sum digit is a "1," and is FALSE if binary bit "*a*" is a "0." During the second forty-microsecond read-write cycle while the bit-counter is at a count of "*b*," line Z is TRUE if bit "*b*" of the sum digit is a "1" and is FALSE if bit "*b*" is a "0," and so on for bit "*c*" and bit "*d*" of the sum digit.

Therefore, with reference to FIG. 60, as line ZA was previously rendered TRUE at TIME–4 preceding the initiation of the second sequence of four read-write cycles, output line MXW is thereafter rendered TRUE when line Z is rendered TRUE, the outputs from the two memory sense amplifiers, via lines ASA and MSA, being respectively inhibited by logical ANDS **1088** and **1091**, and, thus have no effect on the state of line MXW during the second sequence of read-write cycles. Consequently, the corresponding one of "X" driver lines XD*a* through XD*d*, which are selected by lines B*a*M through B*d*M, are permitted to be energized only when the binary bit to be stored in address-A is a "1."

After the four binary bits of the sum digit is stored in the first-order digital position of address-A, the state of the prime input to flipflop **6039** (FIG. 61) is reversed at TIME–4 of the fourth read-write cycle. Consequently, the state of flipflop **6039** is reversed, and output line CFF is rendered FALSE thereby. At the following

TIME–1 after line CFF is FALSE, the state of flipflop **6038** (FIG. 60) is reversed, and line SMC is rendered FALSE thereby. When line SMC is rendered FALSE at TIME–1, line XDW is likewise rendered FALSE.

During the forty-microsecond interval following the second sequence of four read-write cycles, the count of the digit-counter is advanced from "0" to "1," and the "J" and "K" digit-registers are again preset to "0" preparatory to the storage therein of the second-order digits of the two operands located in memory. Thereafter, the second-order digits of the two operands are added, together with the previous carry digit, and the new sum digit is stored in the second-order digital position of address-A in exactly the same sequence as just described with respect to the two first-order digits. Thereafter, the sequence of events is again sequentially repeated until the ten-digit sum of the two ten-digit operands is stored in memory at address-A.

With reference to FIG. 61, at TIME–4 during the last bit time of the addition operation, the state of the reference input to flipflop **6039** is reversed, so that output line CFF is rendered FALSE at TIME–4 by the reversal of state of flipflop **6039**. As all of the inputs to logical AND **1119** are simultaneously TRUE at the last TIME–4, ten microseconds later at TIME–1, the state of flipflop **6040** is reversed, and the state of output line PC is thereby rendered TRUE for a period of forty microseconds. With reference to FIG. 60, when line (CFF)′ is rendered TRUE at TIME–4, ten microseconds later at TIME–1, the state of flipflop **6038** is reversed and renders output line SMC FALSE. When line SMC is rendered FALSE at the last TIME–1 of the subcommand, the "X" and "Y" drivers and grounders for the memory are prevented from subsequently being selectively energized, thus terminating the last read-write cycle of operation.

As shown in FIG. 61, as line PC is TRUE at this time, line AN is rendered TRUE at the last TIME–4 for a period of ten microseconds until the last TIME–1, at which time line AN is rendered FALSE. Therefore, with reference to FIG. 77, when line AN is rendered FALSE at the last TIME–1 indicating completion of the "RAD" subcommand, the state of the prime input to flipflop **6097** is reversed from TRUE to FALSE and thereby causes a corresponding reversal of state of flipflop **6097**, which, in turn renders output line RAD FALSE, thus terminating the sequence of operations of the "RAD" subcommand.

In summary, like all other subcommands and word-

cycles, subcommand "RAD" is essentially turned "ON" at TIME–1. During the first bit time of forty microseconds after subcommand "RAD" is "ON," the control circuitry for the memory is properly conditioned for simultaneous reading and writing with respect to address-A and a selected one of addresses ϕϕ through 99. Thereafter, a sequence of four read-write cycles is initiated, during which the first-order digit of the word in address-A is stored in the "J" digit-register, and, simultaneously therewith, the first-order digit of the word in a selected one of memory addresses ϕϕ through 99 is stored in the "K" digit-register. Following the first sequence of four read-write cycles, the sum and carry digits of the first-order digits of the two operands appear at the output of the adder-subtractor unit. Therefore, during the next bit time of forty microseconds following the first sequence of read-write cycles, the control circuitry for the memory is properly conditioned preparatory to writing the low-order sum digit into the low-order digital position of address-A. Thereafter, a second sequence of four read-write cycles is initiated, during which time the sum digit is stored in the proper digital position of address-A.

The above sequence of events is sequentially repeated for each of the remaining digits of the two operands until the sum of the two operands is derived and stored in address-A. After the addition operation is completed, line RAD remains TRUE for one more bit time of forty microseconds for other control purposes and then is rendered FALSE. It is seen, therefore, that a "RAD" subcommand comprises twenty digit-cycles, each of which is five binary bits in length, plus one spare bit at the end of the subcommand for control purposes. Consequently, the total time required to complete a "RAD" subcommand is 101 binary bit times of forty microseconds each, or approximately .004 second.

The mode of operation of each of subcommands "RSB" (Regular Subtract), "CPA" (Complement in "A"), "CPM" (Complement in Memory Addresses ϕϕ through 99), "DAD" (Add Constant to "B"), and "ARO" (Add Constant to "A") is essentially the same as for a "RAD" subcommand except for minor exceptions, and, consequently, a detailed description thereof is not deemed necessary for a full and complete understanding thereof. An "RSB" subcommand, for example, essentially differs only in that subtract line SAS of the adder-subtractor unit (FIGS. 53 and 54) is rendered TRUE by line RSB at the beginning of the "RSB" subcommand.

TIMING CHART FOR FIRST-ORDER DIGITAL PORTION OF "RAD" SUBCOMMAND

TIME	RAD	(PWϕ)′	FO2	Za	SOO	DDF	IW	FOO	KKF	AD	CFF	CYC	PJϕ
1	T	T	T	T	T	F	F	T	T	F	F	F	F
2	T	F	T	T	T	F	F	T	T	F	F	F	F
3	T	T	T	T	T	T	T	T	T	T	F	F	T
4	T	T	T	F	T	F	F	T	T	F	T	T	F
1	T	T	T	F	T	F	F	T	T	F	T	F	F
4	T	T	T	F	T	F	F	T	T	F	F	F	F
1	T	T	T	F	T	F	F	T	T	F	F	F	F
2	T	T	T	F	T	F	F	T	T	F	F	F	F
3	T	T	T	F	T	T	T	T	T	F	F	F	F
4	T	T	T	T	T	F	F	T	T	F	T	T	F
1	T	T	T	T	T	F	F	T	T	F	T	F	F
4	T	T	T	T	T	F	F	T	T	F	F	F	F
1	T	T	T	T	T	F	F	T	T	F	F	F	F

TIMING CHART FOR FIRST-ORDER DIGITAL PORTION OF "RAD" SUBCOMMAND—Continued

TIME	PKϕ	JJF	SHA	MK	AJ	MYW	AYW	SMC	XDW	
1	F	T	F	F	F	F	T	F	F	1st. Forty Microsecond Interval
2	F	T	F	F	F	F	T	F	F	
3	T	T	F	F	F	F	T	F	F	
4	F	T	T	T	T	T	T	F	F	
1	F	T	T	T	T	T	T	T	T	1st. Four Read-Write Cycles (160 μsec.)
4	F	T	T	T	T	T	T	T	T	
1	F	T	T	T	T	T	T	F	F	2nd. Forty Microsecond Interval
2	F	T	T	T	T	T	T	F	F	
3	F	T	T	T	T	T	T	F	F	
4	F	T	F	F	F	F	T	F	F	
1	F	T	F	F	F	F	T	T	T	2nd. Four Read-Write Cycles (160 μsec.)
4	F	T	F	F	F	F	T	T	T	
1	F	T	F	F	F	F	T	F	F	

Consequently, the difference between the two digits in the "J" and "K" digit-registers is obtained each time instead of the sum thereof, all of which is fully described previously with respect to that portion of the description relating to the adder-subtracter unit.

In order to carry out a "CPA" subcommand, the word stored in address-A is complemented and thereafter stored in address-A. To accomplish this, each digit of the word in address-A is subtracted from "0," and the difference digit is stored in the proper digital position of address-A, with the carry digit being stored in the carry flipflop of the adder-subtracter unit, as previously described. To accomplish this, "0" is obtained by presetting the "J" digit-register to "0" at the beginning of the subcommand and, thereafter, leaving the "J" digit-register in the "preset to zero" condition during the entire subcommand. Consequently, during each sequence of four read-write cycles, the particular order digit of address-A is stored in the "K" digit-register. Thereafter, the digit is subtracted from "0" by the adder-subtracter unit, and, during the second sequence of four read-write cycles, the remainder digit is stored in the proper digital position of address-A. A "CPM" subcommand is carried out in a similar manner; however, in carrying out a "DAD" subcommand, the word in address-B is selectively added to either of the constants 0000000001 or 0000000005, and, therafter, the sum thereof is stored in address-B. This is accomplished by first presetting the "K" digit-register to zero and immediately thereafter presetting the "J" digit-reigister to "1" or "5" at the beginning of the initiation of the subcommand, depending upon the particular constant desired to be added to the word in address-B. After the first-order digit of the word in address-B is stored in the "K" digit-register, the sum digit derived from the contents of the "J" and "K" digit-registers is stored in the first-order digital position of address-B. Thereafter, the "J" digit-register is preset to "0" and remains in the "preset to zero" condition during the remainder of the subcommand. An "ARO" subcommand selectively effects the addition of the word of address-A to the constant 0000000005 and, thereafter, effects storage of the sum in address-A instead of address-B in substantially the same manner as a "DAD" command.

Like the previously-described subcommands, a "SA" and a "SB" subcommand comprises twenty digit-cycles, each five bit times in length, plus one bit time at the end of the subcommand for control purposes. Thus, the length of a "SA" and a "SB" subcommand is 101 bit times of forty microseconds each, or 4.04 milliseconds. In carrying out the dictates of a "SA" subcommand, each of the ten decimal digits of the word in address-A is effectively shifted, either to the next higher-order digital position or to the next lower-order digital position therein, which, in effect, respectively multiplies or divides the word in address-A by "ten." An "SB" subcommand effects the shifting of the word in address-B in a similar manner.

Before describing in detail the modes of operation effected by the "SA" and "SB" subcommands, a brief description of each will now be given in order to facilitate a full and complete understanding thereof.

If it is assumed that the word in address-A is to be shifted one place to the left, the "K" digit-register is first preset to "0" at the beginning of the cycle of operation. Thereafter, the digit-counter is advanced from a count of "9" to a count of "0," and the "J" digit-register is simultaneously preset to "0." After the "J" digit-register is preset to "0," the first-order digit of the word in address-A is transferred to the "J" digit-register, and the contents of the "K" digit-register—i.e., "0"—is stored in the first-order digital position of address-A. Thereafter, the digit-counter is advanced from a count of "0" to a count of "1," and the "K" digit-register is again preset to "0," even though, in fact, the digit-register is already effectively storing a "0.". After the "K" digit-register is preset to "0," the second-order digit of the word in address-A is transferred to the "K" digit-register, and the first-order digit of the word, previously stored in the "J" digit-register, is stored in the second-order digital position of address-A. Thereafter, the digit-counter is advanced from a count of "1" to a count of "2," and the "J" digit-register is again preset to "0." After the "J" digit-register is preset to "0," the third-order digit of the word in address-A is stored therein, and the second-order digit in the "K" digit-register is stored in the third-order digital position of address-A. Thereafter, the digit counter is advanced from a count of "2" to a count of "3," and the "K" digit-register is again preset to "0." The just-stated sequence of events is repeated until each of the digits of the word in address-A is sequentially shifted to the next higher-order digital position therein.

In order to shift the word in address-A one digital position to the right, instead of to the left, as just described, both the bit-counter and the digit-counter are each effectively operated in a reverse direction, and, as a result, reading and writing with respect to address-A are also in reverse order. That is, bit "d" of the tenth-order digit of the word in address-A is the first to be read out, and bit "a" of the first-order digit of the word in address-A is the last to be read out. Therefore, at the beginning of the operation, the "K" digit-register is preset to "0," and the digit-counter is decremented from a count of "0" to a count of "9." Thereafter, the "J" digit-register is preset to "0," after which the tenth-order digit is stored in the "J" digit-register. After the tenth-order digit is transferred to the "J" digit-register, the contents of the "K" digit-register—i.e., "0"—is stored in the tenth-order digital position of address-A. Thereafter, the digit-counter is decremented from a count of "9" to a count of "8," and the "K" digit-register is preset to "0," and so on, until the second-order digit of the word of address-A is shifted to the first-order digital position therein.

It is to be noted that, in order to preserve the sign of a negative number whose complement is to be shifted a prescribed number of places to the right, the "K" digit-register is initially preset to "9," and the numeral "9" is stored in the tenth-order digital position of the word at the beginning of each shifting operation of the word. Otherwise, the sign is not preserved.

As the word in address-B is shifted either to the left or to the right via an "ASB" subcommand in essentially the same manner as just described with respect to address-A, a further description thereof is not deemed necessary for a full and complete understanding of the mode of operation thereof.

As will be seen later, in carrying out a "multiply-dollar-decimal" (MCC—**12**) or a "multiply-and-shift" (MUS—**13**) instruction, the ten-digit words stored in addresses "A" and "B" are effectively combined to form a twenty-digit word, whereby address-B contains the ten lower-order digits and address–A contains the ten higher-order digits of the twenty-digit word. In order to shift such a twenty-digit word one place to the right, a "SA" subcommand is initiated, whereby the word in address-A is shifted to the right in the same manner as just described. However, at the conclusion of the "SA" subcommand, the original first-order digit of the word is address-A is stored in the "K" digit-register, and a "SB" subcommand is thereafter initiated. In this instance, however, the "K" digit-register is not initially preset to "0" at the beginning of the "SB" subcommand. Consequently, when the word in address-B is shifted one place to the right, in the manner just described, the original first-order digit of the word in address-A, which is stored in the "K" digit-register, is stored in the tenth-order digital position of address-B. Thus, each of the twenty digits of the twenty-digit word is effectively shifted to the next lower-order digital position of addresses "A" and "B."

More specifically, a "SA" subcommand for shifting left is initiated and carried out in the following manner: With reference to FIG. 78, and with additional reference to the following timing chart, in initiating a "SA" subcommand, the reference input to flipflop **6101** is properly conditioned, so that, when line AN experiences a TRUE-to-FALSE reversal of state, indicating completion of the previously-initiated sequence of events to the extent that a "SA" subcommand is permitted to be initiated thereafter, output line SA is rendered TRUE by a reversal of state of flipflop **6101** at the first TIME–**1**. With reference to the lower left-hand corner of FIG. 62, the state of line GOO is rendered TRUE at TIME–**1** by line SA and remains TRUE as long as the state of line SA remains TRUE. With reference to the lower left-hand corner of FIG. 71, the states of lines JA and JJF are also rendered TRUE at TIME–**1** by line SA. As shown in FIG. 60, the state of line SOO is rendered TRUE at TIME–**1** due to the fact that the state of line JA is rendered TRUE at that time. Therefore, at the following TIME–**3**, all the inputs to logical AND **1113** (FIG. 61) are simultaneously TRUE, thus rendering line DDF TRUE at TIME–**3**. Ten microseconds later, at TIME–**4**, the state of line DDF is rendered FALSE by line C34, and, consequently, the state of flipflop **6039** is reversed, so that the state of output line CFF is rendered TRUE thereby.

At TIME–**4**, when the state of line CFF is rendered TRUE, the state of line CYC is rendered TRUE and remains TRUE for ten microseconds, at which time line CYC is rendered FALSE. Thus, when line CYC experiences a TRUE-to-FALSE change of state, the bit-counter (FIG. 62) is incremented from a count of "d" to a count of "a," in the same manner as previously described. With reference to FIG. 79, when the state of line DDF is rendered FALSE at TIME–**4**, the state of flipflop **6105** is reversed, so that output line SHA is rendered TRUE thereby at TIME–**4**. As shown in FIG. 71, when line SHA is rendered TRUE, the state of line AJ is rendered TRUE thereby, and, simultaneously therewith, the state of line JA is rendered FALSE thereby.

At TIME–**3** when the state of line DDF is rendered TRUE, line AD (FIG. 61) is rendered TRUE for a period of ten microseconds, after which time line AD is rendered FALSE. When line AD experiences a TRUE-to-FALSE change of state at TIME–**4**, the digit-counter (FIG. 63) is advanced from a count of "9" to a count of "0," in the same manner as previously described. At TIME–**4** when the state of line AD is reversed from TRUE to FALSE, the states of preset lines PJϕ (FIG. 71) and PKϕ (FIG. 70) are both reversed thereby from a TRUE state to a FALSE state. Consequently, the "J" and "K" digit-registers are both preset to "0" at TIME–**4**, in the same manner as previously described. At the second TIME–**1**, the state of flipflop **6038** (FIG. 60) is reversed, and output line SMC is rendered TRUE thereby and thus effects a read-write cycle, in substantially the same manner as previously described with respect to a "RAD" subcommand, to transfer the first-order bit of the first-order digit of the word in address–A, and subsequent storage of the bit in the "J" digit-register (FIG. 71), via selective energization of line JL, all in the manner previously described.

At the third TIME–**1**, the bit-counter is effectively advanced from a count of "a" to a count of "b," a second read-write cycle is initiated, and, consequently, the second-order bit of the first-order digit in address-A is stored in the "J" digit-register. At the fourth TIME–**1**, the bit-counter is again incremented, a third read-write cycle is initiated, and the third-order bit of the first-order digit of the word in address-A is stored in the "J" digit-register. Finally, at the fifth TIME–**1**, the bit-counter is incremented to a count of "d," a fourth read-write cycle is initiated, and, as a result, the fourth-order bit of the first-order digit of the word in address-A is stored in the "J" digit-register.

At TIME–**4** during the fourth read-write cycle, the state of flipflop **6039** (FIG. 61) is reversed, and line CFF is thereby rendered FALSE. Ten microseconds later at the following TIME–**1**, the state of flipflop **6038** (FIG. 60) is reversed, and line SMC is thereby rendered FALSE to prevent a read-write cycle from being initiated at that time. At the following TIME–**3**, thirty microseconds after the state of line CFF is rendered FALSE, the state of line DDF (FIG. 61) is rendered TRUE and remains TRUE until the following TIME–**4**, at which time line DDF is rendered FALSE. When line DDF is rendered FALSE at TIME–**4**, the state of flipflop **6105** (FIG. 79) is reversed, so that output line SHA is thereby rendered FALSE at TIME–**4**. Also at TIME–**4** when the state of line SHA is rendered FALSE, the state of flipflop **6101** is reversed, so that output line SHB is thereby rendered TRUE at TIME–**4**.

With reference to FIG. **71**, when the state of line (SHB)′ is rendered FALSE, the state of line AJ is thereby rendered FALSE. However, with reference to FIG. 69, when the state of line SHB is rendered TRUE, the state of line KA is thereby rendered TRUE, and,consequently, the state of line SOO (FIG. 60) remains unchanged. Also at TIME–4 following the fourth read-write cycle, the state of flipflop **6039** (FIG. 61) is reversed, so that the state of line CFF is thereby rendered TRUE. With reference to FIG. 60, ten microseconds after the state of line CFF is rendered TRUE, the state of flipflop **6038** is reversed, so that the state of line SMC is rendered TRUE at TIME–1. With reference to FIG. 61, ten microseconds after the state of line CFF is rendered TRUE, line CYC experiences a TRUE-to-FALSE reversal of state and thereby advances the bit-counter (FIG. 62) from a count of "*d*" to a count of "*a*." Consequently, at TIME–1 there is again initiated a sequence of four read-write cycles whereby the digit stored in the "K" digit-register—i.e., "0"—is stored in the first-order digital position of address-A.

At TIME–4 following the second sequence of four read-write cycles, the bit-counter being at a count of "*d*," the state of flipflop **6039** (FIG. 61) is reversed, so that the state of line CFF is thereby rendered FALSE. At the following TIME–1 after the state of line CFF is rendered TRUE, the state of flipflop **6038** (FIG. 60) is reversed, so that the state of output line SMC is thereby rendered FALSE. As shown in FIG. 61, at the following TIME–3 after the state of line (CFF)′ is rendered TRUE, the state of line DDF is rendered TRUE for a period of ten microseconds and then is rendered FALSE at the following TIME–4. At TIME–4 when line DDF is rendered FALSE, the state of flipflop **6105** (FIG. 79) is reversed, so that the state of line SHA is thereby rendered TRUE. However, at this time the state of line SHB does not reverse, but, instead, remains TRUE.

With reference to FIG. 61, at TIME–3 when the state of line DDF is rendered TRUE, the state of line AD is rendered TRUE for a period of ten microseconds, after which period the state of line AD is rendered FALSE. When line AD experiences a TRUE-to-FALSE change of state at TIME–4, the digit-counter (FIG. 63) is incremented from a count of "0" to a count of "1," in a manner previously described. At TIME–4 when the state of line (SHA)′ is rendered FALSE, the states of lines KA and KKF (FIG. 69) are thereby rendered FALSE. However, at TIME–4 when the state of line SHA is rendered TRUE, the state of line AK is thereby rendered TRUE, and, consequently, the state of line SOO (FIG. 60) remains TRUE. With reference to FIG. 70, at TIME–4 when lines AD and KKF experience TRUE-to-FALSE changes of states, the state of line PKϕ is reversed from TRUE to FALSE and effects the presetting of the "K" digit-register to "0."

Also at time-4, the state of flipflop **6039** (FIG. 61) is reversed, so that the state of line CFF is thereby rendered TRUE. When the state of line CFF is rendered TRUE, the state of output line CYC is also rendered TRUE and remains TRUE for a period of ten microseconds, at which time the state of line CYC is rendered FALSE. When the state of line CYC experiences a TRUE-to-FALSE reversal of state at TIME–1, the bit-counter (FIG. 62) is advanced from a count of "*d*" to a count of "*a*." Also at TIME–1, the state of flipflop **6038** (FIG. 60) is reversed, so that the state of line SMC is thereby rendered TRUE. Consequently, at TIME–1, a third sequence of four read-write cycles is initiated, in the manner previously described, so that the four binary bits of the second-order digit of the word in address-A are read out and stored in the "K" digit-register (FIG. 70) via lines ASA and KL.

At the following TIME–4 after the bit-counter has reached a count of "*d*," the state of flipflop **6039** (FIG. 61) is reversed, and the state of line CFF is thereby rendered FALSE. Ten microseconds later, the state of flipflop **6038** (FIG. 60) is reversed, and the state of line SMC is rendered FALSE at the following TIME–1 to prevent a read-write cycle from being initiated during the following control period. With reference to FIG. 61, at the following TIME–3, the state of line DDF is again rendered TRUE and remains TRUE for a period of ten microseconds, after which time the state of line DDF is rendered FALSE at the following TIME–4.

With reference to FIG. 79, at TIME–4 when line DDF experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6105** is reversed, so that the state of output line SHA is thereby rendered FALSE. Consequently, when line SHA experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6106** is reversed, so that the state of output line SHB is also rendered FALSE at TIME–4. Consequently, the state of line AK (FIG. 69) is rendered FALSE at TIME–4 when the states of lines SHA and SHB are rendered FALSE. However, at the same time, the states of lines JA and JJF (FIG. 71) are rendered TRUE. Consequently, the state of line SOO (FIG. 60) remains TRUE. Also at TIME–4, when the state of line DDF is rendered FALSE, the state of flipflop **6039** (FIG. 61) is reversed, so that the state of output line CFF is thereby rendered TRUE. When the state of line CFF is rendered TRUE at TIME–4, the state of line CYC again is rendered TRUE and remains TRUE for a period of ten microseconds, after which period its state is rendered FALSE. Consequently, when line CYC experiences a TRUE-to-FALSE reversal of state at TIME–1, the bit-counter (FIG. 62) is advanced from a count of "*d*" to a count of "*a*." With reference to FIG. 60, the state of flipflop **6038** is reversed at TIME–1 and thereby renders the state of line SMC TRUE.

Thus, at TIME–1 following the fourth forty-microsecond control period, a fourth sequence of four read-write cycles is initiated, so that the first-order digit that was originally stored in address-A, but now is stored in the "J" digit-register, is stored in the second-order digital position of address-A. Consequently, it is seen that the digit originally stored in the first-order digital position of address-A is now shifted to the second-order digital position therein, and a "0" is stored in the first-order digital position of address-A.

After completion of the fourth sequence of read-write cycles, the count of the bit-counter is advanced to "*d*." Therefore, at the following TIME–4, the state of flipflop **6039** (FIG. 61) is reversed, so that output line CFF is thereby rendered FALSE. At the following TIME–1 after the state of line CFF is rendered FALSE, the state of flipflop **6038** (FIG. 60) is reversed, so that the state of output line SMC is thereby rendered FALSE to prevent further read-write cycles from being initiated during the following forty-microsecond control period.

The state of line DDF (FIG. 61) is rendered TRUE from TIME–3 to TIME–4 during the following control period and effects advancement of the digit-counter (FIG. 63), via line AD, from a count of "1" to a count of "2." Thereafter, the "J" digit-register (FIG. 71) is preset to "0," and the third-order digit of the word in address-A is stored therein.

The just-described sequence of events is sequentially repeated until each of the digits of the word stored in address-A has been shifted to the next higher-order digital position therein, hereinafter known as "shifting to the left." After each of the digits of the word in address-A has been shifted to the next higher-order digital position therein, the digit-counter is, at that time, at a count of "9," the bit-counter is at a count of "*d*," and the states of lines SHA and SHB are FALSE, thus indicating the end of the shift cycle. Therefore, at TIME–1 of the following forty-microsecond control period, the state of flipflop **6040** (FIG. 61) is reversed, so that output line PC is thereby rendered TRUE. Consequently, the state of line AN is rendered TRUE at the following TIME–4 and remains TRUE for a period of ten microseconds, after which period line AN is rendered FALSE. When line AN experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6101** (FIG. 78) is reversed, so that the state of line SA is thereby rendered FALSE, thus completing the cycle of operation of the "SA" subcommand.

PARTIAL TIMING CHART FOR "SA" SUBCOMMAND

TIME	SA	GOO	JJF	JA	SOO	DDF	CFF	CYC	SHA	SHB	AJ	AD	PJϕ	PKϕ	SMC	KA	KKF	AK	
1	T	T	T	T	T	F	F	F	F	F	F	F	F	F	F	F	F	F	
2	T	T	T	T	T	F	F	F	F	F	F	F	F	F	F	F	F	F	40 μsec. Control Period
3	T	T	T	T	T	T	F	F	F	F	F	T	T	T	F	F	F	F	
4	T	T	F	F	T	F	T	T	T	F	T	F	F	F	F	F	F	F	
1	T	T	F	F	T	F	T	F	T	F	T	F	F	F	T	F	F	F	
																			"AJ" Read-Write Digit-Cycle (160 μsec.)
4	T	T	F	F	T	F	F	F	T	F	T	F	F	F	T	F	F	F	
1	T	T	F	F	T	F	F	F	T	F	T	F	F	F	F	F	F	F	
2	T	T	F	F	T	F	F	F	T	F	T	F	F	F	F	F	F	F	40 μsec. Control Period
3	T	T	F	F	T	T	F	F	T	F	T	F	F	F	F	F	F	F	
4	T	T	F	F	T	F	T	T	F	T	F	F	F	F	F	T	T	F	
1	T	T	F	F	T	F	T	F	F	T	F	F	F	F	T	T	T	F	
																			"KA" Read-Write Digit-Cycle (160 μsec.)
4	T	T	F	F	T	F	F	F	F	T	F	F	F	F	T	T	T	F	
1	T	T	F	F	T	F	F	F	F	T	F	F	F	F	F	T	T	F	
2	T	T	F	F	T	F	F	F	F	T	F	F	F	F	F	T	T	F	40 μsec. Control Period
3	T	T	F	F	T	T	F	F	F	T	F	T	F	T	F	T	T	F	
4	T	T	F	F	T	F	T	T	T	T	F	F	F	F	F	F	F	T	
1	T	T	F	F	T	F	T	F	T	T	F	F	F	F	T	F	F	T	
																			"AK" Read-Write Digit-Cycle (160 μsec.)
4	T	T	F	F	T	F	F	F	T	T	F	F	F	F	T	F	F	T	
1	T	T	F	F	T	F	F	F	T	T	F	F	F	F	F	F	F	T	
2	T	T	F	F	T	F	F	F	T	T	F	F	F	F	F	F	F	T	40 μsec. Control Period
3	T	T	F	F	T	T	F	F	T	T	F	F	F	F	F	F	F	T	
4	T	T	T	T	T	F	T	T	F	F	F	F	F	F	F	F	F	F	
1	T	T	T	T	T	F	T	F	F	F	F	F	F	F	T	F	F	F	
																			"JA" Read-Write Digit-Cycle (160 μsec.)
4	T	T	T	T	T	F	F	F	F	F	F	F	F	F	T	F	F	F	
1	T	T	T	T	T	F	F	F	F	F	F	F	F	F	F	F	F	F	40 μsec. Control Period

If, however, the word in address-A is to be shifted to the right instead of to the left, a detailed description of the mode of operation of the subcommand capable of initiating the required sequence of events is obtained simply by modifying the previous "shifting left" description by substituting B*d* for each occurrence of B*a*, B*c* for each occurrence of B*b*, B*b* for each occurrence of B*c*, B*a* for each occurrence of B*d*, D9 for each occurrence of Dϕ, D8 for each occurrence of D1, D1 for each occurrence of D8, and Dϕ for each occurrence of D9.

If the word in address-B is to be shifted instead of the word in address-A—i.e., via a "SB" subcommand—the preceding descriptions are to be modified by substituting SB for SA, JB for JA, BJ for AJ, KB for KA, and BK for AK.

69. *Instructions Generally*

As previously described, there are eighteen basic instructions to which the computer is responsive. Five of these basic instructions concern the ledger card sensing equipment and the accounting machine portion of the computer; one is concerned with the reading of punched paper tape; and twelve are concerned with arithmetic and control functions within the computer.

The first step to be carried out in the execution of each instruction is the reading-out from memory of the next instruction word in the program and storage of the instruction word in the instruction register. After the

instruction word is stored in the instruction register, the two decimal digits stored in section 1 thereof determine the particular type of instruction, or sequence of operations, to be executed next. For example, if the two-digit decimal number in section 1 of the instruction register is "00," an "enter-keyboard-words" instruction is executed, whereby all data thereafter transferred between data-handling sections of the computer is in accordance with a particular sequence pattern dictated by this particular instruction.

The execution of each instruction concludes by effecting the storage in the word-selecting register of the address in memory of the next regular instruction word, the address in memory of the next regular instruction word being specified by the contents of section 5 of the present instruction word.

After the address of the next instruction word is properly stored in the word-selecting register, the computer may immediately carry out the dictates of the next instruction word read from memory, or may pause for a period of time before executing the next instruction. Whether or not the computer immediately proceeds to carry out the next instruction in the program depends upon the particular mode of operation pre-selected by the operator. If the computer is operating in an "automatic" mode, the next instruction in the program is immediately carried out upon completion of the preceding instruction. However, if the computer is operating in a "manual" mode, all computation stops upon completion of the execution of each instruction and is not resumed until either the manual or the automatic "mode-selection" push button is depressed. Upon depression of one of the mode-selection push buttons, the sequence of events is as follows: The next instruction word specified by the contents of the word-selection register is read out from memory and stored in the instruction register; the two decimal digits thus stored in section 1 of the instruction register are examined to determine the particular instruction to be carried out; and, thereafter, the computer proceeds to carry out the dictates of that particular instruction.

As previously mentioned, for the purpose of convenience in describing in detail the inherently complex mode of operation which the computer executes in carrying out a particular instruction, each of the eighteen different instructions is considered a "subprogram" comprising a plurality of "subinstructions" which are sequentially given to the computer, the sequence of events initiated by the computer in carrying out each of the subinstructions being under the control of the control-counter. Again, as previously mentioned, each of the just-mentioned subinstructions is classified in one of six general categories, labeled "word-cycles," "subcommands," "variable-time-delays," "decisions," "incrementing and decrementing," and "miscellaneous," of which the more complex ones of the various subinstructions have previously been described in detail, and all of which are each related to a specific portion of the computer control circuitry even though such specific portion does not always exist as a separate and distinct entity as such.

70. *Detailed Description of EKW Instructions*

The first subprogram to be described in detail is concerned with the execution of an EKW instruction which effects the storage in memory of amounts entered into the accounting machine keyboard or taken from a preselected one or ones of the totalizers therein. An EKW instruction may be considered as comprising four distinct phases or groups of subinstructions, during the first phase of which all computation and data-handling activity within the computer ceases until an amount is indexed into the keyboard of the accounting machine portion thereof, or until the machine carriage is tabulated to the next stop; however, if an "automatic-resume-program" has previously been specified, all activity ceases only until the carriage reaches a stop. The second phase of the subprogram is concerned with initiating a cycle of operation of the accounting machine; consequently, the high-order digit in section 2 of the instruction register is examined to determine the required cycle of operation of the accounting machine to be initiated. Thereafter, the required machine cycle is automatically initiated by proper actuation of the required motor bar when the resume-program-bar is subsequently depressed by the operator. Upon completion of the machine cycle, the differentially-set positions of the amount racks collectively represent the amount to be stored in memory, as previously described. The third phase of the subprogram is, therefore, concerned with examining the printed circuit switch on the rear of the machine to determine the amount just entered into the keyboard, and subsequent storage of the amount in address-A. If reverse key REV (FIGS. 2 and 44A) has previously been depressed, indicative of a negative amount being entered into the accounting machine keyboard, the word initially stored in address-A is complemented before being subsequently stored in the selected address in memory. However, if reverse key REV is not depressed, the word initially stored in address-A is complemented before being subsequently stored in the selected address in memory. However, if reverse key REV is not depressed, the word initially stored in address-A is subsequently transferred to the selected address in memory.

During the last phase of the subprogram, the contents of sections 3 and 4 of the instruction register are compared for equality to determine whether there are additional words to be entered into memory from the keyboard or totalizers of the accounting machine. If there is an additional entry to be made, section 3 of the instruction register is incremented, and the just-described sequence of events is again repeated. If, however, there are no additional entries to be made, the sequence of events initiated by the EKW instruction is thus terminated.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107A, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing an "EKW" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy the instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register. If the number thus stored in section 1 of the instruction register is "00," carry out the subinstruction listed in Step-2 of the following EKW subprogram:
2	PFϕ-0-3	Preset the F-counter to "0"; thereafter go to Step-3.
3	GO-0-4	Go to Step-4 in the subprogram whenever depression of a motor bar will initiate a machine cycle.
4	L3-5-6	If the low-order digit in section 2 of the instruction register is a "3," go to Step-6; otherwise, go to Step-5.
5	RPB-0-6	Upon depression of the resume-program-bar, go to Step-6.
6	Hϕ-8-7	If the high-order digit in section 2 of the instruction register is a "0," go to Step-7; otherwise, go to Step-8.
7	MBϕ(1)-0-18	Depress the upper motor bar for a "touch" operation; thereafter, go to Step-18.
8	H1-10-9	If the high-order digit of section 2 of the instruction register is a "1," go to Step-9; otherwise, go to Step-10.
9	MBϕ(2)-0-18	Depress the upper motor bar for a "hold" operation; thereafter, go to Step-18.
10	H2-12-11	If the high-order digit of section 2 of the instruction register is a "2," go to Step-11; otherwise, go to Step-12.
11	MB1(1)-0-18	Depress the middle motor bar for a "touch" operation; thereafter, go to Step-18.
12	H3-14-13	If the high-order digit of section 2 of the instruction register is a "3," go to Step-13; otherwise, go to Step-14.
13	MB1(2)-0-18	Depress the middle motor bar for a "hold" operation; thereafter, go to Step-18.
14	H4-16-15	If the high-order digit of section 2 of the instruction register is a "4," go to Step-15; otherwise, go to Step-16.
15	MB2(1)-0-18	Depress the lower motor bar for a "touch" operation; thereafter, go to Step-18.
16	H5-0-17	Go to Step-17 if the high-order digit of section 2 of the instruction register is a "5."
17	MB2(2)-0-18	Depress the lower motor bar for a "hold" operation; thereafter, go to Step-18.
18	L1-20-19	Go to Step-19 if the low-order digit of section 2 of the instruction register is a "1"; otherwise, go to Step-20.
19	OH1-0-22	When printing occurs, place a comma between rows #5 and #6; if there is no significant digit in rows #6 through #10, print a "0" preceding the comma; thereafter, go to Step-22.
20	L2-22-21	If the low-order digit in section 2 of the instruction register is a "2," go to Step-21; otherwise, go to Step-22.
21	OH2-0-22	When printing occurs, place a comma between #8 and #9; if there is no significant digit in rows #9 or #10, print a "0" preceding the comma; thereafter, go to Step-22.
22	PCT-0-23	When the amount racks are traveling in a "setting" direction, go to Step-23 each time the timing rack changes digital position and also upon engagement of the printing-liner.
23	AF-0-24	Increment the F-counter; thereafter, go to Step-24.
24	F9-22-25	Go to Step-25 if the F-counter is at a count of "9"; otherwise, go to Step-22.
25	CA(2)-0-26	Copy into address-A the word collectively indicated by the conditions of the rack readout switches; thereafter, go to Step-26.
26	REV-28-27	Go to Step-27 if the "reverse" key is depressed; otherwise, go to Step-28.
27	CPA-0-28	Complement the word stored in address-A; thereafter go to Step-28.
28	AM-3-29	Copy the word of address-A into the memory address specified by section 3 of the instruction register; thereafter, go to Step-29.
29	I34-30-31	Go to Step-31 if the memory addresses specified by sections 3 and 4 of the instruction register are equal; otherwise, go to Step-30.
30	IN4-0-2	Increment section 3 of the instruction register; thereafter, go to Step-2.
31	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Before proceeding with a detailed description of the mode of operation initiated by each of the eighteen basic instructions—i.e., "enter-keyboard-words" (EKW—00) through "enter-punched-tape" (EPT—17)—it is to be noted that, in an effort to avoid in the following description an undue multiplicity of back-and-forth references with respect to the various figures of the drawings, the particular portions of the computer circuitry utilized in carrying out a particular instruction have essentially been taken from FIGS. 52A through 85 and re-assembled into a composite circuit diagram relating to that particular instruction. For example, certain significant particular portions of the computer circuitry utilized in executing an EKW instruction are combined and logically illustrated in FIGS. 89A and 89B as a composite circuit diagram.

As previously described, the first subinstruction initiated and carried out in the execution of each instruction is a "MI" word-cycle, which effects the transfer to the instruction register of the word in memory previously stored at the address indicated by the contents of the word-selecting register. Also, as previously described, a "MI" word-cycle is initiated each time the state of line MI is rendered TRUE. As shown in FIG. 76, if the computer is operating in an "automatic" mode, as indicated by a TRUE state of line AUT, a "MI" word-cycle is automatically initiated by a reversal of state of flipflop **6092** when the state of line AN is rendered FALSE, line AN being rendered FALSE when the previously-initiated cycle of operation is completed. However, if the computer is operating in a "manual" mode, a "MI" word-cycle is initiated only by a depression of push-button MN1—MN2 (FIGS. 1 and 84), which also causes a reversal of state of flipflop **6092** (FIG. 76). After the next instruction word is transferred to the instruction register via the "MI" word-cycle, if the two decimal digits stored in section 1 thereof correspond to the code designation for an "enter-keyboard-words" instruction—i.e., "00"—the state of line EKW is rendered TRUE in a manner previously described with respect to FIG. 56, and, consequently, an "enter-keyboard-words" instruction is thereafter executed in the following manner.

It is, however, assumed that when the "enter-keyboard-words" instruction is first initiated: the able arm portion (**497**) of the accounting machine is in the non-deflected position shown in FIG. 38, and, consequently, the contacts of switch **496** are closed, thus indicating that motor bars **23**, **27**, and **28** are not disabled by able arm **497**; that the carriage of the accounting machine is not tabulating over a carriage stop, as indicated by a closure of the contacts of switch **510** (FIGS. 43 and 82); that the accounting machine is in home position, ready for a cycle of operation thereof to be initiated, as indicated by a closure of the contacts of switch **528** (FIGS. 41 and 82); and that a paper-tape-punching operation is not to be initiated, as indicated by a closure of contacts SC41—3 (FIG. 82) of switch SC41 (FIG. 13). The just-mentioned assumptions having been satisfied, the state of line GO (FIG. 82) is rendered TRUE, thus indicating that a cycle of operation of the accounting machine is now permitted to be initiated.

With reference now to FIG. 61, the state of line AN is reversed from TRUE to FALSE at TIME–1, indicating completion of the previously-initiated "MI" word-cycle, as previously described. With reference now to FIG. 72, when line AN is thus rendered FALSE, the state of lines MIN and PFϕ are likewise reversed from TRUE to FALSE. As previously described with reference to FIG. 72, when the state of line PFϕ is reversed from TRUE to FALSE, the F-counter is thereby preset to a count of "0." With reference to FIG. 89A, as a result of line MIN thus being rendered FALSE, the state of flipflop **6089** is reversed, so that line KEY is rendered TRUE thereby.

As all of the inputs to logical AND **1743** are simultaneously TRUE when line KEY is thus rendered TRUE, the state of line EKL is thereby rendered TRUE and thus effects illumination of the enter-keyboard-words lamp EK (see also FIG. 2), indicating that the computer is ready for an amount to be indexed into the keyboard of the accounting machine portion thereof. Also, prior to line EKL being rendered TRUE, the state of one of lines Lϕ, L1, or L2 is previously rendered TRUE, depending upon whether the low-order digit in section 2 of the instruction register (FIG. 57) is a "0," a "1," or a "2," respectively. Therefore, if the low-order digit is a "0," decimal-point lamp Pϕ (see also FIG. 2), located between amount rows #2 and #3 on the accounting machine keyboard, is illuminated when line EKL is rendered TRUE; if the low-order digit is a "1," decimal-point lamp P1 between amount rows #5 and #6 is illuminated; and, if the low-order digit is a "2," decimal-point lamp P2 between amount rows #8 and #9 is illuminated. As will later be seen, a decimal-point and enter-keyboard-words lamps both remain illuminated until the resume-program-bar RPB (FIG. 2) is depressed.

With reference to FIG. 38, when the resume-program-bar push button is depressed, the common plunger of switches RP1 and RP2 is actuated thereby. As shown in FIG. 84, when the resume-program-bar is depressed, the normally-opened switch contacts RP2 are closed, and normally-closed switch contacts RP1 are opened thereby. Consequently, at the following TIME–1 after the resume-program-bar is depressed, the state of flipflop **6138** is reversed, so that output line RP is thereby rendered TRUE. At TIME–1 following the release of the resume-program-bar, the state of flipflop **6138** is again reversed, so that line RP is rendered FALSE thereby. As shown in FIG. 78, as a result of line RP experiencing a TRUE-to-FALSE reversal of state, the state of flipflop **6099** is reversed, and line RPB is rendered TRUE thereby.

With reference back to FIG. 89A, at TIME–3 after line RPB is thus rendered TRUE, the reference input to flipflop **6118** reverses from TRUE to FALSE, and, consequently, the state of flipflop **6118** is reversed and line WOA is thereby rendered TRUE. It is to be noted that, if the low-order digit in section 2 of the instruction register (FIG. 57) is a "3," as indicated by line L3 being TRUE, indicating an automatic-resume-program operation, neither of the decimal-point lamps is illuminated, but, instead, the state of flipflop **6118** is reversed at TIME–3 even though the resume-program-bar was not manually depressed. However, after the state of line WOA is rendered TRUE, upper motor bar solenoid MBϕ is energized if the high-order digit in section 2 of the instruction register is either a "0" or a "1"(see also FIG. 38); middle motor bar solenoid MB1 is energized if the high-order digit in section 2 is either a "2" or a "3"; and lower motor bar solenoid MB2 is energized if the high-order digit is either a "4" or a "5." It is to be appreciated, however, that the selected motor bar solenoid is energized when line WOA is rendered TRUE, regardless of whether the high-order digit in section 2 of the instruction register is any one of the digits "0" through "5."

After the selected one of the upper, middle, or lower motor bars is automatically actuated, a cycle of operation of the accounting machine is thus initiated thereby. With reference to FIG. 72, when a machine cycle is initiated, the movable arm of switch **540** is deflected to the left, as viewed, so that line (NT)′ is rendered TRUE thereby, line NT, of course, being rendered FALSE. However, if for some reason a machine cycle is not initiated when a selected one of the motor bars is energized, line NT remains TRUE. With reference now to FIG. 81, when the state of flipflop **6118** is reversed, so that line (WOA)′ is rendered FALSE, the state of flipflop **6120** is reversed, so that line (RF)′ is rendered FALSE thereby. Approximately 250 milliseconds later, however, the state of flipflop **6120** reverts back to its initial condition, so that line (RF)′ is rendered TRUE thereby. Therefore, if it is assumed that a machine cycle was not initiated when line WOA was initially rendered TRUE, as evidenced by line NT remaining TRUE, the state of flipflop **6118** is reversed at the following TIME–**1** by line AN after line (RF)′ is rendered TRUE, thereby de-energizing the previously-energized motor bar solenoid. However, at the following TIME–**3** after the motor bar is physically restored to its normal undeflected position, so that line GO is again rendered TRUE, the state of flipflop **6118** is again reversed, so that line WOA is again rendered TRUE, thereby re-energizing the pre-selected motor bar solenoid. If a machine cycle is still not initiated when line WOA is rendered TRUE for the second time, the above sequence of events is continually repeated until a machine cycle is finally initiated.

As previously described with respect to FIGS. 7 and 51*g*, photocell **160** is utilized in conjunction with auxiliary timing rack **151** and light source **164** to provide an electrical impulse each time the auxiliary rack, and hence the amount racks, are translated from one digital position to another while traveling in a "setting" direction toward the rear of the machine. With additional reference to FIG. 72, each time a slot on timing rack **151** passes between light source **164** and photocell **160**, the state of the prime input to flipflop **6071** is reversed from TRUE to FALSE; consequently, the state of flipflop **6071** is reversed, so that line PT is rendered TRUE and line (PT)′ is simultaneously rendered FALSE. However, when the tooth portion of timing rack **151** passes between light source **164** and photocell **160**, the state of flipflop **6071** is again reversed, so that line PT is thereby rendered FALSE and line (PT)′ is simultaneously rendered TRUE. As output lines PT and (PT)′ of flipflop **6071** are logically connected as input lines to the reference and prime inputs, respectively, of flipflop **6072**, the state of flipflop **6072** is reversed, so that output line PCT is rendered TRUE at times corresponding to each digital position of the timing rack, and is rendered FALSE during each period of time the timing rack is being translated from one digital position to another.

With reference back to FIG. 89B, each time line PCT experiences a TRUE-to-FALSE reversal of state, line AF likewise experiences a TRUE-to-FALSE reversal of state. As a result, the F-counter (FIG. 72) is incremented by a count of "1" each time the timing rack is translated from one digital position to another. As shown in FIG. 89A, upper motor bar solenoid MBϕ, if initially energized, is de-energized when the F-counter is advanced from a count of "0" to a count of "1" unless the high-order digit in section 2 of the instruction register is a "1"; middle motor bar solenoid MB**1**, if initially energized, is de-energized when the F-counter is advanced from a count of "0" to a count of "1" unless the high-order digit in section 2 of the instruction register is a "3"; and lower motor bar solenoid MB**2**, if initially energized, is de-energized when the F-counter is advanced from a count of "0" to a count of "1" unless the high-order digit in section 2 of the instruction register is a "5."

It is evident, therefore, that either a touch operation or a hold operation of a particular motor bar is obtained simply by maintaining the corresponding motor bar solenoid energized for a predetermined period of time. However, when line WOA is subsequently rendered FALSE, any previously-energized motor bar solenoid is de-energized thereby. When the F-counter reaches a count of "8," as indicated by line F**8** being TRUE, order-hook solenoid OH**1** (see also FIG. 35) is energized if the low-order digit in section 2 of the instruction register is a "1," whereas order-hook solenoid OH**2** is energized if the low-order digit in section 2 is a "2." As previously described with respect to FIG. 35, when order-hook solenoid OH**1** is energized, all order-hooks from the third to and including the eighth one are unlatched from their respective type sectors. However, when order-hook solenoid OH**2** is energized, all order-hooks from the third to and including the eleventh one are unlatched from their respective type sectors. Also, as previously described, the eighth and the eleventh ones of type sectors **67** (FIG. 34) respectively print the sixth and ninth order degits of the amount indexed in the accounting machine keyboard, with each digit being followed by a comma.

When the timing rack leaves digital position #8, the state of F-counter line F**8** (FIG. 89A) is rendered FALSE and thereby causes a reversal of state of flipflop **6089**, which, in turn renders line KEY FALSE. Due to the fact that timing rack **151** is provided with only eight slots, as shown in FIG. 8, additional means are utilized to increment the F-counter from a count of "8" to a count of "9" when the timing rack has essentially reached digital position #9. As previously described with respect to FIGS. 3A and 41, when aligner bar **72** is brought into engagement with the notches formed on the lowermost end of arm **58** to precisely align all of the type sectors **67** in their differentially-set positions just prior to the printing operation, the contacts of switch **550** are closed as a result of clockwise rotational movement of follower arm **519**. Consequently, with reference back to FIG. 72, when the timing rack is translated to its #9 digital position, so that the printing-aligner bar is brought into engagement with the type sectors, a closure of the contacts of switch **550** causes a subsequent reversal of state of flipflop **6071**, which, in turn, causes the F-counter to be incremented from a count of "8" to a count of "9" in essentially the same manner as just described.

With reference back to FIG. 89A, when the F-counter is advanced from a count of "8," as indicated by line F**8** being rendered FALSE, any previously-energized order-hook solenoid is de-energized thereby. At TIME–**1** after the F-counter has been incremented to a count of "9," the state of flipflop **6118** is again reversed, so that line WOA is rendered FALSE thereby. When line WOA is thus rendered FALSE, any previously-energized motor bar solenoid is de-energized thereby. With reference now to FIG. 89B, also when line WOA is thus rendered FALSE, the state of flipflop **6080** is reversed, so that line CA is rendered TRUE thereby. As a result of line CA thus being rendered TRUE, a "CA" word-cycle is initiated and thereafter executed, whereby the word collectively indicated by the conditions of the rack-readout switches is read out and stored in address-A in the following manner, the word indicated by the rack-readout switches being identical to the mount just entered into the accounting machine keyboard, as previously described with respect to FIGS. 3B, 5A, and 5B.

As diagrammatically illustrated in FIG. 67B, digit-counter output lines Dϕ through D**9** are individually connected to a different one of conductors **119** disposed on printed circuit board **116**. Thus, suppose that the first-order amount rack is differentially set in digital position #5 corresponding to depression of amount key #5 in the low-order amount row of the machine keyboard; i.e., the first-order digit of the amount indexed in the keyboard is a "5." Consequently, when a machine cycle is subsequently initiated, so that the first-order amount rack is stopped in the #5 digital position, line Dϕ is electrically connected to line **120***e* by way of printed circuit conductors **118** and **119** due to the action of wiper blades **113–114**, as previously described with respect to FIGS. 3A, 5A, and 5B. Thus it is evident that, when the digit-counter (FIG. 63) is at a count of "0," so that line Dϕ is TRUE, and the bit-counter (FIG. 62) is simultaneously at a count of "*a*," so that line B*a* is true, the state of output line RSS is TRUE, indicating that the low-order bit of the first-order digit is a binary "1." When the bit-counter advances to a count of "*b*," line RSS is rendered FALSE, thus indicating that bit "*b*" is a binary "0"; when the bit-counter advances to a count of "*c*," line RSS is rendered TRUE, thus indicating that bit "*c*" is a binary "1"; and, when the bit-counter advances to a count of "*d*," line RSS is rendered FALSE, thus indicating that bit "*d*" of the first-order digit is a binary "0." Therefore, as bits "*a*" and "*c*" of the digit each have a value of binary "1," and as bits "*b*" and "*d*" each have a value of binary "0," the differentially-set position of the first-order amount rack corresponds to binary 0101—i.e., decimal "5"—which, in turn, corresponds to the assumed value of the first-order digit of the amount indexed in the accounting machine keyboard.

Therefore, it is evident that the bit and digit counters effectively operate together as a serializer which, in a sense, "scans" the differentially-set positions of the ten amount racks, starting with the first-order amount rack and ending with the tenth-order amount rack, and essentially generates a train of impulses via line RSS, which impulse train corresponds to the value of the amount entered into the accounting machine keyboard, the first impulse, or lack of an impulse, corresponding to the binary value of bit "*a*" of the first-order digit, and the last impulse, or lack of an impulse, corresponding to the binary value of bit "*d*" of the tenth-order digit of the amount.

With reference back to FIG. 89B, lines RSS, CA, and EKW are each connected as an input to logical AND **1137**. Consequently, as a result of line CA being rendered TRUE, a "CA(**2**)" word-cycle is executed, whereby the amount indexed in the accounting machine keyboard is sequentially stored in address-A via line MXW in essentially the same manner as previously described with respect to FIGS. 60, 52A, and 52B. After the word is stored in address-A, line AN is rendered FALSE, and, consequently, the state of flipflop **6080** is reversed, so that line CA is rendered FALSE, thus terminating the "CA" word-cycle operation.

As previously described with respect to FIGS. 2 and 44A, when reverse key "REV" is manually depressed, indicating that the amount just indexed in the keyboard of the accounting machine is a negative number, the movable arm of switch **572** is actuated thereby. With reference to FIG. 74, when the reverse key is depressed, the movable arm of switch **572** is deflected upwardly from the position shown, so that the state of line REV is rendered TRUE thereby. Therefore, as depression of the reverse key signifies that the algebraic sign of the word just stored in address-A is negative, and, as a negative word is stored in memory as its complement, the word just stored in address-A must be complemented before being stored in the selected address in memory. Thus, with reference to FIG. 89B, when line CA is thus rendered FALSE at the end of the previously-initiated "CA(**2**)" word-cycle, the state of flipflop **6083** is reversed, so that line CPA is rendered TRUE thereby. When line CPA is thus rendered TRUE, a "CPA" subcommand is thereafter executed, whereby the word in address-A is complemental and its complement is stored back in address-A in exactly the same manner as previously described.

When the previously-initiated "CPA" subcommand is terminated, so that line CPA is rendered FALSE, the state of flipflop **6077** is reversed, so that line AM is thereby rendered TRUE. When line AM is rendered TRUE, line TS3 is likewise rendered TRUE. Therefore, in response to lines AM and TS3 simultaneously being rendered TRUE, an "AM" word-cycle is thereafter executed, whereby the word stored in address-A is stored in the selected address in memory as specified by the contents of section 3 of the instruction register. It is to be noted that, had the reverse key not been depressed, the word in address-A is immediately stored in memory without being complemented when line CA is rendered FALSE. This is due to the fact that the state of flipflop **6077** is reversed when line CA is rendered FALSE, because line (REV)′ is TRUE, in this instance, indicative of non-depression of the reverse key.

After the word in address-A is stored in the selected address in memory specified by section 3 of the instruction register, the state of flipflop **6077** is reversed when line AN is rendered FALSE, indicating that the previously-initiated "AM" word-cycle is completed; consequently, the states of lines AM and TS3 are both rendered FALSE. As previously described with respect to FIG. 59, if the contents of sections 3 and 4 of the instruction register are of equal magnitude, indicating that there are no other words to be indexed into the accounting machine keyboard, line **I34** is rendered FALSE, and line (**I34**)′ is simultaneously rendered TRUE. However, if sections 3 and 4 are of unequal magnitude, indicating that there are additional words to be indexed into the keyboard, line **I34** is rendered TRUE, and line (**I34**)′ is simultaneously rendered FALSE.

Thus, with reference back to FIG. 89B, if there are no other words to be indexed into the keyboard, the state of flipflop **6107** is reversed when line AM is rendered FALSE and line STD is rendered TRUE thereby. When line STD is thus rendered TRUE, line **TS5** is likewise rendered TRUE. In response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register.

However, if there are additional words to be indexed into the accounting machine keyboard, line **IN4** is rendered FALSE when line AN is rendered FALSE, and, as a result, section 3 of the instruction register (FIG. 58) is incremented by a count of "1" in the same manner as previously described. If, before being incremented, the contents of section 3 of the instruction register is not equal to the contents of section 4 thereof, the state of flipflop **6089** (FIG. 89A) is reversed when line AM is rendered FALSE at the end of the previously-initiated "AM" word-cylcle. As a result, line KEY is again rendered TRUE, and the F-counter (FIG. 72) is again preset to "0" via line PFϕ. Prior to a subsequently-initiated machine cycle when line GO is rendered TRUE, enter-keyboard-words lamp EK is illuminated, thus providing a visual indication that another amount is to be indexed in the keyboard; thereafter, the just-described sequence of events is again repeated.

It is to be noted that, even though line STD is rendered TRUE upon termination of the EKW subprogram, section 3 of the instruction register is incremented via line **IN4** when line AM is rendered FALSE, regardless of the result of the comparison between the contents of sections 3 and 4 of the instruction register. It is also to be noted that, if it is desired that a plurality of words be stored in sequentially-numbered addresses in memory during the execution of a single EKW subprogram, the same motor bar, decimal-point lamp, and order-hook solenoid are utilized for each word entered. However, if a different motor bar, decimal-point lamp, or order-hook solenoid is desired to be utilized, a second EKW subprogram is initiated by a second EKW instruction word which specifies the desired combination thereof.

71. *Detailed Description of POW Instructions*

The next subprogram to be described in detail is concerned with the execution of a single POW instruction which effects the sequential printing-out of from one to one hundred words which were previously stored in sequentially numbered addresses in memory. Like the just-described EKW instruction, a POW instruction may also be considered as comprising four distinct phases or groups of subinstructions.

During the first phase, the computer, in a sense, ascertains whether or not depression of a motor bar is capable of initiating a cycle of operation of the accounting machine. As previously described, prior to the machine cycle, each of the ten amount racks is physically located at "home" position, which position corresponds to a digital value of "0." Thus, the selected word in memory is read out and examined digit by digit, starting with the low-order digit thereof, to determine those digits of the word having a value of "0." Simultaneously therewith, the rack-stopping solenoids corresponding to the digital positions of those digits of the word having a value of "0" are energized to prevent any movement of the associated amount racks when a machine cycle is subsequently initiated. Thereafter, the state of a particular sign memory flipflop is selectively conditioned to correspond to the algebraic sign of the word to be printed out. If the algebraic sign of the word to be printed out is negative, credit balance key CR. BAL. (FIG. 2) is effectively depressed via selective energization of credit balance solenoid CBS, as previously described with respect to FIG. 37 and also as shown in FIG. 81.

When a machine cycle is subsequently initiated in order to carry out the first half of a credit balance operation, the word read out from memory is subtracted from the contents of the "X" totalizer, which normally stores a word of all zeros. When the second machine cycle is subsequently initiated in order to carry out the final half of a credit balance operation, a total operation is taken with respect to the "X" totalizer, so that the complement of the word just read out from memory is printed in red, with a "CR" symbol to the right thereof.

Due to the fact that the differentially-set positions of the ten amount racks are determined solely by the computer during the execution of a POW instruction, rather than by a totalizer or by depressed keys on the accounting machine keyboard, it is therefore necessary to automatically release the amount racks and all previously depressed amount keys when a total operation is called for during the execution of a POW instruction. Thus, in order to permit the automatic release of the amount racks and the keys prior to the initiation of a total-taking operation, rack release solenoid RRS (FIGS. 39 and 81) is selectively energized.

During the second phase of the subprogram, the particular motor bar operation specified by the value of the low-order digit in section 2 of the instruction is executed in order to initiate the required machine cycle.

The third phase of the POW instruction begins when the machine starts to cycle, so that the timing rack and all non-stopped amount racks simultaneously begin their travel in a "setting" direction toward the rear of the machine, those amount racks corresponding to the digital positions of the digits of the word having a value of "0" remaining in home position. Each time the timing rack is translated from a position indicative of one digital value to a position indicative of the next higher successively higher-order digital value after leaving home position, each of the digits of the word is essentially "examined" to determine whether its digital value corresponds to the digital value represented by the instantaneous position of the amount racks. If such correspondence exists, all corresponding digital-order rack-stopping solenoids are simultaneously energized and thereby arrest further movement of all corersponding amount racks. After all of the amount racks have been differentially positioned to correspond to the word just read out from memory, a normal printing operation takes place.

During the final phase of a POW instruction, the contents of sections 3 and 4 of the instruction register are compared for equality to determine whether or not there are additional words to be printed out from memory. If there are additional words to be printed out, section 3 of the instruction register is incremented, and the entire sequence of events, just described, is again repeated. Otherwise, the instruction is concluded by effecting the copying into the word-selecting register the address of the next instruction word, which is located in section 5 of the instruction register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107B, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "POW" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register thereafter, if the number thus stored in section 1 of the instruction register is "01," carry out the subinstruction listed in Step-2 of the following POW subprogram.
2	PFφ-0-3	Preset the F-counter to "0"; thereafter, go to Step-3.
3	GO-0-4-T	Go to Step-4 in the subprogram whenever depression of a motor bar will initiate a machine cycle.
4	AD-0-5	Increment the digit-counter; thereafter, go to Step-5.
5	MJ-3-6	Copy into the "J" digit-register the digit corresponding to the count in the digit counter of the word stored in memory at the address indicated by section 3 of the instruction register; thereafter, go to Step-6.
6	IFJ-27-7	If the digit in the F-counter is equal to the digit in the "J" digit-register, go to Step-7; otherwise, go to Step-27.
7	Dφ-9-8	Go to Step-8 if the digit-counter is at a count of "0"; otherwise, go to Step-9.
8	RAφ-0-27	Energize the first-order rack-stopping solenoid; thereafter, go to Step-27.
9	D1-11-10	Go to Step-10 if the digit-counter is at a count of "1" otherwise, go to Step-11.
10	RA1-0-27	Energize the second-order rack-stopping solenoid; thereafter, go to Step-27.
11	D2-13-12	Go to Step-12 if the digit-counter is at a count of "2"; otherwise, go to Step-13.
12	RA2-0-27	Energize the third-order rack-stopping solenoid; thereafter, go to Step-27.
13	D3-15-14	If the digit-counter is at a count of "3," go to Step-14; otherwise, go to Step-15.
14	RA3-0-27	Energize the fourth-order rack-stopping solenoid; thereafter, go to Step-27.
15	D4-17-16	If the digit-counter is at a count of "4," go to Step-16; otherwise, go to Step-17.
16	RA4-0-27	Energize the fifth-order rack-stopping solenoid; therafter, go to Step-27.
17	D5-19-18	If the digit-counter is at a count of "5," go to Step-18; otherwise, go to Step-19.
18	RA5-0-27	Energize the sixth-order rack-stopping solenoid; thereafter, to go Step-27.
19	D6-21-20	If the digit-counter is at a count of "6," go to Step-20; otherwise, go to Step-21.
20	RA6-0-27	Energize the seventh-order rack-stopping solenoid; thereafter, go to Step-27.
21	D7-23-22	If the digit-counter is at a count of "7," go to Step-22; otherwise, go to Step-23.
22	RA7-0-27	Energize the eighth-order rack-stopping solenoid; thereafter, go to Step-27.
23	D8-25-24	If the digit-counter is at a count of "8," go to Step-24; otherwise, go to Step-25.
24	RA8-0-27	Energize the ninth-order rack-stopping solenoid; thereafter, go to Step-27.
25	D9-0-26	If the digit-counter is at a count of "9," go to Step-26; otherwise go to Step-27.
26	RA9-0-27	Energize the tenth-order rack-stopping solenoid; thereafter, go to Step-27.
27	D9-4-28	If the digit-counter is at a count of "9," to go Step-28; otherwise, go to Step-4.
28	OBM(4)-0-29	Condition the state of line OBM to correspond to the sign of the word just read out from memory; thereafter, go to Step-29.
29	OBM(2)-30-31	If the state of line OBM is TRUE, go to Step-31; otherwise, go to Step-30.
30	CBS-0-31	Energize the credit balance solenoid; thereafter, go to Step-31.
31	RRS-0-32	Energize the rack-release solenoid; thereafter, go to Step-32.
32	Hφ-34-33	If the high-order digit in section 2 of the instruction register is "0," go to Step-33; otherwise, go to Step-34.
33	MBφ(1)-0-44	Depress the upper motor bar for a touch operation; thereafter, go to Step-44.
34	H1-36-35	If the high-order digit in section 2 of the instruction register is a "1," go to Step-35; otherwise, go to Step-36.
35	MBφ(2)-0-44	Depress the upper motor bar for a hold operation; thereafter, go to Step-44.
36	H2-38-37	If the high-order digit in section 2 of the instruction register is a "2," go to Step-37; otherwise, go to Step-38.
37	MB1(1)-0-44	Depress the middle motor bar for a touch operation; thereafter, go to Step-44.
38	H3-40-39	If the high order digit in section 2 of the instruction register is a "3," go to Step-39; otherwise, go to Step-40.
39	MB1(2)-0-44	Depress the middle motor bar for a hold operation; thereafter, go to Step-44.
40	H4-42-41	If the high-order digit in section 2 of the instruction register is a "4," go to Step-41; otherwise go to Step-42.
41	MB2(1)-0-44	Depress the lower motor bar for a touch operation; thereafter, go to Step-44.
42	H5-0-43	If the high-order digit in section 2 of the instruction register is a "5," go to Step-43.
43	MB2(2)-0-44	Depress the lower motor bar for a hold operation; thereafter, go to Step-44.

Step	Subinstructions	Description
44	L1-46-45	If the low-order digit in section 2 of the instruction register is a "1," go to Step-45; otherwise, go to Step-46.
45	OH1-0-48	When printing occurs, place a comma between rows #5 and #6; if there is no significant digit in rows #6 through #10, print a zero preceding the comma; thereafter, go to Step-48.
46	L2-48-47	If the low-order digit in section 2 of the instruction register is a "2," go to Step-47; otherwise, go to Step-48.
47	OH2-0-48	When printing occurs, place a comma between rows #8 and #9; if there is no significant digit in rows #9, or #10, print a zero preceding the comma; thereafter, go to Step-48.
48	PCT-0-49	When the amount racks are traveling in a "setting" direction, go to Step-49 each time the timing rack changes digital position, and also upon engagement of the printing-liner.
49	AF-0-50	Increment the F-counter; thereafter, go to Step-50.
50	F9-51-75	If the F-counter is at a count of "9," go to Step-75; otherwise, go to Step-51.
51	AD(1)-0-52	Increment the digit-counter; thereafter, go to Step-52.
52	MJ-3-53	Copy into the "J" digit-register the digit corresponding to the count in the digit counter of the word stored in memory at the address specified by section 3 of the instruction register; thereafter, go to Step-53.
53	IFJ-74-54	If the digit in the F-counter is equal to the digit in the "J" digit-register, go to Step-54; otherwise, go to to Step-74.
54	Dϕ-56-55	If the digit-counter is at a count of "0," go to Step-55; otherwise, go to Step-56.
55	RAϕ-0-74	Energize the first-order rack-stopping solenoid; thereafter, go to Step-74.
56	D1-58-57	If the digit-counter is at a count of "1," go to Step-57; otherwise, go to Step-58.
57	RA1-0-74	Energize the second-order rack-stopping solenoid; thereafter, go to Step-74.
58	D2-60-59	If the digit-counter is at a count of "2," go to Step-59; otherwise, go to Step-60.
59	RA2-0-74	Energize the third-order rack-stopping solenoid; thereafter, go to Step-74.
60	D3-62-61	If the digit-counter is at a count of "3," go to Step-61; otherwise, go to Step-62.
61	RA3-0-74	Energize the fourth-order rack-stopping solenoid; thereafter, go to Step-74.
62	D4-64-63	If the digit-counter is at a count of "4," go to Step-63; otherwise, go to Step-64.
63	RA4-0-74	Energize the fifth-order rack-stopping solenoid; thereafter, go to Step-74.
64	D5-66-65	If the digit-counter is at a count of "5," go to Step-65; otherwise, go to Step-66.
65	RA5-0-74	Energize the sixth-order rack-stopping solenoid; thereafter, go to Step-74.
66	D6-68-67	If the digit-counter is at a count of "6," go to Step-67; otherwise, go to Step-68.
67	RA6-0-74	Energize the seventh-order rack stopping solenoid; thereafter, go to Step-74.
68	D7-70-69	If the digit-counter is at a count of "7," go to Step-69; otherwise, go to Step-70.
69	RA7-0-74	Energize the eighth-order rack-stopping solenoid; thereafter, go to Step-74.
70	D8-72-71	If the digit-counter is at a count of "8," go to Step-71; otherwise, go to Step-72.
71	RA8-0-74	Energize the ninth-order rack-stopping solenoid; thereafter, go to Step-74.
72	D9-0-73	If the digit-counter is at a count of "9," go to Step-73; otherwise, go to Step-74.
73	RA9-0-74	Energize the tenth-order rack-stopping solenoid; thereafter, go to Step-74.
74	D9-51-48	If the digit-counter is at a count of "9," go to Step-48; otherwise, go to Step-51.
75	CA(2)-0-76	Copy into address-A the word collectively indicated by the condition of the rack readout switches; thereafter, go to Step-76.
76	RSB-3-77	Subtract the word located in memory at the address specified by section 3 of the instruction register from the word in address-A and store the remainder in address-A; thereafter, go to Step-77.
77	OBN(1)-80-78	If the state of line OBN is TRUE, go to Step-78; otherwise, go to Step-80.
78	RPB-0-79	Upon depression of the resume-program bar, go to Step-79.
79	OBN(3)-0-2	Preset line OBN to a FALSE state; thereafter, go to Step-2.
80	I34-81-83	If the addresses in memory specified by sections 3 and 4 of the instruction register are equal, go to Step-83; otherwise, go to Step-81.
81	IN4-0-82	Increment section 3 of the instruction register; thereafter, go to Step-82.
82	OBM(3)-0-2	Preset line OBM to a FALSE state; thereafter, go to Step-2.
83	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from memory and stored in the instruction register, if the two-decimal-digit number stored in section 1 thereof corresponds to the code des-

ignation for a POW instruction—i.e., "01"—the state of line POW is rendered TRUE in a manner previously described with respect to FIG. 56, and, consequently, a "print-out-words" instruction is thereafter executed in the following manner:

With reference to FIGS. 90A through 90C, there is logically illustrated therein a composite circuit diagram of particular portions of the computer circuitry utilized in executing a POW instruction. As shown in FIG. 90A, at TIME-1 after the initiation of a "MI" word-cycle, the states of flipflops **6095** and **6096** are simultaneously reversed, if not already in reversal states, so that lines (OBM)′ and (OBN)′ are respectively rendered TRUE thereby. As shown in FIG. 90B, the state of line PFϕ is rendered FALSE upon completion of the previously-initiated "MI" word-cycle when the state of line AN is rendered FALSE; consequently, the F-counter (FIG. 72) is preset to "0."

It is assumed that, upon initiation of the POW instruction, able arm **497** (FIG. 38) of the accounting machine is in the non-deflected position, as shown, and, consequently, the contacts of switch **496** (FIGS. 38 and 82) are closed, thus indicating that the motor bars are not disabled; that the carriage of the accounting machine is not tabulating over a carriage stop as indicated by a closure of the contacts of switch **510** (FIGS. 43 and 82); that the machine is in home position, so that the contacts of switch **528** (FIGS. 41 and 82) are closed; and that a paper-tape-punching operation is not to be initiated as indicated by a closure of contacts SC**41**—**3** (FIG. 82) of switch SC**41** (FIG. 13). The just-mentioned conditions having been satisfied, the state of line GO (FIG. 82) is rendered TRUE, indicating that a cycle of operation of the accounting machine is permited to be initiated.

Thus, with reference back to FIG. 90A, all of the inputs to logical AND **1408** are simultaneously TRUE just prior to the completion of the previously-initiated "MI" word-cycle. Consequently, when line AN is rendered FALSE upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6066** is reversed, so that line MJ is thereby rendered TRUE. With reference to FIG. 90B, line TS3 is rendered TRUE at the beginning of the instruction when line POW is rendered TRUE. Consequently, in response to lines MJ and TS3 being rendered TRUE, a "MJ" word-cycle is thereafter executed, whereby the word stored in memory at the address specified by the contents of section 3 of the instruction register is read out digit by digit, starting with the low-order digit thereof, and stored digit by digit in the "J" digit-register (FIG. 71). At TIME-3 after the initiation of a "MJ" word-cycle, the state of flipflop **6118** (FIG. 90A) is reversed, so that line WOA is thereby rendered TRUE.

After each of the digits of the word is read out from memory via a "MJ" word-cycle and sequentially stored in the "J" digit-register, the contents of the F-counter and the "J" digit-register are compared for equality in a manner previously described with respect to FIG. 72. However, due to the fact that the F-counter is initially preset to "0," each digit stored in the "J" digit-register is essentially examined in order to determine whether its value is "0." If the digit stored in the "J" digit-register has a value of "0" when the "compare for equality" is made with respect to the contents of the F-counter and the "J" digit-register, the state of line (IFJ)′ is rendered TRUE; otherwise, the state of line (IFJ)′ remains FALSE.

With reference now to FIG. 90C, after the state of line WOA is rendered TRUE at the beginning of the "MJ" word-cycle, line EVG experiences a TRUE-to-FALSE reversal of state each time the "J" digit-register is preset to "0" via a TRUE-to-FALSE reversal of state of line PJϕ if the digit presently stored in the "J" digit-register has a value of "0," as indicated by a TRUE state of line (IFJ)'. Consequently, when the digit-counter is at a count of "1," as indicated by a TRUE state of line O1, the state of flipflop **6108** is reversed, so that line (RAϕ)' is rendered FALSE if the value of the first-order digit of the word just read from memory is zero. When the digit-counter is subsequently incremented to a count of "2," line (RA1)' is rendered FALSE by a reversal of state of flipflop **6109** if the value of the second-order digit of the word is also zero, and so on, until line (RA8)' is rendered FALSE by a reversal of state of flipflop **6116** if the value of the ninth-order digit of the word just read out from memory is zero. If the value of the tenth-order digit is zero, as indicated by a TRUE state of line (IFJ)', the state of flipflop **6117** is reversed, and line (RA9)' is rendered FALSE when line AN is rendered FALSE, indicating completion of the "MJ" word-cycle.

It is seen, therefore, that the states of selected ones of lines (RAϕ)' through (RA9)', corresponding to the digital positions of those digits of the word having a value of zero, are successively rendered FALSE by corresponding reversal of states of flipflops **6108** through **6117**, respectively. Thus, after the digit-counter reaches a count of "9," as indicated by a FALSE state of line (D9)', the outputs from corresponding ones of logical ORS **3497** through **3506** are rendered FALSE when line PCT is subsequently rendered FALSE. Consequently, those ones of rack-stopping solenoids RAϕL through RA9L corresponding to those digital positions of the digits of the word having a value of zero are simultaneously energized and thereafter prevent, in a manner previously described, any movement of the associated amount racks when a machine cycle is subsequently initiated. In other words, after the first "MJ" word-cycle is terminated, all amount racks corresponding to the digital positions of those digits of the word having a value of zero are locked in home position prior to the initiation of a machine cycle.

As previously stated with respect to FIG. 3A, in initiating an addition or a subtraction cycle of operation of the accounting machine, depression of the amount keys causes zero stop pawls **46** to be rocked counter-clockwise and thereby release amount racks **49** for rearward movement when a machine cycle is subsequently initiated. In a total-taking operation, however, no amount keys are depressed. Thus, additional means are utilized in a manner previously described with respect to FIGS. 39 and 40, whereby the zero stop pawls are rocked counter-clockwise, and, additionally, all accidentally-depressed amount keys are released at the beginning of the machine cycle. Consequently, with reference to FIG. 90B, rack-release solenoid RRS is energized when the state of line WOA is rendered TRUE, and thereby allows the just-mentioned additional means to function in the manner previously described.

With reference to FIG. 90A, if the high-order digit of the word just read out from memory and stored in the "J" digit-register is a "9," indicating the complement of a negative number, the state of line (J9)' is rendered FALSE. Thus, at TIME-3 during the last bit time period of the "MJ" word-cycle, when the state of line EG is reversed from TRUE to FALSE, the state of flip flop **6095** remains unchanged, so that line (OBM)' remains TRUE. However, if the algebraic sign of the word just read out from memory is positive, so that line (J9)' is TRUE, the state of line (OBM)' is rendered FALSE at TIME-3 by a change of state of flipflop **6095**. Consequently, line (OBM)', in the later instance, does not remain TRUE for a time period sufficient to allow credit balance solenoid CBS (FIG. 90B) to be responsive thereto. However, if the algebraic sign of the word just read out from memory is negative, so that line (OBM)', remains TRUE, credit balance solenoid CBS is energized for a time period sufficient for normal operation thereof.

As previously described with respect to FIG. 37, as a result of credit balance solenoid CBS being energized, the amount collectively indicated by the differentially-set positions of the amount racks is subtracted, during the first machine cycle, from the "X" totalizer, which normally stores a word of all zeros. During the second machine cycle, a total operation is executed with respect to the "X" totalizer, so that the complement of the word just read out from memory is printed in red, with a "CR" symbol disposed to the right thereof.

As shown in FIG. 90B, when line WOA is rendered TRUE at the beginning of the "MJ" word-cycle, upper motor bar solenoid MBϕ is energized if the high-order digit in section 2 of the instruction register is a "0" or a "1"; middle motor bar solenoid MB1 is energized if the high-order digit in section 2 is either a "2" or a "3"; and lower motor bar solenoid MB2 is energized if the high-order digit is either a "4" or a "5". However, approximately 130 milliseconds after the preselected motor bar is automatically depressed via energization of the corresponding motor bar solenoid, all amount racks not locked in home position are simultaneously translated in a setting direction toward the rear of the machine.

As previously described with respect to FIG. 72 and with respect to the mode of operation of an EKW subprogram, the state of output line PCT is rendered TRUE at each translational position of the timing rack corresponding to a digital value and is rendered FALSE during the time the timing rack is being translated from one position to a successively higher-order digital value position. Therefore, with reference back to FIG. 90A, when the timing and amount racks first begin their translational movement toward the rear of the machine, the state of line (PCT)' is reversed from TRUE to FALSE, and, as a result, the state of flipflop **6088** is reversed, so that line JHR is thereby rendered TRUE. At the following TIME-1, the state of flip-flop **6088** is again reversed, and line JHR is thereby rendered FALSE. As shown in FIG. 90B, each time line JHR experiences a TRUE-to-FALSE reversal of state, the state line AF likewise experiences a TRUE-to-FALSE reversal of state. Thus, the F-counter (FIG. 72) is incremented by a count of "1" each time the racks are simultaneously translated from one position to a successively higher-order digital valued position.

With reference to FIG. 90A, each time the state of line JHR reverses from TRUE to FALSE, the state of flipflop **6066** is reversed, so that line MJ is thereby rendered TRUE; consequently, a "MJ" word-cycle is thereby initiated at the following TIME-1 after each time line PCT is rendered TRUE while the racks are traveling in a setting direction. As just described, during the execution of a "MJ" word-cycle, all of the digits of the word stored in memory at the address specified by the contents of section 3 of the instruction register are sequentially read out, stored in the "J" digit-register, and compared with the digital count of the F-counter.

Thus, during the time the amount racks are being simultaneously translated from home position to their #1 digital positions, the state of line JHR is reversed from TRUE to FALSE, the F-counter is advanced to a count of "1," and simultaneously therewith, a "MJ" word-cycle is initiated. Consequently, each digit of the word read out from memory is effectively examined to determine whether its value is "1." Thereafter, those ones of rack-stopping solenoids RAϕL through RA9L (FIG. 4) corresponding to the digital positions of those digits of the word having a value of "1" are simultaneously energized to arrest movement of and thus differentially position the corresponding amount racks at the #1 digital valued positions thereof in the same manner as just described.

The above sequence of events is sequentially repeated until all of the ten amount racks are differentially set at those digital valued positions collectively representative of the word just read from memory. It is to be noted, however, that a "MJ" word-cycle is also initiated when the F-counter reaches a count of "9" even though all amount racks not previously stopped are automatically stopped at their high-order digital valued positions. However, with reference to FIG. 90B, when the F-counter reaches a count of "8," as indicated by a TRUE state of line F8, order-hook solenoid OH1 is energized if the low-order digit in section 2 of the instruction register is a "1," whereas order-hook solenoid OH2 is energized if the low-order digit in section 2 is a "2." As previously described with respect to the EKW subprogram, when order-hook solenoid OH1 is energized, all order-hooks from the third to and including the eighth ones are unlatched from their respective type sectors. However, when order-hook solenoid OH2 is energized, all order-hooks from the third to and including the eleventh ones are unlatched from their respective type sectors. As previously described with respect to FIGS. 3A, 41, and 72, when printing aligner bars **72** engage the proper tooth of corresponding ones of the members **58** in order to precisely align each of the type sectors **67** prior to a printing operation, the contacts of switch **550** are closed, so that the state of flipflops **6071** and **6072** are reversed, so that line (PCT)′ is rendered FALSE thereby.

As a result of line (PCT)′ thus being rendered FALSE, the state of flipflop **6088** (FIG. 90A) is again reversed, so that line JHR is again rendered TRUE. Thus, at the following TIME–1 when line JHR is rendered FALSE, line AF (FIG. 90B) experiences a TRUE-to-FALSE reversal of state to effect the advancement of the F-counter from a count of "8" to a count of "9." With reference to FIG. 90B, when the F-counter is incremented from a count of "8" to a count of "9," any previously-energized order-hook solenoid (OH1—OH2) is de-energized. With reference back to FIG. 90A, when line AN is rendered FALSE after the F-counter reaches a count of "9," the state of flipflop **6118** is reversed, so that line WOA is thereby rendered FALSE. With reference now to FIGS. 90B and 90C, as a result of line WOA thus being rendered FALSE, all previously-energized motor bar, rack-release, credit-balance, and rack-stopping solenoids are de-energized and thus rendered ineffective.

With reference to FIG. 90A, when the F-counter reaches a count of "9," as indicated by line F9 being rendered TRUE, the state of flipflop **6080** is reversed, and line CA is thereby rendered TRUE upon completion of the following "MJ" word-cycle, as indicated by line MJ being rendered FALSE. In response to line CA thus being rendered TRUE, a "CA" word-cycle is thereafter executed, whereby the word collectively indicated by the differentially-set positions of the amount racks is read out, in the manner previously described with respect to the EKW subprogram, and thereafter stored in address-A via line MXW (FIG. 90B). Upon completion of the previously-initiated "CA" word-cycle, as indicated by line CA being rendered FALSE, the state of flipflop **6100** (FIG. 90A) is reversed, so that line RSB is thereby rendered TRUE. In response to line RSB thus being rendered TRUE, a "RSB" subcommand is thereafter executed, whereby the word initially read out from the address in memory as specified by the contents of section 3 of the instruction register is subtracted digit by digit from the word just stored in address-A. If each digit of the word just read out from memory is equal in magnitude to the corresponding digit of the word stored in address-A, line ZNO remains FALSE, and, as a result, the state of flipflop **6096** remains unchanged, so that line (OBN)′ remains TRUE. However, if a digit of the word in address-A is not equal to the corresponding digit of the corresponding word in memory, indicating that the word just printed out does not correspond to the word just read out from memory, line ZNO experiences a TRUE-to-FALSE reversal of state, thereby causing the state of flipflop **6096** to be reversed, so that line (OBN)′ is thereby rendered FALSE, thus indicating that a printing error has occurred. When the state of line (OBN)′ is thus reversed from TRUE to FALSE, the state of flipflop **6107** is not reversed by logical AND **1716** to indicate completion of the "POW" subprogram, as is normally the case.

As previously described with respect to FIG. 59, if the contents of section 3 of the instruction register is equal to the contents of section 4 thereof, indicating that there are no other words to be printed out from memory, the state of line (I34)′ is rendered TRUE. Line (I34)′, however, is rendered FALSE if there are additional words to be printed out.

Therefore, with reference back to FIG. 90A, if the word just printed out is equal in magnitude to the word just read out from memory, as indicated by a TRUE state of line (OBN)′, and if there are no other words to be printed out, as indicated by a TRUE state of line (I34)′, the state of flipflop **6107** is reversed by the output of logical AND **1716** when line AN experiences a TRUE-to-FALSE reversal of state upon completion of the previously-initiated "RSB" subcommand; consequently, line STD is thereby rendered TRUE, thus indicating the end of the POW subprogram. When line STD is thus rendered TRUE, line TS5 (FIG. 90B) is likewise rendered TRUE. Thus, as previously described with respect to the EKW subprogram, in response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register.

However, if there are additional words to be printed out, line IN4 (FIG. 90B) is rendered FALSE when line AN is rendered FALSE upon completion of the previously initiated "RSG" subcommand. Consequently, section 3 of the instruction register is incremented, and the same "POW" subprogram is again subsequently executed in the same manner as just described. It is to be noted, however, that, even though line STD is rendered TRUE, indicating the end of the "POW" subprogram, section 3 of the instruction register is, nevertheless, incremented via line **IN4** regardless of the result of the comparison between the contents of sections 3 and 4 of the instruction register. It is also to be noted that, if a plurality of words stored in sequentially-numbered addresses in memory are to be sequentially printed out during the execution of a single "POW" subprogram, the same motor bar and order-hook solenoids are utilized for each word printed out. However, if a different motor bar or order hook is desired to be utilized, a second "POW" instruction is to be initiated, which specifies the desired combination thereof.

72. Detailed Description of ROC Instruction

Before describing the mode of operation executed by the computer in carrying out a "record-on-card" subprogram, a description will first be given with respect to the particular ledger card which is utilized in a novel manner by the present computer in accordance with the present invention.

As previously described with respect to FIG. 88A, the ledger cards utilized by the present computer are capable of having recorded thereon historical, current, and fixed data in both machine-readable and human-readable languages and, consequently, are readily adaptable to be utilized in the mechanization of record-keeping and accounting systems and procedures. As illustrated in FIG. 88A, each ledger card is provided with a vertical strip **385** of ferromagnetic material approximately one-half-thousandth inch thick and approximately one inch wide, which is coated along substantially the entire left margin of its face surface. As further illustrated in FIG. 88B, the magnetic strip is capable of being effectively divided into three magnetically isolated vertical channels, each of which is adapted to have binary information magnetically recorded therein. As previously described with respect to the ledger card handling equipment, and as diagrammatically illustrated in FIG. 88B, the leftmost channel is adapted to have "linefind" information magnetically recorded therein, the centrally-disposed channel is adapted to have synchronizing "clock" information magnetically recorded therein, and the rightmost channel is adapted to have data information magnetically recorded therein, whereby each reproduced linefind information is in the form of either a signal impulse or a pair of oppositely-going impulses, the reproduced position of which is indicative of the position of the next posting line, and whereby the reproduced clock information is in the form of a train of equally-spaced impulses of alternating polarity and disposed with reference to adjacent ones of which impulses is located a binary data bit in the data channel.

As will be seen later, due to the fact that the ledger card is translated at a linear speed of approximately ten inches per second with respect to the recording and reproducing portion of the card-handling equipment, and, as the recording clock frequency is approximately two hundred cycles per second (square wave), the recording density is therefore approximately forty binary bits per inch. As four binary bits are utilized by the present computer to represent a single decimal digit, the digit density is equal to substantially ten decimal digits per inch. Thus, a ledger card having a length of approximately twelve and three quarters inches is capable of having magnetically recorded on the ferromagnetic strip thereof approximately one hundred decimal digits and, additionally, provides sufficient space to have printed on the face surface thereof approximately fifty-seven typewritten lines at six lines per inch.

Before a previously unused ledger card is fed into the computer, it is desirable that each of the three magnetic channels be magnetically polarized in one direction only, in order to provide a proper premagnetized background for subsequently-recorded data thereon. After each posting operation is completed, and before the ledger card is ejected from the computer, a linefind discontinuity or indication is recorded on the magnetic strip at a vertical position indicative of the position of the next posting line. The linefind indication is obtained simply by reversng the direction of magnetization of the linefind channel at the position corresponding to the next posting line. Thereafter, the linefind indication is effectively shifted downwardly by a distance of one posting line after each posting operation is completed. As will later be seen, three distinct linefind indications may selectively be recorded on the strip, if so desired, in spaced positions with respect to each other to effectively divide the face of the ledger card into three printing zones, one disposed above the other, whereby the next line of posting is effected in any selected one of the three printing zones.

The method of recording utilized by the present computer in magnetically recording information in each of the channels of the ferromagnetic strip is the so-called "two-level non-return" system. That is, the length portion of the strip within a channel either is magnetized in one direction—i.e., north-south—or is magnetized in the opposite direction—i.e., south-north. Therefore, a north-south polarization is capable of being indicative of, say, a binary "1," whereas a south-north magnetic polarization is thus indicative of a binary "0." Likewise, if the magnetic strip were initially polarized north-south downwardly from the top portion thereof, as illustrated in FIG. 88B, a south-north polarization is capable of being indicative of a linefind magnetic discontinuity.

During a recording operation, the words are sequentially recorded on the magnetic strip, starting at the top of the card with the highest-order bit of the highest-order digit of the first word and ending with the lowest-order bit of the lowest-order digit of the last word at the bottom of the strip. When the card is subsequently read, the low-order bit of the low-order digit of the last recorded word is read or magnetically sensed first, and the high-order bit of the high-order digit of the first recorded word is last to be read. Thus it is evident that the ledger card is translated in one direction during a recording operation and is translated in the opposite direction during a reading operation.

In order to conserve space during a recording operation, all non-significant zeros located to the left of the highest-order significant digit are suppressed and replaced by an "end-of-word" indication comprising the recording of binary **1111**. Thus, if a word of all zeros is to be recorded, only an end-of-word indication is recorded instead.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107D, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "ROC" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "03", carry out the subinstruction listed in Step-2 of the following ROC program.
2	GO-0-3	Go to Step-3 in the subprogram whenever depression of a motor bar will initiate a machine cycle.
3	Lϕ-4-6	If the low-order digit stored in section 2 of the instruction register is a "0," go to Step-6; otherwise, go to Step-4.
4	L1-5-8	If the low-order digit stored in section 2 of the instruction register is a "1," go to Step-8; otherwise, go to Step-5.
5	L2-0-7	If the low-order digit stored in section 2 of the instruction register is a "2," go to Step-7.
6	BB-0-8	Record single linefind information on the ledger card; thereafter, go to Step-8.
7	LL-0-8	Record multiple linefind information on the ledger card; thereafter, go to Step-8.
8	MIC-0-9	Go to Step-9 as soon as the ledger card is being translated at full velocity outwardly from the carriage of the machine and is in position for recording thereon.
9	AD(2)-0-10	Decrement the digit-counter; thereafter, go to Step-10.
10	PKϕ-0-11	Preset the "K" digit-register to "0"; thereafter, go to Step-11.
11	MK-4-12	Copy into the "K" digit-register the digit, indicated by the count of the digit-counter, of the word stored in memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step-12.
12	Kϕ-14-13	If the digit stored in the "K" digit register is a "0", go to Step-13; otherwise, go to Step-14.
13	Dϕ-9-14	If the digit-counter is at a count of "0", go to Step-14; otherwise, go to Step-9.
14	KEY-0-15	When the 200 cps clock line SAM is rendered FALSE, go to Step-15.
15	D9-16-18	If the digit-counter is at a count of "9", go to Step-18; otherwise, go to Step-16.
16	ROS-0-17	Record an end-of-word (*) notation on the ledger card; thereafter, go to Step-17.
17	Kϕ-18-22	If the digit stored in the "K" digit register is a "0", go to Step-22; otherwise, go to Step-18.
18	SES-0-19	Record the synthronizing clock signal information on the ledger card, and, simultaneously therewith, record on the ledger card the digit stored in the "K" digit-register; thereafter, go to Step-19.
19	Dϕ-20-22	If the digit-counter is at a count of "0", go to Step-22; otherwise, go to Step-20.
20	AD(2)-0-21	Decrement the digit-counter; thereafter, go to Step-21.
21	MK-4-18	Copy the digit into the "K" digit register as indicated by the count of the digit-counter, from the word stored in memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step-18.
22	I34-23-24	If the addresses in sections 3 and 4 of the instruction register are equal, go to Step-24; otherwise, go to Step-23.
23	DE2-0-9	Decrement the contents of section 4 of the instruction register; thereafter, go to Step-9.
24	LK-0-25	Go to Step-25 after the ledger card has been completely ejected from the accounting machine carriage.
25	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

With reference to FIG. 76, the state of flipflop **6092** is reversed upon completion of the previously-initiated subprogram, due to a TRUE-to-FALSE reversal of state of line AN. As previously described, when the state of flipflop **6092** is reversed, so that line MI is rendered

TRUE, a "MI" word-cycle is initiated whereby the word stored in memory at the address specified by the contents of the word-selecting register is read out and stored in the instructions register. Thereafter, if the two-decimal-digit number stored in section 1 of the instruction register corresponds to the code designation for a "ROC" instruction—i.e., "03"—the state of line ROC (FIG. 56) is rendered TRUE. As a result, a "record-on-card" subprogram is thereafter executed in the following manner:

It is first assumed that, upon initiation of the "ROC" subprogram, a ledger card is positioned in the accounting machine carriage, the posting operation on the card is completed, and it is desired to magnetically record line-find information in addition to the last line of posting on the magnetic strip thereof while the card is being ejected from the machine. With reference to FIG. 82, it is further assumed that the able arm of the accounting machine is in the non-deflected position, so that the contacts of switch **496** are closed, thus indicating that the motor bars are not disabled; that the carriage of the machine is not tabulating over a carriage stop so that the contacts of switch **510** are closed; that the machine is in home position so that the contacts of switch **528** are closed; and that a paper-tape-punching operation is not to be initiated, as indicated by a closure of switch contacts SC41—3. The just-mentioned assumptions being satisfied, line GO is rendered TRUE, thus indicating that the accounting machine is ready for a cycle of operation thereof to be initiated.

As it is assumed that the machine carriage is in home position, the contacts of switch **579** are closed, so that line HC is rendered TRUE thereby. Also, as the carriage of the machine is now closed, the movable arm of switch **333** is in the position shown, so that the state of line (D)′ is TRUE. In addition, when line ROC is initially rendered TRUE, lines REC and *i* are likewise rendered TRUE and remain TRUE for substantially the entire time duration of the execution of the "ROC" subprogram. Consequently, when line *i* is initially rendered TRUE, the states of all of the inputs to logical AND **1767** are, for the first time, simultaneously TRUE, and, as a result, solenoid Y is energized thereby. As previously described with respect to FIG. 38, when solenoid Y is energized, carriage throat shaft **458** is rotated counter-clockwise thereby to effect the tripping of the carriage throat clutch, not shown, which allows the carriage to be opened by the power shaft of the machine. During the carriage-opening operation, the front feed chute is tilted forward to the position shown in FIG. 18, so that compression rolls **296** are positioned in close proximity to the card-handling platen **241**, with the ledger card disposed therebetween. In addition, after the carriage is opened, the movable arm of the switch **333** (FIG. 82) is deflected downwardly, as viewed, and thereby renders line D TRUE and line (D)′ FALSE. When line (D)′ is thus rendered FALSE, solenoid Y is de-energized thereby.

As previously described, when the front feed chute is in the position shown in FIG. 18 and the compression rolls **296** are adjacent to, but not latched against, the card-drive platen **241**, the brushes **336** and N are electrically connected together by the commutator bar **342**, as shown in FIG. 28B. However, when the paper guide motor is energized, the cam **391** (FIG. 19) is rotated counter-clockwise to the position shown, to complete the latching operation in a manner previously described. When the latching operation is completed, so that the ledger card is secured between the compression rolls **296** and the platen **241**, the commutator bar **342** will have been rotated clockwise to the position shown in FIG. 28A. As a result, the brushes E and **339** are electrically connected together by the commutator bar **342**.

As diagrammatically illustrated in FIG. 83, the brushes **336** and **339** function in essentially the same manner as the movable arm of a single-pole double-throw switch.

Thus, prior to the latching operation, line E is substantially at ground potential, and the potential on line N is substantially equal to the negative potential applied to brushes **336—339**. Therefore, after carriage-opening solenoid Y (FIG. 82) is de-energized by line (D)′ being rendered FALSE and line D simultaneously being rendered TRUE, paper-guide relay solenoid PGR (FIG. 83) is energized by gate **1777**. Energization of relay solenoid PGR initiates the closure of normally-open contacts PGR—1, PGR—2, and PGR—5, and simultaneously initiates the opening of normally-closed contacts PGR—3 and PGR—4. As shown, an opening of contacts PGR—3 and PGR—4 and a closure of contacts PGR—1 and PGR—2 cause a reversal of subsequently-applied current flow through the armature of paper guide motor PGM, which is of the permanent magnet type, and thereby cause a reversal of subsequent rotation thereof. A closure of contacts PGR—5, of course, causes armature current to flow through paper-guide motor PGM.. Therefore, when the paper guide relay is energized, armature current is applied to the paper-guide motor, so that its armature is rotated in a direction to effect the latching of the compression rolls **296** against the card-drive platen **241**, as illustrated in FIG. 19.

After the latching operation is completed, so that the commutator **340** is rotated by the paper-guide motor to the position shown in FIG. 28A, line E (FIG. 83) is rendered FALSE and thereby causes the paper-guide motor to be de-energized by effecting a de-energization of paperguide relay PGR, and thus initiating the return of the relay contacts thereof to their normal conditions, as shown. In addition, with reference to FIG. 17, after the latching operation is completed, the radiant energy from the light source **373** is reflected from the ledger card back onto the sensitive areas of photocells PCB and PCC and thereby energize both. Consequently, with reference back to FIG. 83, when photocells PCB and PCC are energized, lines B and C are respectively rendered TRUE thereby in the manner previously described. When line E is rendered FALSE, due to the fact that the paper guide is now in a latched condition, lines (E)′ and N are, of course, both rendered TRUE.

As shown in FIG. 82, when line (E)′ is rendered TRUE, energization of solenoid X is effected by gate **1769**. As previously described with respect to FIGS. 16 and 17, energization of solenoid X allows subsequent rotation of the square shaft **283** and thereby allows the lower compression rolls **288** to be disengaged from the platen **71**. As also previously described with respect to FIG. 19, when the square shaft **283** is rotated counter-clockwise, as viewed, to cause the lower compression rolls to be disengaged from the accounting machine platen, the contacts of the switch **545** are operated thereby. Therefore, with reference back to FIG. 82, after the lower compression roll disengagement operation is completed, the movable arm of switch **545** is deflected downwardly, as viewed, and thus causes the state of line G to thereafter be TRUE. Simultaneously therewith, line (G)′ is rendered FALSE and thereby causes solenoid X to be de-energized. When line G is thus rendered TRUE, card drive motor CDM is caused to be energized by gate **1770**. As a result, the ledger card is driven toward the rear of the machine and back into the chute **291** by the platen **241**, in the manner previously described with respect to FIGS. 17 through 19.

As previously described with respect to that portion of the preceding description relating to the format of a "ROC" instruction word, the numeral "0" in the low-order digital position of section 2 thereof causes a " record single line find information" operation to subsequently be initiated; the numeral "1" in the low-order digital position of section 2 thereof causes a "record no linefind information operation to subsequently be initiated; and the numeral "2" in the low-order digital position of section **2** thereof causes a "record multiple linefind information" operation to subsequently be initiated. As previously described with respect to FIG. 57, if the value of the digit stored in the low-order digital position of section 2 of the instruction register is a "0," a "1," or a "2," the state of one of lines Lϕ through L2 is respectively rendered TRUE.

With reference now to FIG. 85, if it is assumed that single linefind information is desired to be recorded, as indicated by a TRUE state of line Lϕ, lines L1 and L2, of course, being FALSE, the reference input to flipflop **6140** is reversed from TRUE to FALSE when line G is rendered TRUE in the manner just described. Consequently, the state of flipflop **6140** is reversed, so that the upper end, as viewed, of the winding of linefind head **346** is rendered TRUE. Simultaneously therewith, the output of gate **1800** is rendered TRUE, so that the center tap of the winding of the linefind head 346 is thereby rendered FALSE. As a result, current is caused to flow between the center tap and the upper end of the winding of linefind head **346** in such a direction as to magnetize the linefind channel portion of the magnetic strip of the ledger card in the opposite direction to the pre-magnetized direction. However, approximately ten to forty milliseconds later, the state of flipflop **6140** reverts back to its original condition, so that the lower end of the winding of linefind head 346 is now rendered TRUE instead of its upper end, as before. Consequently, current is caused to flow between the center tap and the lower end of the linefind winding in such a direction as to magnetize the remaining portion of the linefind channel in the same direction as the pre-magnetized direction, thereby simultaneously erasing all previously-recorded linefind information.

It is now assumed that multiple linefind information is desired to be recorded, so that line L2 is initially rendered TRUE instead of line Lϕ. Thus, when the state of flipflop **6140** is initially reversed in the manner just described, the state of flipflop **6141** is likewise reversed simultaneously therewith. Thus, again, the states of the upper end and center tap of the winding of linefind head **346** are rendered TRUE and FALSE, respectively, as before described. Consequently, the linefind channel is, again, first magnetized in the opposite pre-magnetized direction by the first reversal of state of flipflop **6140** and, thereafter, is magnetized in the same direction as the pre-magnetized direction by a second reversal of state of flipflop 6140, as before described. However, approximately sixty milliseconds after the state of flipflop **6141** is initially reversed, its state reverts back to its initial condition, so that the state of the center tap of linefind head **346** is rendered TRUE thereby, thus essentially de-energizing the line-find head and thereby preventing further erasure of previously-recorded linefind information, all of which is diagrammatically illustrated in FIG. 88B. It is to be noted that, if it is desired for no linefind information to be recorded in the linefind channel, as indicated by a TRUE state of line **L1**, the linefind head is prevented from being energized due to the fact that the state of line (L1)' is thus FALSE and thereby prevents the output of logical AND **1798** from subsequently being rendered TRUE.

With reference to FIG. 17, when the ledger card has been driven to the full extent into chute **291**, the reversal spot located at the top margin of the ledger card, indicated as **376** in FIG. 88A, passes over the junction of the leg members **371** and **372**. Consequently, the radiant energy from light source **373** is absorbed by the dark area thereof to the extent that photocell PCB is thus de-energized. With reference to FIG. 83, when photocell PCB is de-energized, the state of line B is rendered FALSE thereby. When line B is thus rendered FALSE, the state of flipflop **6122** is reversed, so that line P is rendered TRUE, line (P)', of course, being simultaneously rendered FALSE. When line (P)' is thus rendered FALSE, card-drive relay coil CDR is de-energized, and, as a result, relay contacts CDR—**5** are closed and thereby render line (U)' TRUE.

With reference to FIG. 82, also when coil CDR is de-energized, relay contacts CDR—**1** and CDR—**2** are opened, whereas relay contacts CDR—**3** and CDR—**4** are closed, as shown, thus causing a reversal of subsequently-applied current through the armature of motor CDM resulting in a subsequent reversal of rotation thereof. However, when line (P)' is rendered FALSE, card-driven motor CDM is de-energized due to the fact that the output of gate **1770** is simultaneously rendered FALSE. With reference back to FIG. 83, when line (P)' is thus rendered FALSE, the state of flipflop **6123** is reversed, so that card-drive brake solenoid CDB is energized. Consequently, when solenoid CDB is energized, the ledger card is brought to a sudden standstill in the manner previously described with respect to FIG. 14. Approximately 120 milliseconds after solenoid CDB is energized, flipflop **6123** reverts back to its initial state of energization, so that solenoid CDB is thereby de-energized.

With reference to FIG. 82, when line (U)' is rendered TRUE due to the card-drive relay coil being de-energized, current is applied through the armature of card-drive motor CDM via gate **1771** in such a direction that the direction of rotation of card-drive motor is reversed and the ledger card is driven outwardly from the carriage.

As soon as the ledger card begins to be translated outwardly from the accounting machine carriage toward the loading table of the card-handling equipment, the reversal spot on the top margin thereof is moved away from photocell PCB (FIG. 17), and photocell PCB is consequently exposed to light source 373. With reference back to FIG. 83, when photocell PCB is exposed to radiant energy, line B is rendered TRUE thereby.

With reference to FIG. 85, when line B is again rendered TRUE, all of the inputs to logical AND **1801** are simultaneously TRUE, and, as a result, the reference input to flipflop **6143** is reversed from a TRUE state to a FALSE state. Consequently, flipflop **6143** reverses state, so that line SCD is rendered TRUE when the reversal spot on the top margin of the ledger card is moved away from photocell PCB. Approximately 75 milliseconds later, however, flipflop **6143** again reverses state, and line SCD is thereby rendered FALSE. As will be seen later, flipflip **6143** provides a 75-millisecond delay in order to allow the ledger-card-driving means sufficient time to accelerate the ledger card to full velocity before any information is magnetically recorded thereon, thus obtaining a recording margin at the top of the card of approximately one half-inch in width.

With reference to FIGS. 92A through 92C, there is logically illustrated therein a composite circuit diagram of particular portions of the computer circuitry utilized in executing a "ROC" instruction. With particular reference to FIG. 92B, when line SCD experiences a TRUE-to-FALSE reversal of state, indicating that the ledger card has reached full velocity and is in position for recording thereon, the state of flipflop **6093** is reversed, so that line MIC is rendered TRUE thereby.

As previously described with respect to section (*b*) of FIG. 50, multivibrator **6126**, shown in FIG. 92A, is a free-running 400-cycle-per-second square-wave generator synchronized with respect to the changes of state of line C41. Consequently, the reference output of multivibrator **6126** is rendered TRUE at TIME–1 when the state of line C41 reverses from TRUE to FALSE and, approximately 1240 microseconds later at TIME–**1**, is rendered FALSE and, approximately 1240 microseconds later at TIME–1, is again rendered TRUE, and so on. As shown, the reference and prime output lines from multivibrator **6126** are respectively connected to the reference and prime input lines of flipflop **6133**. Therefore, the state of line ICC is sequentially reversed from TRUE to FALSE and back to TRUE at a 400-

cycle-per-second rate but essentially 180 degrees out of phase with changes of state of the reference output of multivibrator **6126**. By the same token, when line ROC is rendered TRUE at the beginning of the "ROC" sub-program, the state of line GC is reversed at the same rate as the state of line ICC but essentially 180 degrees out of phase therewith. Flipflop **6102** operates as a frequency divider which essentially divides the output at line GC by units of "2." Thus, it is evident, the state of line SAM is sequentially reversed from TRUE to FALSE to TRUE at a 200-cycle-per-second rate, with each change of state thereof occurring at TIME–**1**. In other words, line SAM is rendered FALSE at the first TIME–1, remains FALSE for approximately 2480 microseconds and is rendered TRUE at TIME–1, remains TRUE for approximately 2480 microseconds and is rendered FALSE at TIME–**1**, and so on.

Therefore, with reference to FIG. 98B, when line GC is first rendered TRUE at TIME–**1** after line ROC is rendered TRUE, the reference input to flipflop **6088** is reversed from TRUE to FALSE. Consequently, flipflop **6088** reverses state, so that line JHR is rendered TRUE thereby. At the following TIME–1, the state of flipflop **6088** is again reversed, and line JHR is thereby rendered FALSE. Thus, every 2480 microseconds at TIME–**1** after line ROC is rendered TRUE, line JHR is rendered TRUE and remains TRUE for 40 microseconds and is rendered FALSE at the following TIME–**1**. With reference now to FIG. 92A, when the reversal spot on the ledger card is moved away from photocell PCB and the ledger card is being translated outwardly from the carriage at maximum velocity, all the inputs to logical AND **1366** are TRUE after line MIC is rendered TRUE. Consequently, when line JHR is thereafter rendered FALSE, the state of flipflop **6061** is reversed, so that line TH is rendered TRUE. At TIME–**3** after line TH is rendered TRUE, line AD experiences a TRUE-to-FALSE reversal of state. As shown in FIG. 92C, when line ROC is initially rendered TRUE, line DBD is likewise rendered TRUE.

As previously described with respect to the digit-counter (FIG. 62), after line DBD is rendered TRUE, subsequent successive TRUE-to-FALSE reversal of states of line AD causes a successive decrementing of the count of the digit-counter. It is seen, therefore, as line DBD is TRUE when line AD is rendered FALSE, that the digit-counter is decremented from a count of "0" to a count of "9." Also, as shown in FIG. 92A, when line TH is rendered TRUE, line KKF is likewise rendered TRUE. Therefore, when line AD is rendered FALSE, line PKϕ also experiences a TRUE-to-FALSE reversal of state, and, as a result, the "K" digit-register (FIG. 70) is preset to "0."

With reference to FIG. 92C, when lines ROC and TH are simultaneously TRUE, line MK is rendered TRUE. When line MK is thus rendered TRUE, line TS4 is likewise rendered TRUE. Thus, in response to lines MK and TS4 being TRUE, a "MK" digit-cycle is thereafter executed whereby the high-order digit of the word stored in memory at the address specified by the contents of section 4 of the instruction register is stored in the "K" digit-register. At TIME–1 after the high-order digit of the word in memory is stored in the "K" digit-register, the state of flipflop **6061** (FIG. 92A) is reversed, and line TH is thereby rendered FALSE. When line TH is thus rendered FALSE, the state of flipflop **6095** (FIG. 92B) is reversed, and line OBM is thereby rendered TRUE. However, forty microseconds later at the following TIME–**1**, the state of flipflop **6095** is again reversed, and line OBM is rendered FALSE.

As previously described with respect to FIG. 70, whenever the "K" digit-register is storing a decimal digit having a value of zero, line Kϕ is rendered TRUE; otherwise, line (Kϕ)′ is rendered TRUE. If it is assumed that the digit just stored in the "K" digit-register has a value other than zero, the state of flipflop **6089** (FIG. 92B) is reversed when line OBM is rendered FALSE. Consequently, line KEY is thereby rendered TRUE, indicating that the computer is ready to magnetically record the digit now stored in the "K" digit-register on the magnetic strip of the ledger card. Actual recording on the ledger card, however, does not begin until line SAM (FIG. 92A) is rendered FALSE. When line SAM is subsequently rendered FALSE due to a reversal of state of flipflop **6102** approximately 2480 microseconds after being rendered TRUE, the state of flipflop **6089** (FIG. 92B) is again reversed, so that line KEY is thereby rendered FALSE.

With reference to FIG. 92C, when line KEY experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6104** is reversed, and line SES is rendered TRUE and remains TRUE until all data and clock information has been magnetically recorded on the ledger card. With reference now to FIG. 85, as lines SES and ROC remain TRUE during the entire recording operation, recording current effectively flows through the winding of clock recording-reproducing head **347** in one direction or in the opposite direction, depending upon whether the state of line SAM is TRUE or FALSE. Thus, it is evident, when line SES is rendered TRUE, an approximately 200-cycle-per-second square-wave signal is magnetically recorded in the clock channel of the ledger card while the card is being translated outwardly from the carriage at a linear speed of approximately ten inches per second.

As previously described with respect to FIG. 70, the output of the "K" digit-register is effectively serialized, in that the digit stored therein is effectively "read out" bit-by-bit and the state of line KS is sequentially conditioned either TRUE or FALSE for each bit, depending upon whether the bit just read out is either a "1" or a "0," respectively. Therefore, when line KEY (FIG. 61) is rendered FALSE, line CYC is reversed from TRUE to FALSE. As a result, the bit-counter (FIG. 62) is decremented from a binary count of "*a*" to a binary count of "*d*." With reference back to FIG. 70, if it is assumed that the digit previously stored in the "K" digit-register is a "9"—i.e., binary 1001—when the bit-counter is at a binary count of "*d*," so that line B*d* is TRUE, line KS is rendered TRUE, indicating that bit "*d*" of the digit stored in the "K" digit-register is a binary "1"; when the bit-counter is subsequently decremented to a count of "*c*" due to a change of state line (GC)′, hence line CYC, line KS is rendered FALSE, indicating that bit "*c*" of the digit stored in the "K" digit-register is a binary "0"; when the bit-counter is subsequently decremented to a count of "*b*," line KS remains FALSE, indicating that bit "*b*" is also "0"; and, finally, when the bit-counter is decremented to a count of "*a*," line KS is rendered TRUE, indicating that bit "*a*" is a binary "1."

As previously described, the states of flipflops **6102** (FIG. 92A) and **6088** (FIG. 92B) are simultaneously reversed by a change of state of line (GC)′, so that lines SAM and JHR are simultaneously rendered TRUE. Forty microseconds later, however, the state of flipflop **6088** is again reversed, so that line JHR is rendered FALSE.

With reference to FIG. 92C, it is evident that the state of line OBN is conditioned TRUE or FALSE via flipflop **6096** each time the output of gate **1629** is rendered FALSE, to correspond to the value of the binary bit just read out from the "K" digit-register. That is, if the bit is a binary "1," line OBN is rendered TRUE; if the bit is a binary "0," line OBN remains FALSE. With reference to FIG. 85, it is evident that, forty microseconds after each reversal of state of line SAM, current is effectively caused to flow through the winding of data recording-reproducing head **348** in one direction when line OBN is rendered TRUE, and is caused to flow in an opposite direction when line OBN is rendered FALSE. Thus, it is seen the data information is magnetically recorded in synchronism with the 200-cycle-per-second clock information.

With reference back to FIG. 92A, during the time the tenth-order digit of the word read out from memory is being recorded on the ledger card, so that the bit-counter is again at a count of "a," the output of gate **1370** reverses from TRUE to FALSE when the state of line JHR again experiences a TRUE-to-FALSE reversal of state. As a result, the state of flipflop **6061** is reversed, and line TH is again rendered TRUE for the second time. At the following TIME–3 after line TH is rendered TRUE, the state of line AD is reversed, so that the digit-counter is decremented from a count of "9" to a count of "8." Simultaneously therewith, the "K" digit-register is preset to zero via change of state of line PKϕ.

With reference to FIG. 92C, when line TH is thus rendered TRUE, line MK is likewise rendered TRUE for the second time. Again, when line MK is thus rendered TRUE, line TS4 is rendered TRUE. Consequently, in response to lines MK and TS4 thus being rendered TRUE, a second "MK" digit-cycle is thereafter executed, whereby the ninth-order digit of the word stored in memory at the address specified by the contents of section 4 of the instruction register is read out, stored in the "K" digit-register, and thereafter magnetically recorded in the data channel of the ledger card in substantially the same manner as just described. With reference to FIG. 92B, at TIME–3 after the digit-counter reaches a count of "0," so that line Dϕ is TRUE, and the bit-counter reaches a count of "a," so that line Ba is TRUE, thus indicating that the first-order digit of the first word read from memory is being magnetically recorded on the ledger card, the state of flipflop **6094** is reversed, so that line MOU is rendered TRUE thereby.

After the first-order digit of the first word read from memory is magnetically recorded on the ledger card, the contents of sections 3 and 4 of the instruction register are compared for equality in the same manner as previously described. If the contents of sections 3 and 4 are not equal, indicating that there are additional words to be read from memory and subsequently recorded on the card, section 4 is decremented, via line DE2, approximately twenty microseconds after line MOU is rendered TRUE. Thereafter line MOU is rendered FALSE and the above sequence of events is again repeated, in substantially the same manner as just described. When the contents of sections 3 and 4 of the instruction register are of equal magnitude, indicating that there are no more words to be read from memory and magnetically recorded on the ledger card, line (I34)′ is rendered TRUE in the same manner as previously described with respect to FIG. 59.

Thus, as the state of line (I34)′ is now TRUE, the state of flipflop **6094** is again reversed in response to the next succeeding TRUE-to-FALSE reversal of state of line (GC)′, and, as a result, line MOU is thus rendered FALSE. With reference to FIG. 92C, when line MOU is thus rendered FALSE, the state of flipflop **6104** is reversed, so that line SES is rendered FALSE thereby, thereafter terminating all recording of clock and data information in the respective channels of the ledger card strip. When line SES is thus rendered FALSE, the state of flipflop **6090** is reversed, so that line LK is rendered TRUE thereby.

As previously mentioned, when a word is read out from memory to substantially be magnetically recorded on the ledger card, all non-significant zeros are essentially replaced by a single "end-of-word" notation, the end-of-word notation being recorded as a binary 1111.

With reference to FIG. 92B, if it is now assumed that the tenth-order digit of the word read out from memory and initially stored in the "K" digit-register is a zero, the state of flipflop **6089** remains unchanged, so that line KEY remains FALSE when line OBM is rendered FALSE as before. However, if line KEY was previously rendered TRUE prior to the initiation of the "ROC" subprogram, the state of flipflop **6089** is reversed when line SAM is thereafter rendered FALSE, and, as a result, line KEY is rendered FALSE thereby. When line KEY is rendered FALSE at the beginning of the recording operation, the state of flipflop **6098** is reversed, so that line ROS is thereby rendered TRUE. Therefore, with reference to FIG. 92C, due to the fact that line Bd is TRUE at the beginning of the recording operation and line JHR is rendered TRUE when line (GC)′ is first rendered FALSE, as before described, the state of flipflop **6096** is reversed when line JHR is rendered FALSE approximately forty microseconds after line ROS is first rendered TRUE. Consequently, line OBN is rendered TRUE thereby. Approximately five milliseconds after line OBN is rendered TRUE, the state of flipflop **6096** is reversed when lines SAM and (GC)′ are both rendered FALSE. With reference to FIG. 92B, approximately five milliseconds after line (OBN)′ is thus rendered TRUE, the state of flipflop **6093** is again reversed when lines SAM and (GC)′ are both again simultaneously rendered FALSE. As a result, line ROC is thereby rendered FALSE.

With reference now to FIG. 85, when line ROS is thus rendered TRUE and remains TRUE for approximately ten milliseconds, during which time both of lines SES and ROC are TRUE, current is caused to flow through the winding of data recording-reproducing head **348**, via gate **1796**, in the same direction for approximately ten milliseconds, and in such a direction therethrough as to cause four successive binary "1's" to be recorded in the data channel. At the following TIME–1 when line JHR is rendered FALSE after line (ROS)′ is rendered TRUE, the state of flipflop **6096** (FIG. 92C) is reversed, and, as a result, line OBN is rendered TRUE and remains TRUE until lines (GC)′ and SAM are subsequently rendered FALSE. As line (OBN)′ is simultaneously rendered FALSE when line OBN is rendered TRUE, the state of flipflop **6093** (FIG. 92B) is prevented from being reversed until line SAM is subsequently rendered FALSE approximately 160 microseconds after line ROS is rendered TRUE. However, when line SAM is rendered FALSE, the state of flipflop **6098** is reversed, and line ROS is rendered FALSE, thus terminating the end-of-word recording operation. It is to be noted that, while the state of line ROS is TRUE, the state of line OBN is not sequentially conditioned, as before, to successively correspond to the value of the binary bits of digit stored in the "K" digit-register.

With reference to FIG. 17, just prior to the ledger card being completely ejected from the carriage, the bottom edge thereof first passes over the top of the rightmost extension of arm **374**. Thereafter, photocell PCC is de-energized due to the fact that the radiant energy from light source **373** is not reflected by the ledger card onto the light-sensitive area thereof, as before. As previously described, photocell PCC is connected in the circuit of preamplifier **4733** (FIG. 83) in such a manner that the state of line C is rendered FALSE when photocell PCC is de-energized. When line C is thus rendered FALSE, the state of flipflop **6122** is reversed, so that line (P)′ is rendered FALSE thereby, thus de-energizing card-drive relay coil CDR. When coil CDR is de-energized, relay contacts CDR–5 are closed, and line (U)′ is thereby rendered TRUE. With reference to FIG. 82, when line (P)′ is thus rendered FALSE, card-drive motor CDM is de-energized. With reference back to FIG. 83, also when line (P)′ is rendered FALSE, the state of flipflop **6123** is reversed, so that card-drive brake coil CDB is energized for a period of approximately 120 milliseconds and thus causes the card-drive motor CDM to be brought to a sudden standstill in the manner previously described with respect to FIG. 14. After a time lapse of approximately 120 milliseconds, the state of flipflop **6123** is again reversed, so that card-drive brake coil CDB is de-energized thereby.

With reference back to FIG. 17, when the bottom edge of the ledger card passes over the junction of arms **371**—**372**, photocell PCB is de-energized due to the fact that the radiant energy from the light source **373** is thereafter not reflected by the ledger card onto the active area thereof. As previously described, photocell PCB is connected in the circuit of pre-amplifier **4736** (FIG. 83) in such a manner that the state of line (B)′ is rendered TRUE when photocell PCB is de-energized. Thus, when line (B)′ is rendered TRUE, all of the inputs to gate **1775** are, for the first time, simultaneously TRUE, so that paper-guide motor PGM is energized and is thereafter caused to rotate in the opposite direction, due to the fact that the paper-guide relay coil PGR was previously de-energized, so that relay contacts PGR1 through PGR4 are caused to resume their normal conditions, as shown. As previously described with respect to FIG. 19, when the paper-guide motor is caused to be rotated in a direction opposite to that during a latching operation, the cam **301** is rotated clockwise, as viewed, so that the compression rolls **296** and **241** are unlatched from each other. During the unlatching operation, the commutator **340** is rotated to the position shown in FIG. 28A. Consequently, with reference to FIG. 83, the brushes **336** and **339** are effectively moved downwardly to the position shown, so that lines (E)′ and N are rendered FALSE thereby. When the state of line N is thus rendered FALSE, the state of flipflop **6122** is again reversed, so that line (P)′ is rendered TRUE and thereby re-energizes card-drive relay coil CDR. When coil CDR is energized, contacts CDR-5 are opened and thereby effect de-energization of paper-guide motor PGM.

With reference to FIG. 82, when the state of line P is thus rendered FALSE, the state of line (EOC)′ is thereby rendered TRUE. Therefore, with reference to FIG. 92C, when line (EOC)′ is thus rendered TRUE, indicating that the ledger card is completely ejected from the carriage, the state of flipflop **6090** is reversed upon occurrence of the next TRUE-to-FALSE reversal of state of line AN. When the state of flipflop **6090** is thereby reversed, line LK is rendered FALSE. As shown in FIG. 92A, when line LK is rendered FALSE, the state of flipflop **6107** is reversed, so that line STD is rendered TRUE thereby. When line STD is thus rendered TRUE, line TS5 is likewise rendered TRUE. In response to lines STD and TS5 thus being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register in the manner previously described. In addition, when line STD is rendered TRUE, line (STD)′, of course, being rendered FALSE, line REC (FIG. 82) is rendered FALSE, thus terminating the sequence of events initiated by the "ROC" instruction.

73. *Detailed Description of ECW Instruction*

In executing an "ECW" instruction, the control counter portion of the computer operates in conjunction with the card-handling portion thereof to effect the individual reproduction of linefind, clock, and data information, each being previously recorded magnetically in a different one of the three channels of the magnetic strip portion of the ledger card in the manner just described.

The subprogram for the execution of such an "ECW" instruction may be divided into three distinct phases. During the first phase of the subprogram, the data information, in the form of a sequence of words, is first read from the magnetic strip and is thereafter stored in the designated memory address locations. If the ledger card does not have any information magnetically recorded thereon, such as, for example, a new ledger card, even though there is no information magnetically reproduced,

the sequence of operation nevertheless remains substantially unchanged. During the second phase of the subprogram, after the ledger card has reached the maximum rearward position within the accounting machine carriage, the card either is immediately ejected from the carriage or is placed in a proper position within the carriage ready for a position operation thereon. After the ledger card is either ejected or placed on the next posting line, the third and last phase of the subprogram causes the carriage to be closed and the preselected motor bar to be depressed for either a touch or a hold operation.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107C, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing an "ECW" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "02", carry out the subinstruction listed in Step-2a of the following ECW subprogram.
2a	PJϕ-0-2b	Preset the "J" digit-register to "0"; thereafter, go to Step-2b.
2b	RLR-0-3	Preset "R" counter to correspond to low-order digit of section 2 of the Instruction Register and thereafter go to Step-3.
3	OBN(3)-0-4	Preset line OBN to a FALSE state; thereafter, go to Step-4.
4	CCR-20-5	Go to Step-5 when a synchronizing clock impulse is magnetically picked up from the ledger card. However, when the ledger card is all the way into the carriage just prior to reversal, go to Step-20 if a clock impulse has not been picked up.
5	CYC-0-6	Increment the bit-counter; thereafter, go to Step-6.
6	JL(1)-0-7	Store in the "J" digit-register the data bit read from the ledger card, which bit corresponds to the count of the bit-counter; thereafter, go to Step-7.
7	Bd-4-8	If the bit-counter is at a count of "d," go to Step-8; otherwise, go to Step-4.
8	D9-11-9	If the digit-counter is at a count of "9," go to Step-9; otherwise, go to Step-11.
9	CM-3-10	Clear the address in memory specified by section-3 of the instruction register; thereafter, go to Step-10.
10	OBN(2)-0-11	Preset line OBN to a TRUE state; thereafter, go to Step-11.
11	EOW-14-12	If the "J" digit-register is storing "15," go to Step-12; otherwise, go to Step-14.
12	D9-13-17	If the digit-counter is at a count of "9," go to Step-17; otherwise, go to Step-13.
13	AD(1)-0-12	Increment the digit-counter; thereafter, go to Step-12.
14	AD(1)-0-15	Increment the digit-counter; thereafter, go to Step-15.
15	JM-3-16	Copy the digit stored in the "J" digit-register memory at the digital position indicated by the count of the digit-counter, the address in memory being specified by section 3 of the instruction register; thereafter, go to Step-16.
16	D9-4-17	If the digit-counter is at a count of "9," go to Step-17; otherwise, go to Step-4.
17	I34-18-19	If the memory addresses specified by sections 3 and 4 of the instruction register are equal, go to Step-19; otherwise, go to Step-18.
18	IN4-0-4	Increment section 3 of the instruction register; thereafter, go to Step-4.
19	L1-22-23	If the low-order digit stored in section 2 of the instruction register is a "1," go to Step-23; otherwise, go to Step-22.
20	OBN (1)-22-21.	If the state of line OBN is TRUE, go to Step-21 otherwise, go to Step-22.
21	MOU (2)-0-4	Eject the ledger card; thereafter, go to Step-4.
22	LFA-0-24	Go to Step-24 when the next linefind impulse is magnetically picked up.
23	MOU (2)-0-27.	Eject the ledger card; thereafter, go to Step-27.
24	DR-0-25	Decrement the R-counter; thereafter, go to Step-25.
25	Rϕ-22-26	If the R-counter is at a count of "0," go to Step-26; otherwise, go to Step-22.
26	L-0-27	Position the ledger card on the next posting line; thereafter, go to Step-27.
27	L1-28-30	If the low-order digit in section 2 of the instruction register is a "1," go to Step-30; otherwise, go to Step-28.
28	L3-29-30	If the low-order digit in section-2 of the instruction register is a "3," go to Step-30; otherwise, go to Step-29.
29	RPB-0-30	Upon depression of the resume-program bar, go to Step-30.
30	CLC-0-31	Close the accounting machine carriage; thereafter, go to Step-31.

Step	Subinstructions	Description
31	GO-0-32	Go to Step-32 in the Subprogram whenever depression of a motor bar will initiate a machine cycle
32	Hϕ-34-33	If the high-order digit in section 2 of the instruction register is a "0," go to Step-33; otherwise, go to Step-34.
33	MBϕ (1)-0-45	Depress the upper motor bar for a touch operation thereafter, go to Step-45.
34	H1-36-35	If the high-order digit in section 2 of the instruction register is a "1," go to Step-35; otherwise. go to Step-36.
35	MBϕ (2)-0-45	Depress the upper motor bar for a hold operation; thereafter, go to Step-45.
36	H2-38-37	If the high-order digit in section 2 of the instruction register is a "2," go to Step-37; otherwise, go to Step-38.
37	MB1 (1)-0-45	Depress the middle motor bar for a touch operation; thereafter, go to Step-45.
38	H3-40-39	If the high-order digit in section 2 of the instruction register is a "3," go to Step-39; otherwise, go to Step-40.
39	MB1 (2)-0-45	Depress the middle motor bar for a hold operation; thereafter. go to Step-45.
40	H4-42-41	If the high-order digit in section 2 of the instruction register is a "4," go to Step-41; otherwise, go to Step-42.
41	MB2 (1)-0-45	Depress the middle motor bar for a hold operation; thereafter, go to Step-45.
42	H5-44-43	If the high-order digit in section 2 of the instruction register is a "5," go to Step-43; otherwise, go to Step-44.
43	MB2 (2)-0-45	Depress the lower motor bar for a hold operation; thereafter, go to Step-45.
44	H6-0-48	If the high-order digit in section 2 of the instruction register is a "6," go to Step-48.
45	PCT-0-46	When the amount racks are traveling in a setting direction, go to Step-46 in the subprogram each time the timing rack changes from one digital valued position to another, and also upon engagement of the printing-aligner.
46	AF-0-47	Increment the F-counter; thereafter, go to Step-47.
47	F9-45-48	If the F-counter is at a count of "9," go to Step-48; otherwise, go to Step-45.
48	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

With reference to FIG. 76, upon completion of the previously-initiated subprogram, the state of flipflop **6092** is reversed by a TRUE-to-FALSE reversal of state of line AN. As previously described, when the state of flipflop **6092** is reversed, so that line MI is rendered TRUE, a "MI" word-cycle is initiated, whereby the word stored in memory, at the address indicated by the contents of the word-selecting register, is read out and thereafter stored in the instruction register. Thereafter, if the two-decimal digit number stored in section 1 of the instruction register corresponds to the code designation for an "ECW" instruction—i.e., "02"—the state of line ECW (FIG. 56) is rendered TRUE. As a result of line ECW thus being rendered TRUE, an "enter-card-words" subprogram is executed in the following manner:

At TIME-**4** during the last bit time after the previously-initiated "MI" word-cycle is substantially completed, line AN (FIG. 61) is again rendered TRUE and, ten microseconds later at TIME-**1**, is again rendered FALSE. With reference back to FIG. 76, when line AN is rendered FALSE upon completion of the "MI" word-cycle operation, the state of flipflop **6092** is again reversed, and line MI is rendered FALSE, thus terminating the "MI" word-cycle operation. As previously mentioned with respect to FIG. 72, as long as lines MI and AN are both TRUE, line MIN is TRUE. Thus, when line AN is rendered FALSE, line MIN is likewise rendered FALSE.

Reference is now made to FIGS. 91A through 91C, wherein there is logically illustrated a composite circuit diagram of various portions of the computer circuitry utilized in executing an "ECW" instruction. With particular reference to FIG. 91B, when line MIN experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6094** is reversed, and line MOU is thereby rendered TRUE. As long as lines MOU and ECW are both TRUE, line RED is likewise TRUE.

Upon initiation of an "ECW" instruction, it is assumed that the carriage of the accounting machine is in home position, so that the contacts of switch **579** (FIG. 82) are closed and line HC is thereby rendered TRUE; that, after the carriage reaches home position, depression of a motor bar is capable of initiating a cycle of operation of the accounting machine, as indicated by the contacts of each of serially-connected switches **496**, **510**, **528**, and **K1—7** being closed, so that line GO is thereby rendered TRUE; and that the carriage of the accounting machine is closed, so that the movable arm of switch **333** is in the position shown, so that line (D)′ is TRUE.

Therefore, when line RED is initially rendered TRUE, line *i* likewise being rendered TRUE, all of the inputs to gate **1767** are simultaneously TRUE. As a result, solenoid Y is energized and thereafter causes the accounting machine carriage to be opened in the manner previously described with respect to FIG. 38. If it is assumed that the lower compression rolls **288** are in the latched position, as shown in FIG. 17, the movable arm of switch **545** (FIG. 82) is therefore in the position shown (see also FIG. 19), so that line (G)′ is thereby rendered TRUE. After the carriage opening operation is completed, the movable arm of switch **333** is deflected downwardly, as viewed, so that line (D)′ is rendered FALSE, thereby de-energizing carriage-open solenoid Y. However, when line D is rendered TRUE when line (D)′ is rendered FALSE, solenoid X is energized and thereafter allows the lower compression rolls **288** to fall away from the platen **71** in the manner previously described with respect to FIGS. 15 through 17. After the lower compression rolls are disengaged from the accounting machine platen, the movable arm of the switch **545** (FIG. 82) is deflected downwardly, in the manner previously described with respect to FIG. 19, whereby line (G)′ is rendered FALSE, so that solenoid X is thereby de-energized. However, when line G is rendered TRUE, all of the inputs to gate **1757** (FIG. 81) are simultaneously TRUE, and, as a result, enter-card lamp EC (see also FIGS. 1 and 2) is illuminated, thus providing a visual indication that the computer is ready to receive a ledger card.

After a ledger card of the type shown in FIG. 88A is placed face down on the loading table, with its leftmost vertical edge against the card-alignment surfaces, as illustrated in FIG. 22, both of the photocells PCA**1** and PCA**2** are de-energized thereby. With reference to FIG. 83, photocells PCA**1** and PCA**2** are connected in the circuit of pre-amplifiers **4734** and **4735**, respectively, and in a manner previously described with respect to FIG. 51*h*, so that lines A1 and A2 are respectively rendered TRUE when photocells PCA**1** and PCA**2** are de-energized. With reference to FIG. 22, when the ledger card is manually moved upwardly as viewed, photocell PCB is energized in a manner previously described with respect to FIG. 17. With reference now to FIG. 83, when photocell PCB is energized, line B is thereby rendered TRUE. As a result of the inputs to gate **1776** all being TRUE simultaneously, paper-guide relay coil PGR is energized. Consequently, relay contacts PGR—**1**, PGR—**2**, and PGR—**5** are closed, whereas contacts PGR—**3** and PGR—**4** are opened thereby. As a result, paper-guide motor PGM is thereby energized and thereafter rotates latching cam **301** (FIG. 19) counter-clockwise to latch the compression roll **296** against the card-handling platen **241**, as shown, with the forward edge portion of the ledger card being firmly secured therebetween.

After the latching operation is completed, line E is rendered FALSE and line N is rendered TRUE in a manner previously described with respect to FIGS. 16 and 28B. As shown in FIG. 83, when line E is thus rendered FALSE, paper-guide relay coil PGR is de-energized, and the contacts operated thereby resume their normal conditions, as shown. As a result, paper-guide motor PGM is thereby de-energized. With reference to FIG. 82, also when line E is thus rendered FALSE, the state of flipflop **6121** is reversed, so that line (KK)′ is momentarily rendered FALSE. However, after a time lapse of approximately 250 milliseconds, the state of flip-flop **6121** reverts back to its original state, so that line (KK)′ is rendered TRUE, and, as a result, card-drive motor CDM is energized thereby.

When the card-drive motor is energized at this time, the ledger card is automatically driven into the carriage of the accounting machine in the same manner as previously described with respect to FIG. 17, so that the magnetic strip on the leftmost margin thereof passes directly over the magnetic recording-reproducing head assembly 343, shown in FIG. 19. As the leading edge of the ledger card passes over photocell PCC (FIG. 17), the radiant energy from the light source **373** is reflected from the ledger card onto the active area thereof, so that photocell PCC is energized thereby. As previously described with respect to FIG. 83, line C is rendered TRUE as a result of photocell PCC being energized. With reference now to FIG. 85, when line C is thus rendered TRUE, all of the inputs to gate **1804** are simultaneously TRUE for the first time. Consequently, the lowermost one of the reference inputs to each of flipflops **6144** and **6145** is rendered FALSE, so that the flipflops are thereafter properly conditioned to be responsive to negative-going impulses which are respectively derived by data head **348** and clock head **347** from the information previously stored in the data and clock channels, respectively, on the magnetic strip of the ledger card, flipflop **6144** being utilized for D.C. restoration of impulses derived from the data channel, and flipflop **6145** being utilized for D.C. restoration and frequency doubling of the impulses derived from the clock channel. Thus, when the information zone of the magnetic strip portion of the ledger card first passes over the magnetic heads, clock and data impulses are derived, thereafter amplified, D.C. restored, and then essentially fed to other portions of the computer circuitry by way of lines CCR and CSA, respectively.

For example, as the state line REC is FALSE, the train of clock impulses derived by clock-head **347** and having an alternating polarity at a repetition rate of approximately 200 cycles per second appear amplified and inverted at the output of transistor gate **1805** in the same manner as previously described with respect to FIG. 48*i*. In response to each of the clock impulses, the state of flipflop **6145** is reversed, and line CCR is rendered TRUE. However, approximately 1240 microseconds later, the state of flip-flop **6145** reverts back to its initial state, so that line CCR is rendered FALSE thereby, all of which is fully illustrated in FIG. 87D.

With reference now to FIG. 91A, at TIME–**1** after a clock impulse is derived and line CCR is rendered TRUE by a reversal of state of flipflop **6145** (FIG. 85), the state of flip-flop **6087** is reversed, so that line GC is thereby rendered TRUE, line (GC)′, of course, being simultaneously rendered FALSE. When line (GC)′ experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6088** is reversed, so that line JHR is rendered TRUE thereby. Forty microseconds later at the following TIME–**1** after line JHR is rendered TRUE, the state of flipflop **6088** is again reversed, and line JHR is rendered FALSE. Thus, it is evident, each time a clock impulse is derived from the clock channel by the clock pick-up head, line JHR is rendered TRUE for one bit time period of forty microseconds. As illustrated in FIG. 87D, the state of line CSA is rendered TRUE by flipflop **6144** (FIG. 85) whenever the impulse derived by the data pick-up head is indicative of a binary "1," and is rendered FALSE whenever the data impulse is indicative of a binary "0."

With reference now to FIG. 91B, at TIME–**3** after line JHR is first rendered TRUE, line PJϕ experiences a TRUE-to-FALSE reversal of state and thereby causes the "J" digit-register (FIG. 71) to be preset to zero. Also at TIME–**3** after line JHR is rendered TRUE, the state of line CYC is reversed from TRUE to FALSE. As a result, the bit-counter (FIG. 62) is incremented from a binary count of "*d*" to a binary count of "*a*."

If it is assumed that the first decimal digit to be magnetically picked up from the ledger card by the data head is a "9"—i.e., binary 1001, as illustrated in FIG. 87D—line CSA is rendered TRUE at the first TIME–**1**, indicating that the binary value of bit "*a*" of the digit is a "1." Thus, with reference back to FIG. 91B, as line CSA is now TRUE, indicating that bit "*a*" of the digit is a binary "1," the state of line JL is reversed from TRUE to FALSE at TIME–**4** after line JHR is rendered TRUE. Therefore, with reference to FIG. 71, as the bit-counter is now at a binary count of "*a*," so that line B*a* is TRUE, the state of flipflop **6070** is reversed when line JL is thus rendered FALSE, and, as a result, line J*a* is thereby rendered TRUE, indicating that bit "*a*" of the digit stored in the "J" digit-register is a binary "1." As bits "*b*" and "*c*" of the first digit to be read from the ledger card both have a value of "0," as illustrated in FIG. 87D, the state of line CSA (FIG. 85) is rendered FALSE by a reversal of state of flipflop **6144** as a result of the negative-going impulse derived by the data head, and remains FALSE for two bit time periods until the next positive-going impulse is derived indicative of the binary value of bit "*d*."

As line CSA is now FALSE, indicating that the binary value of bit "*b*" is "0," line JL (FIG. 91B) is likewise rendered FALSE. At TIME–**2** after line JHR is rendered TRUE for the second time by the second clock impulse derived by the clock head, the bit-counter is advanced from a binary count of "*d*" to a binary count of "*c*" due to a change of state of line CYC. Therefore, with reference back to FIG. 71, as line JL remains FALSE before and after line B*b* from the bit-counter is rendered TRUE, the state of flipflop **6069** remains unchanged, so that the state of line J*b* is FALSE, indicating that the binary value of bit "*b*" of the digit stored in the "J" digit-register is a "0." The above-described sequence of events is sequentially repeated until lines J*a* and J*d* of the "J" digit-register are each rendered TRUE, lines J*b* and J*c* remaining FALSE. Thus, the states of lines J*a* through J*d* collectively represent the binary value of the digit just read from the ledger card—i.e., "1001."

With reference back to FIG. 91B, after the first-order digit is magnetically read from the ledger card and is stored in the "J" digit-register via line JL, the digit-counter being at a count of "9" and the bit-counter being at a count of "*d*," the state of flipflop **6082** is reversed and line CM rendered TRUE at the first TIME–**1** after line JHR is rendered TRUE for the fifth time. When line CM is thus rendered TRUE, line **TS3** is likewise rendered TRUE. Consequently, in response to lines CM and **TS3** being rendered TRUE, a "CM" word-cycle is thereafter executed, whereby the address in memory specified by the contents of section 3 of the instruction register is cleared by the writing of all zeros therein. Upon completion of the just-initiated "CM" word-cycle, line AN experiences a TRUE-to-FALSE reversal of state, so that the state of flipflop **6082** is reversed and line CM is thereby rendered FALSE. With reference to FIG. 91A, when line CM experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6096** is reversed, so that line (OBN)′ is thereby rendered FALSE.

It is to be noted at this point that the specified address in memory is not cleared until it has been determined that the ledger card just fed into the computer is not a new card; i.e., a card having no information magnetically recorded thereon. As a result of the "J" digit-register being loaded with a digit having a value other than zero, it is thereby determined that at least one decimal digit is magnetically recorded on the ledger card. Also, by clearing the particular address in memory before a digit is stored therein, the computer is immediately permitted to proceed with respect to the next successive address in memory whenever an end-of-word notation is read from the ledger card, an end-of-word notation (i.e., binary 1111) indicating that the remainder of the high-order digits of the word are zeros. Thus, as the remainder of the high-order digital positions in the particular address in memory are already preset to zero, the digit-counter is thus permitted to return to a count of "9" when an end-of-word notation is detected, without the need of executing successive read-write cycles with respect to each of the remaining digits of the word having a value of zero.

With reference to FIG. 91A, when line CM is thus rendered FALSE, the state of flipflop **6061** is reversed, so that line TH is thereby rendered TRUE. At TIME–1 after line TH is thus rendered TRUE, the state of flipflop **6104** is reversed, so that line SES is rendered TRUE thereby. Also, with reference to FIG. 91B, at TIME–3 after line TH is thus rendered TRUE, line AD experiences a TRUE-to-FALSE reversal of state and thereby causes the digit-counter to be incremented from a count of "9" to a count of "0." In addition, due to the fact that the digit now stored in the "J" digit-register is other than an end-of-word notation, so that line (EOW)′ is TRUE, line JM is also rendered TRUE when line TH is thus rendered TRUE. When line JM is rendered TRUE, line TS3 is likewise rendered TRUE. Consequently, in response to lines JM and TS3 thus being rendered TRUE and the digit-counter being at a count of "0," a "JM" digit-cycle is thereafter executed, whereby the digit just stored in the "J" digit-register (i.e., "9") is stored in the first-order digital position of the address in memory specified by the contents of section 3 of the instruction register.

In the present example, as the first digit read from the ledger card and stored in memory is not an end-of-word notation, and as there are therefore other digits of the word magnetically recorded on the card, the "J" digit-register is again preset to "0" by a TRUE-to-FALSE reversal of state of line PJφ at TIME–3 after line JHR is again rendered TRUE. Thereafter, the bit-counter is successively incremented from a count of "*d*" through counts "*a*," "*b*," and "*c*" and back to a count of "*d*," during which time the second digit magnetically read from the ledger card is stored in the "J" digit-register via line JL in exactly the same manner as just described. Thereafter, the digit-counter is advanced from a count of "0" to a count of "1," and the digit thus stored in the "J" digit-register is thereafter stored in the second-order digital position of the selected address in memory.

The above-described sequence of events is sequentially repeated until each of the digits of the first word is read from the ledger card and thereafter stored in the selected address in memory. With reference to FIG. 91A, it is noted that line TH is rendered FALSE at each TIME–1 by a reversal of state of flipflop **6061** after each "JM" read-write cycle is completed, as indicated by line (CFF)′ being rendered TRUE. Thus, when line TH is thereby rendered FALSE, the state of flipflop **6104** is reversed and line SES thereby rendered TRUE, whenever either the digit-counter is at a count of "9," as indicated by line D9 being rendered TRUE, or an end-of-word notation is detected, as indicated by line EOW being rendered TRUE in the manner previously described with respect to FIG. 71. If an end-of-word notation is not detected, the state of flipflop **6104** is again reversed, and line SES is thereby rendered FALSE at the following TIME–1 after the digit-counter reaches a count of "9." As shown in FIG. 91B, after the digit-counter reaches a count of "9," line IN4 is reversed from a TRUE state to a FALSE state when line SES is rendered FALSE; as a result, section 3 of the instruction register is incremented by a count of "1" in the manner previously described.

It is therefore evident that line SES is rendered TRUE after each word read from the ledger card is stored in the selected address in memory. Consequently, when line SES is subsequently rendered FALSE, it is necessary to ascertain whether or not there is an additional word to be magnetically read from the ledger card and subsequently stored in memory. This is accomplished by the comparing of the contents of sections 3 and 4 of the instruction register for equality in the manner previously described with respect to FIG. 59. As before described, if equality exists between sections 3 and 4 of the instruction register, line (I34)′ is rendered TRUE; otherwise, line (I34)′ remains FALSE. If there is inequality between the contents of sections 3 and 4 of the instruction register, the first-order digit of the next word is magnetically read from the ledger card and stored in the "J" digit-register, as before, a "CM" word-cycle is again initiated, and the above-described sequence of events is again repeated. If, however, there is equality between the contents of sections 3 and 4 of the instruction register, the state of flipflop **6098** (FIG. 91A) is reversed when line SES is rendered FALSE, and, as a result, line ROS is thereby rendered TRUE.

After the magnetic strip of the ledger card has been completely scanned by the pick-up heads, the card continues to be driven into the carriage of the machine, with the forward edge thereof disposed furthermost in the chute **291** in the manner previously described with respect to FIG. 17. However, when the reversal spot (indicated as **376** in FIG. 88A), located in the top margin of the ledger card, passes over the junction of the members **371** and **372** (FIG. 17), the radiant energy from the light source **373** is absorbed by the dark area thereof to the extent that the photocell PCB is thus de-energized. With reference to FIG. 83, when photocell PCB is de-energized, the state of line B is rendered FALSE thereby. When line B is thus rendered FALSE, the state of flipflop **6122** is reversed, so that line P is rendered TRUE, line (P)′, of course, being simultaneously rendered FALSE.

When line (P)′ is thus rendered FALSE, card-drive relay coil CDR is de-energized, and, as a result, relay contacts CDR—**5** are closed and thereby render line (U)′ TRUE. With reference to FIG. 82, also when coil CDR is de-energized, relay contacts CDR—**1** and CDR—**2** are opened, whereas relay contacts CDR—**3** and CDR—**4** are closed, as shown, thus causing a reversal of subsequently-applied current through the armature of motor CDM and a subsequent reversal of rotation thereof. However, when line (P)′ is rendered FALSE, card-drive motor CDM is de-energized by the output of gate **1770** simultaneously being rendered FALSE. With reference back to FIG. 83, when line (P)′ is thus rendered FALSE, the state of flipflop **6123** is reversed, so that card-drive brake solenoid CDB is energized. Consequently, when solenoid CDB is energized, the ledger card is brought to a sudden standstill in the manner previously described with respect to FIG. 14. However, approximately 120 milliseconds after solenoid CDB is energized, flipflop **6123** reverts back to its initial state, so that solenoid CDB is thereby de-energized.

With reference to FIG. 82, when the state of flipflop **6122** (FIG. 83) is reversed, so that line P is rendered TRUE, line ECC is also rendered TRUE, thus indicating that the ledger card has been read. After the ledger card has been read, the card is first reversed and thereafter ejected from the carriage, or is thereafter properly positioned in the carriage for a posting operation thereon, depending upon the value of the low-order digit in section 2 of the instruction register. For example, with reference to FIG. 91B, if it is assumed that the low-order digit in section 2 of the instruction register is an "8," indicating that the ledger card is to be ejected, line (I6*d*)′ is therefore FALSE, and, as a result, the state of flipflop **6094** remains unchanged, so that line MOU, hence line RED, both remain TRUE. Therefore, with reference to FIG. 82, when line (U)′ is rendered TRUE due to the card-drive relay coil being de-energized, armature current is applied to card-drive motor CDM via gate **1771**, so that the direction of rotation thereof is reversed, thus driving the ledger card outwardly from the carriage.

When the reversal spot at the top margin of the ledger card leaves the junction of the leg members **371** and **372** (FIG. 17), photocell PCB is again energized, and line

B (FIG. 83) is again rendered TRUE. Also, when the trailing edge of the ledger card passes over the aperture of the paper guide **303** on its way outwardly from the carriage, photocell PCC is de-energized, and line C is thus rendered FALSE. However, when the trailing edge of the ledger card passes over the junction of the leg members **371** and **372**, photocell PCB is again de-energized, and line B is again rendered FALSE.

Consequently, with reference to FIG. 83, as all of the inputs to logical AND **1775** are simultaneously TRUE after the ledger card is ejected from the carriage onto the loading table, paper-guide motor PGM is thus energized and thereafter effects unlatching of the compression rolls **296** from the card-drive platen **241** in the same manner as previously described with respect to FIG. 19. After the unlatching operation is completed, movable arm **336—339** (FIG. 83) is essentially deflected downwardly, as viewed, so that line E is rendered TRUE and line N is rendered FALSE thereby. When the state of line N is thus rendered FALSE, the state of flipflop **6122** is reversed, so that card-drive relay coil CDR is again energized. As a result of relay coil CDR thus being energized, contacts CDR—**5** are opened and thereby render line (U)′ FALSE, and contacts CDR—**1** through CDR—**4** (FIG. 82) revert to their initial conditions, as shown. In response to line (U)′ thus being rendered FALSE, card-drive motor CDM (FIG. 82) and paper-guide motor PGM (FIG. 83) are both simultaneously de-energized.

When the state of flipflop **6122** (FIG. 83) is reversed, so that line P is rendered FALSE, the state of line EOC (FIG. 82) is likewise rendered FALSE thereby. With reference to FIG. 91B, when line EOC is rendered FALSE, indicating that the ledger card has been completely ejected from the carriage onto the loading table, the output of gate **1610** experiences a TRUE-to-FALSE reversal of state. As a result, the state of flipflop **6094** is reversed, and lines MOU and RED are both rendered FALSE. With reference to FIG. 81, enter-card lamp EC is extinguished when line RED is thus rendered FALSE, thus providing a visual indication that no other ledger cards are to be fed into the computer at the present time. It is to be noted, however, that, while line RED is TRUE, enter-card lamp EC is illuminated each time photocell PCC is de-energized and is extinguished each time photocell PCC is energized. With reference now to FIG. 82, as it has been assumed that the numeral "8" is stored in the low-order digital position of section 2 of the instruction register, the state of line **16***d*, hence line CLC, is therefore TRUE. Consequently, when line (RED)′ is rendered TRUE, carriage open-close solenoid Y is thus energized and thereafter effects closure of the accounting machine carriage in the same manner as previously described with respect to FIG. 38. During the carriage-closing operation, the lower compression rolls **288** (FIG. 17) are brought into engagement with the accounting machine platen **71**, so that the movable arm of the switch **545** (FIG. 82) is returned to the position shown. As a result, solenoid Y is de-energized thereby. After the carriage is closed, the movable arm of switch **333** is thereafter in the position shown, so that line (D)′ is thereby rendered TRUE and line D is rendered FALSE. With reference to FIG. 83, when line D is thus rendered FALSE, the state of flipflop **6124** is reversed, so that solenoid CHP is energized and thereafter effects unlocking of the accounting machine carriage in the manner previously described with respect to FIG. 29. However, approximately 40 milliseconds after the state of flipflop **6124** is initially reversed, its state reverts back to its initial condition, so that solenoid CHP is thereby de-energized. As shown in FIG. 91A, the state of flipflop **6118** is reversed and line WOA is rendered TRUE at TIME–3 after line (D)′ is rendered TRUE. When line (WOA)′ is thus rendered FALSE, the state of flipflop **6120** is reversed, so that line (RF)′ is thereby rendered FALSE. However, approximately 250 milliseconds thereafter, the state of flipflop **6120** reverts back to its initial condition, so that line (RF)′ is rendered TRUE.

With reference to FIG. 91C, when line WOA is rendered TRUE, a selected one of motor bar solenoids MBϕ, MB1, or MB2 is energized, depending upon the value of the high-order digit stored in section 2 of the instruction register, all in the same manner as previously described in detail. Consequently, a selected one of the upper, middle, or lower motor bars is automatically actuated and thereby initiates a cycle of operation of the accounting machine portion of the computer. With reference to FIG. 72, when the accounting machine begins to cycle, the movable arm of switch **540** is deflected to the left, as viewed, so that line (NT)′ is thereby rendered TRUE. There after, each time the timing rack is translated from one digital value position to a successively higher-order digital value position, the state of flipflop **6072** is reversed, so that line PCT experiences a TRUE-to-FALSE reversal of state, all of which has previously been described in detail with respect to the "EKW" instruction. Also, as previously described, a TRUE-to-FALSE reversal of state of line PCT effectively increments the F-counter by a count of "1."

With reference to FIG. 91A, after the F-counter reaches a count of "9," so that line F9 is TRUE, all the inputs to gate **1703** are thereafter TRUE. As previously described with respect to FIG. 61, at TIME–1 after line PC is rendered TRUE by a reversal of state of flipflop **6040**, line AN experiences a TRUE-to-FALSE reversal of state. Thus, with reference back to FIG. 91A, the state of flipflop **6188** is reversed and line WOA is rendered FALSE when line AN is rendered FALSE at TIME–1 after line F9 is rendered TRUE. When line WOA is thus rendered FALSE, the state of flipflop **6107** is reversed, so that line STD is rendered TRUE.

With reference to FIG. 91B, when line STD is thus rendered TRUE, line TS5 is likewise rendered TRUE. Thereafter, in response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is executed whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events initiated by the "ECW" instruction.

It is to be noted that, if the numeral "6" had initially been stored in the high-order digital position of section 2 of the instruction register indicating "no motor bar operation," line WOA (FIG. 91A), after being initially rendered TRUE, remains TRUE for a period of approximately 250 milliseconds and is thereafter rendered FALSE by a reversal of state of flipflop **6118** due to line AN subsequently being rendered FALSE after line (RF)′ is rendered TRUE by a reversal of state of flipflop **6120**. Thus, when line WOA is rendered FALSE in this instance, line H6 being TRUE, the state of flipflop **6107** is reversed, as before, so that line STD is thereby rendered TRUE.

It will now be assumed that the ledger card, initially fed into the card-handling portion of the computer, is to be properly positioned in the accounting machine carriage on the next posting line ready for a posting operation thereon, rather than being ejected as just described. In this instance, the carriage of the machine is opened; the lower compression rolls are thereafter disengaged from the accounting machine platen; and the ledger card is manually placed face down on the loading table and thereafter automatically driven into the accounting machine carriage in exactly the same manner as just described. After the data information is magnetically picked up from the magnetic strip portion of the ledger card and stored in the selected address or addresses in memory, the direction of rotation of the card-drive motor is again reversed, as before, and the ledger card is thereafter driven outwardly from the accounting machine carriage in exactly the same manner as just described. However, as a line-finding operation is now desired, so that the digit initially stored in the low-order digital position of section 2 of the instruction register is either one of the digits "1" through "3" and "5" through "7," the state of flip flop 6094 (FIG. 91B) is reversed at TIME–1 after line EOC is rendered TRUE, indicating that the ledger card is completely scanned, due to a TRUE-to-FALSE reversal of state of the output of gate **1611**. Consequently, the states of lines MOU and RED are simultaneously rendered FALSE just after the initial forward direction of movement of the ledger card is reversed.

With reference to FIG. 85, as the state of line (L)′ is normally TRUE, the state of output line Q is likewise rendered TRUE by gates **1802** and **1811** when line (RED)′ is rendered TRUE. Consequently, with reference to FIG. 82, due to the fact that line Q is rendered TRUE simultaneously with line *i* being rendered FALSE when line MOU is rendered FALSE, the card-drive motor CDM remains energized and continues to drive the ledger card outwardly from the accounting machine carriage. As previously described with respect to FIG. 17, when the reversal spot at the top of the ledger card leaves the junction of the arms **371** and **372**, the photocell PCB is energized by the light source **373**. Also, as previously described with respect to FIG. 83, line B is rendered TRUE when the photocell PCB is energized. With reference to FIG. 85, when line B is thus rendered TRUE, the lowermost reference input to flipflop **6146**, as viewed, is thereby rendered FALSE, and line M14 is simultaneously rendered TRUE. When the lowermost reference input to flipflop **6146** is thus rendered FALSE, the flipflop is thereafter properly conditioned to be responsive via line (Rϕ)′, to a predetermined one of the negative-going linefind impulses derived by linefind head **346** from the information previously stored in the linefind channel of the magnetic strip portion of the ledger card.

Consequently, each time a negative-going linefind impulse is derived by pick-up head **346** as the ledger card is being driven outwardly from the accounting machine carriage, the impulse appears amplified and inverted at the output of transistor gate **1806**. As a result, the state of output line LFA experiences a FALSE-to-TRUE-to-FALSE reversal of state each time a linefind discontinuity is magnetically sensed by pick-up head **346**.

It is to be appreciated at this point that, when an "ECW" subprogram is first initiated, the R-counter is preset to a count of "1," "2," or "3," depending upon whether the ledger card is to be respectively positioned on the first, second, or third posting line. That is, if the low-order digit in section 2 of the instruction register is a "1" or a "5," the R-counter is preset to a count of "1"; if the low-order digit in section 2 is a "2" or a "6," the R-counter is preset to a count of "2"; and if the low-order digit is a "3" or a "7," the R-counter is preset to a count of "3," all of which is clearly illustrated in FIG. 68.

As previously noted, a "0" in the low-order digital position of section 2 of the instruction register initiates a "enter-new-card-with-manual-resume-program" cycle of operation.

If it is assumed that the low-order digit in section 2 of the instruction register is a "7"—i.e., "stop on third linefind with automatic-resume-program"—the R-counter is preset to a count of "3," so that lines R*a* and R*b* (FIG. 68) are rendered TRUE, whereas lines R*c* and R*d* remain FALSE.

With reference to FIG. 91B, each time the state of line LFA reverses from TRUE to FALSE in response to a linefind discontinuity being detected, the state of line DR likewise reverses from TRUE to FALSE. Consequently, the R-counter is decremented each time a linefind discontinuity is detected. Therefore, when the third linefind discontinuity is detected, the R-counter is decremented from a count of "1" to a count of "0," so that line Rϕ (FIG. 68) is thereby rendered TRUE. With reference to FIG. 85, when the state of line (Rϕ), is rendered FALSE, the state of flipflop **6146** is reversed, so that line (L)′ is thereby rendered FALSE. As a result, line Q likewise is rendered FALSE. With reference to FIG. 82, as a result of line Q thus being rendered FALSE, the card-drive motor CDM is de-energized. Also, as shown in FIG. 83, when line (L)′ is rendered FALSE, the state of flipflop **6123** is reversed, so that the card-drive brake solenoid CDB is energized for a period of approximately 120 milliseconds and thereby causes the ledger card to be brought to a sudden standstill.

At this point, the ledger card is properly positioned with respect to the next posting line and is held tightly pressed against the card-drive platen **241** (FIG. 19) by means of the compression rollers **296** in order to maintain proper register thereof. In addition, the carriage is now open, so that additional forms may manually be inserted therein, if desired. With reference to FIG. 82, line EOC is likewise rendered FALSE when line (L)′ is rendered FALSE. Due to the fact that line CLC is rendered TRUE as a result of an "automatic" resume-program operation being specified by the value of the low-order digit of section 2 of the instruction register, carriage-close solenoid Y is energized when line (EOC)′ is thus rendered TRUE and thereafter causes the accounting machine carriage to be closed in the same manner as previously described.

It is to be noted, however, that, if a manual resume-program is specified by the value of the low-order digit of section 2 of the instruction register, line CLC is rendered TRUE upon subsequent depression of the resume-program bar, at which time line RPB is rendered TRUE.

During the carriage-closing operation, the lower compression rolls **288** (FIG. 17) are spring-urged into engagement with the accounting machine platen **71**, with the lower edge of the ledger card firmly secured therebetween, and the movable arm of the switch **545** (FIG. 82) is deflected to the position shown, so that line (G)′ is rendered TRUE and line G is rendered FALSE, thus causing solenoid Y to be de-energized. When line (G)′ is thus rendered TRUE, the paper-guide motor PGM is energized and thereafter effects unlatching of the compression rollers **296** (FIG. 19) from the card-handling platen **241**. After the unlatching operation is completed, the paper guide **282** is spring-urged to the position shown in FIG. 17, and the switch arm **336—339** (FIG. 82) is effectively deflected downwardly to the position shown, so that line E is rendered TRUE and line N is rendered FALSE thereby. Also, after the unlatching operation is completed, the plunger of switch **333** (FIG. 16) is released by the tail of member **329**, so that the movable arm of switch **333** (see FIG. 82) is deflected to the position shown, so that line (D)′ is rendered TRUE and line D is rendered FALSE thereby. When line D is thus rendered FALSE, the state of flipflop **6124** (FIG. 83) is reversed, and, as a result, carriage-home-unlock solenoid CHP is energized for a period of approximately 40 milliseconds and thereby allows the accounting machine carriage to thereafter be moved away from home position when a machine cycle is subsequently initiated.

When line N is thus rendered FALSE, the state of flipflop **6122** is reversed, so that line (P)′ is rendered TRUE and card-drive relay solenoid CDR is simultaneously energized thereby. As a result of solenoid CDR thus being energized, relay contacts CDR–5 are opened and thereby cause paper-guide motor PGM to be de-energized. Simultaneously therewith, relay contacts CDR–**1** and CDR–2 (FIG. 82) are closed, whereas relay contacts CDR–3 and CDR–4 are opened thereby. With reference to FIG. 91A, at the following TIME–3 after line (D)′ is rendered TRUE, the state of flipflop **6118** is reversed, so that line WOA is rendered TRUE thereby. As a result of line WOA thus being rendered TRUE, the particular motor bar depression indicated by the value of the high-order digit in section 2 of the instruction register is initiated in exactly the same manner as previously described in detail.

After a machine cycle is initiated, the state of flipflop **6107** is reversed in the same manner as previously described, so that line STD is thereby rendered TRUE. As shown in FIG. 91B, when line STD is thus rendered TRUE, line TS5 is likewise rendered TRUE. Consequently, in response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is thereafter executed whereby the contacts of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events initiated in accordance with an "ECW" subprogram.

74. Detailed Description of SHF Instruction

In executing a "shift" instruction, one of five possible subprograms is carried out, depending upon the value of the high and low order digits stored in section 2 of the instruction register. For example, if the number stored in section 2 of the instruction register is "00," a simple copy operation is carried out, whereby the word stored in memory at the address specified by the contents of section 3 of the instruction register is copied into the address in memory specified by the contents of section 4 of the instruction register; if the high-order digit stored in section 2 of the instruction register is a "2," the word stored in memory at the address specified by section 3 of the instruction register is shifted to the left by the number of digital positions specified by the value of the low-order digit stored in section 2 of the instruction register, and, thereafter, the result is stored in the address in memory specified by section 4 of the instruction register; if the high-order digit stored in section 2 of the instruction register is a "3," the word stored in memory at the address specified by section 3 of the instruction register is shifted to the right by the number of digital positions specified by the value of the low-order digit in section 2 and is rounded off to the nearest half-cent, and, thereafter, the result is stored in memory at the address specified by section 4 of the instruction register; if the high-order digit in section 2 of the instruction register is a "4," the word stored in memory at the address specified by section 3 of the instruction register is shifted to the right by the number of digital positions specified by the value of the low-order digit in section 2 of the instruction register while preserving its alegbraic sign, and, thereafter, the word is stored in memory at the address specified by section 4 of the instruction register; and, if the high-order digit in section 2 of the instruction register is a "5," the word stored in memory at the address specified by section 3 of the instruction register is shifted to the right by the number of digital positions specified by the value of the low-order digit in section 2 of the instruction register, without preserving its algebraic sign, and, thereafter, the word is stored in memory at the address specified by section 4 of the instruction register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107E, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing a "SHF" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents thus stored in section 1 of the instruction register is "04," carry out the subinstruction listed in Step 2 of the following SHF subprogram.
2	MA-3-3	Copy into address-A the word stored in memory at the address specified by section 3 of the instruction register, and, simultaneously therewith, copy the word digit by digit into the "J" digit-register; thereafter, go to Step 3.
3	OBM(4)-0-4	Preset the state of line OBM to correspond to the algebraic sign of the word just read from memory; thereafter, go to Step 4.
4	RLR-0-5	Preset the R-counter to a count corresponding to the low-order digit of section 2 of the instruction register; thereafter, go to Step 5.
5	Rϕ-6-18	If the R-counter is at a count of "0," go to Step 18; otherwise, go to Step 6.
6	H2-7-10	If the high-order digit in section 2 of the instruction register is a "2," go to Step 10; otherwise, go to Step 7.
7	H3-8-11	If the high-order digit in section 2 of the instruction register is a "3," go to Step 11; otherwise, go to Step 8.
8	H4-9-12	If the high-order digit in section 2 of the instruction register is a "4," go to Step 12; otherwise, go to Step 9.
9	H5-0-14	If the high-order digit in section 2 of the instruction register is a "5," go to Step 14.
10	SA(1)-0-16	Shift the word of address-A one place to the left; thereafter, go to Step 16.
11	R1-12-15	If the R-counter is at a count of "1," go to Step 15; otherwise, go to Step 12.
12	OBM(2)-13-14.	If the state of line OBM is TRUE, go to Step 14; otherwise, go to Step 13.
13	PK9-0-14	Preset the "K" digit-register to "9"; thereafter, go to Step 14.
14	SA(2)-0-16	Shift the word of address-A one place to the right; thereafter, go to Step 16.
15	ARO-0-12	Add the constant 0000000005 to the word stored in address-A; thereafter, go to Step 12.
16	SB(1)-0-17	Shift the word of address-B one place to the left; thereafter, go to Step 17.
17	DR-0-5	Decrement the R-counter; thereafter, go to Step 5.
18	AM-4-19	Copy the word of address-A into memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 19.
19	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "SHF" instruction (i.e., "04"), the state of line SHF is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "shift" instruction is thereafter executed in the following manner:

With reference to FIGS. 93A and 93B, there is logically illustrated therein a composite circuit diagram of particular portions of the computer circuitry utilized in executing a "SHF" instruction. With particular reference to FIG. 93B, at TIME-1 after the initiation of a "MI" word-cycle, the state of flipflop **6095** is reversed and line OBM is rendered FALSE thereby. Upon completion of the "MI" word-cycle, the state of flipflop **6091** is reversed due to a TRUE-to-FALSE reversal of state of line MIN, and, as a result, the state of line MA is thereby rendered TRUE. Due to the fact that line TS3 is also rendered TRUE when line MA is rendered TRUE, a "MA" word-cycle is thereafter executed whereby the word stored in memory at the address specified by section 3 of the instruction register is read out and stored in address-A. Simultaneously therewith, the word from memory is copied digit by digit into the "J" digit-register via gate **1413** (FIG. 71), all in the manner previously described. Consequently, after the "MA" word-cycle is terminated, the "J" digit-register is storing the tenth-order digit of the word just read from memory. Upon completion of the "MA" word-cycle, the state of flipflop **6091** is again reversed when line AN experiences a TRUE-to-FALSE reversal of state, and, as a result, line MA is thereby rendered FALSE.

If the tenth-order digit thus stored in the "J" digit-register has a value other than "9," so that the line (J9)′ is TRUE, indicating that the algebraic sign of the word just read from memory is positive, the state of flipflop **6095** is reversed upon completion of the previously-initiated "MA" word-cycle, the R-counter having previously been reset to zero via a TRUE-to-FALSE reversal of state of line PRϕ when line STD is rendered FALSE at the beginning of the "shift" subprogram. Consequently, line OBM is rendered TRUE thereby. If, however, the tenth-order digit thus stored in the "J" digit-register has a value of "9," so that line (J9)' is FALSE, indicating that the algebraic sign of the word is negative, the state of flipflop **6095** remains unchanged, so that line OBM remains FALSE. It is evident, therefore, that line OBM is rendered TRUE if the algebraic sign of the word read from memory is positive and is rendered FALSE if the algebraic sign of the word read from memory is negative.

Also, when the previously-initiated "MA" word-cycle is completed to the extent that line EG is rendered FALSE, the state of line RLR is likewise reversed from TRUE to FALSE. As a result of the state of line RLR being reversed from TRUE to FALSE, the low-order digit of section 2 of the instruction register is stored in the R-counter in the same manner as previously described with respect to FIG. 68. Thus, if the R-counter is thereafter storing a "0," indicating a "no shift" or copy operation, the state of flipflop **6077** is reversed when line AN is subsequently rendered FALSE and line AM is thereby rendered TRUE. When line AM is rendered TRUE, line TS4 is likewise rendered TRUE. Consequently, in response to lines AM and TS4 thus being rendered TRUE, an "AM" word-cycle is thereafter executed whereby the word stored in address-A is copied into the memory address specified by the contents of section 4 of the instruction register. If, however, the digit thus stored in the R-counter is other than "0," the word in address-A is to be shifted by the number of digital positions corresponding to the value of the digit stored in the R-counter.

With reference to FIG. 93A, if it is assumed that the word in address-A is to be shifted to the left, as indicated by the state of line H2 being TRUE, the state of flipflop **6101** is reversed when line AN is rendered FALSE, and, as a result, line SA is thus rendered TRUE. As previously described, when line SA is rendered TRUE, line DBD (FIG. 93B) remaining "FALSE," a "SA" subcommand is thereafter executed whereby the word in address-A is shifted to the left by one digital position. After the shifting operation is completed, the state of flipflop **6101** is reversed, and line SA is rendered FALSE thereby. When line SA is thus rendered FALSE, the state of flipflop **6103** is reversed, so that line SB is rendered TRUE thereby. With reference to FIG. 93B, just prior to line SA being rendered FALSE, line DR is rendered FALSE upon occurrence of the next succeeding TRUE-to-FALSE reversal of state of line AN and thereby causes the R-counter (FIG. 68) to be decremented by a count of "1." If, after being decremented, the R-counter is at a count other than "0," a second "SA" subcommand is thereafter initiated by a reversal of state of flipflop **6101** (FIG. 93A), whereby the word in address-A is again shifted to the left by one digital position. Thereafter, the R-counter is again decremented, and a subsequent "SA" subcommand is initiated if the R-counter, after being decremented for the second time, is still at a count other than "0." However, if, after being decremented, the R-counter is at a count of "0," so that line (Rϕ)' is rendered FALSE, a subsequently-initiated "SA" subcommand is thereby prevented.

With reference to FIG. 93B, if the word originally stored in address-A is to be shifted to the right instead of to the left, so that the state of a selected one of lines H3 through H5 is TRUE instead of line H2, line DBD is rendered TRUE each time line SA is rendered TRUE. Consequently, each time a "SA" subcommand is initiated, the word in address-A is shifted to the right by one digital position instead of to the left, in exactly the same manner as previously described in detail. However, it is again to be noted that a TRUE state of line H4 initiates a sequence of "SA" subcommands whereby the digit "9" is copied into the tenth-order digital position each time a negative word is shifted to the right by one digital position, thus preserving the algebraic sign of the word; a TRUE state of line H5, however, initiates a sequence of "SA" shift right subcommands whereby the algebraic sign of the word is not preserved; and a TRUE state of line H3 initiates a sequence of "SA" shift right subcommands whereby the word is first shifted to the right by one less than the number of shifts specified while preserving its algebraic sign, and, thereafter, the constant 0000000005 is added thereto. Thereafter, the word is shifted by one more digital position. For example, if it is assumed that it is desired for the word to be shifted to the right by two digital positions and then rounded to the nearest half-cent (i.e., the contents of section 2 of the instruction register being "32"), the sequence of events for carrying out this particular shift instruction is as follows:

Upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6097** (FIG. 93B) is reversed, and line MA is thereby rendered TRUE. Line MA, being rendered TRUE, initiates a "MA" word-cycle whereby the word stored in memory at the address specified by section 3 of the instruction register is read out and stored in address-A. Just prior to the termination of the "MA" word-cycle, the low-order digit of section 2 of the instruction register—i.e., "2"—is stored in the R-counter. Thus, with reference to FIG. 93A, the output of gate **1671** experiences a TRUE-to-FALSE reversal of state upon completion of the previously-initiated "MA" word-cycle. Consequently, the state of flipflop **6101** is reversed, and line SA is rendered TRUE, so that a "SA" subcommand is thereafter executed, whereby the word in address-A is shifted to the right by one digital position. After the shifting operation is completed, the state of flipflop **6101** is reversed, and line SA is thereby rendered FALSE. When line SA is thus rendered FALSE, the state of flipflop **6103** is reversed, so that line SB is thereby rendered TRUE. With reference to FIG. 93B, also when line SA is thus rendered FALSE, line DR is likewise rendered FALSE. Consequently, the R-counter is decremented from a count of "2" to a count of "1." As the R-counter is now at a count of "1," so that line R1 is TRUE, as the high-order digit in section 2 of the instruction register is a "3," so that line H3 is TRUE, and as line SB has just been rendered TRUE upon completion of the "SA" subcommand, the state of flipflop **6078** is reversed upon occurrence of the next TRUE-to-FALSE reversal of state of line AN. As a result, line ARO is thereby rendered TRUE. As previously described, in response to line ARO thus being rendered TRUE, an "ARO" subcommand is thereafter executed whereby the constant 0000000005 is added to the word in address-A. Upon completion of the "ARO" subcommand, the state of flipflop **6078** is reversed by a TRUE-to-FALSE reversal of state of line AN, and line ARO is thus rendered FALSE. With reference to FIG. 93A, when line ARO is thus rendered FALSE, the state of flipflop **6101** is again reversed for the second time, so that a second "SA" subcommand is thereafter executed, whereby the word in address-A is again shifted to the right by one digital position. Upon completion of the second "SA" subcommand, the state of flipflop **6101** is again reversed by a TRUE-to-FALSE reversal of state of line AN, and line SA is thereby rendered FALSE.

With reference to FIG. 93B, just prior to line SA thus being rendered FALSE, line DR is rendered FALSE upon occurrence of the next successive TRUE-to-FALSE reversal of state of line AN. As a result, the R-counter is decremented from a count of "1" to a count of "0." With reference back to FIG. 93A, as the R-counter is now at a count of "0," so that line (Rϕ)' is thus rendered FALSE, subsequently-initiated "SA" subcommands are thereby prevented. With reference now to FIG. 93B, upon occurrence of a subsequent TRUE-to-FALSE reversal of state of line AN after line Rϕ is rendered TRUE, the state of flipflop **6077** is reversed, and line AM is thereby rendered TRUE. When line AM is rendered TRUE, line TS4 is likewise rendered TRUE.

Consequently, in response to lines AM and TS4 thus being rendered TRUE, an "AM" word-cycle is thereafter executed whereby the word in address-A is stored in memory at the address specified by the contents of section 4 of the instruction register. Upon completion of the "AM" word-cycle, so that line AM is rendered FALSE by a reversal of state of flipflop **6077**, the state of flipflop **6107** is reversed, so that line STD is thereby rendered TRUE. When line STD is rendered TRUE, line TS5 is likewise rendered TRUE. Consequently, a "STD" subinstruction is thereafter executed whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events executed in accordance with a "SHF" subprogram.

75. *Detailed Description of CMA Instruction*

In the execution of a "CMA" instruction, words of all zeros are stored in a predetermined number of addresses in memory, the first address in memory being specified by the contents of section 3 of the instruction register and the last address in the string of addresses being specified by the contents of section 4 of the instruction register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107F, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "CMA" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "05," carry out Step 2 of the following "CMA" subprogram.
2	CM-3-3	Clear the address in memory specified by the contents of section 3 of the instruction register by writing all zeros therein; thereafter, go to Step 3.
3	I34-4-5	If the contents of sections 3 and 4 of the instruction register are equal, go to Step 5; otherwise, go to Step 4.
4	IN4-0-2	Increment section 3 of the instruction register; thereafter, go to Step 2.
5	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word cycle, during which time the next instruction word stored in memory at the address indicated by the contents of the word-selecting register is read out and stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "CMA" instruction (i.e., "05"), the state of line CMA is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "CMA" instruction is thereafter executed in the following manner:

At TIME–**4** after the "MI" word-cycle is substantially completed, line AN (FIG. 61) is again rendered TRUE and ten microseconds later at TIME–**1** is again rendered FALSE. With reference back to FIG. 76, when line AN is rendered FALSE upon completion of the "MI" word-cycle operation, the state of flipflop **6092** is again reversed, and line MI is rendered FALSE thereby, thus terminating the "MI" word-cycle operation. As previously mentioned with respect to FIG. 72, as long as both of lines MI and AN are TRUE, line MIN is TRUE. Consequently, when line AN is thus rendered FALSE, line MIN is likewise rendered FALSE.

Reference is now made to FIG. 94, wherein there is logically illustrated a composite circuit diagram of various portions of the computer circuitry utilized in executing a "CMA" instruction. As shown, when line MIN experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6032** is reversed, and line CM is thereby rendered TRUE. When line CM is thus rendered TRUE, line TS3 is likewise rendered TRUE. Consequently, a "CM" word-cycle is thereafter executed whereby a word of all zeros is stored in the address in memory specified by section 3 of the instruction register in exactly the same manner as previously described. As previously described with respect to FIG. 59, if the contents of sections 3 and 4 of the instruction register are of equal magnitude, line (**134**)′ is rendered TRUE; otherwise, line (**134**)′ is rendered FALSE. Thus, with reference back to FIG. 94, if it is assumed that line (**134**)′ is FALSE, indicating that a successively higher-order address in memory is also to be cleared, the state of line CM remains TRUE. When line AN subsequently experiences a TRUE-to-FALSE reversal of state, line IN4 is rendered FALSE thereby, and, as a result, section 3 of the instruction register is incremented by a count of "1." Thereafter, a second "CM" word-cycle is executed, whereby a word of all zeros is stored in memory at the address now specified by section 3 of the instruction register. If it is now assumed that the contents of sections 3 and 4 of the instruction register are of equal magnitude, so that line (I**34**)′ is rendered TRUE, the state of flipflop **6982** is reversed by a subsequent TRUE-to-FALSE reversal of state of line AN, and, as a result, line CM is rendered FALSE thereby. When line CM is thus rendered FALSE, the state of flipflop **6107** is reversed, and lines STD and TS5 are both rendered TRUE. Consequently, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the "CMA" subprogram.

76. *Detailed Description of MB Instructions*

A "motor bar" instruction simply initiates selective depression of a predetermined one of motor bars **23**, **27**, or **28** (FIGS. 2 and 38) for either a "touch" or a "hold" operation and thereafter immediately proceeds to the next instruction. As previously described, the particular type of motor bar depression is determined by the value of the high-order digit of section 2 of the instruction register. For example, if the high-order digit of section 2 of the instruction register is a "0," upper motor bar **23** is actuated for a touch operation; if the value of the high-order digit is a "1," upper motor bar **23** is actuated for a hold operation; if the high-order digit is a "2," middle motor bar **27** is actuated for a touch operation; if the high-order digit is a "3," middle motor bar **27** is actuated for a hold operation; if the high-order digit is a "4," lower motor bar **28** is actuated for a touch operation; and, if the high-order digit is a "5," lower motor bar **28** is actuated for a hold operation. Also, as previously described, the value of the low-order digit of section 2 of the instruction register determines the position of the decimal point when the word is subsequently printed out. For example, if the low-order digit is a "0," a normal decimal point is printed between the second and third order digits of the word printed out; if the low-order digit is a "1," a decimal point in the form of a "comma" is printed between the fifth and sixth order digits of the word printed out; if the low-order digit is a "2," a comma is printed between the eighth and ninth order digits of the word printed out.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107G, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "MB" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "06," go to Step 2 of the following "MB" subprogram.
2	PFϕ-0-3	Preset the F-counter to "0"; thereafter go to Step 3.
3	GO-0-4	Go to Step 4 whenever depression of a motor bar is capable of initiating a cycle of operation of the accounting machine.
4	Hϕ-6-5	If the high-order digit of section 2 of the instruction register is a "0," go to Step 5; otherwise, go to Step 6.
5	MBϕ(1)-0-16	Depress the upper motor bar for a touch operation; thereafter, go to Step 16.
6	M1-8-7	If the high-order digit of section 2 of the instruction register is "1," go to Step 7; otherwise, go to Step 8.
7	MBϕ(2)-0-16	Depress the upper motor bar for a hold operation; thereafter, go to Step 16.
8	H2-10-9	If the high-order digit of section 2 of the instruction register is a "2," go to Step 9; otherwise, go to Step 10.
9	MB1(1)-0-16	Depress the middle motor bar for a touch operation; thereafter, go to Step 16.
10	H3-12-11	If the high-order digit of section 2 of the instruction register is a "3," go to Step 11; otherwise, go to Step 12.
11	MB1(2)-0-16	Depress the middle motor bar for a hold operation; thereafter, go to Step 16.
12	H4-14-13	If the high-order digit of section 2 of the instruction register is a "4," go to Step 13; otherwise, go to Step 14.
13	MB2(1)-0-16	Depress the lower motor bar for a touch operation; thereafter, go to Step 16.
14	H5-0-15	If the high-order digit of section 2 of the instruction register is a "5," go to Step 15.
15	MB(2)-0-16	Depress the lower motor bar for a hold operation; thereafter, go to Step 16.
16	L1-18-17	If the low-order digit of section 2 of the instruction register is a "1," go to Step 17; otherwise, go to Step 18.
17	OH1-0-20	When printing occurs, print a comma between the fifth and sixth order digits and, if there are no higher-order significant digits, print a zero preceding the comma. Thereafter, go to Step 20.
18	L2-20-19	If the low-order digit of section 2 of the instruction register is a "2," go to Step 19; otherwise, go to Step 20.
19	OH2-0-20	When printing occurs, place a comma between the eighth and ninth order digits, and, if there are no higher-order significant digits, print a "0" preceding the comma. Thereafter, go to Step 20.
20	PCT-0-21	When the amount racks are traveling in a "setting" direction, go to Step 21 each time the timing rack is translated to a different digital valued position and also when the printing aligner is engaged.
21	AF-0-22	Increment the F-counter; thereafter, go to Step 22.
22	F9-20-23	If the F-counter is at a count of "9," go to Step 23; otherwise, go to Step 20;
23	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "MB" instruction (i.e., "06"), the state of line MB is rendered TRUE in a manner previously described with respect to FIG. 56, and, as a result, a "motor bar" instruction is thereafter executed in the following manner:

With reference to FIG. 95, there is logically illustrated therein a composite circuit diagram of particular portions of the computer circuitry utilized in executing a "MB" instruction. As shown therein, upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6089** is reversed when line MIN experiences a TRUE-to-FALSE reversal of state, and, as a result, the state of line KEY is rendered TRUE thereby. Also when line MIN is initially rendered TRUE, line PFϕ experiences a TRUE-to-FALSE reversal of state, thus causing the F-counter (FIG. 72) to be preset to "zero." As previously described in detail, if it is assumed that the accounting machine is in a condition such that actuation of a motor bar is capable of initiating a cycle of operation thereof, line GO is rendered TRUE, indicating that a cycle of operation of the accounting machine is permitted to be initiated at this time. It being assumed that line GO is TRUE, the state of flipflop **6118** is reversed at TIME-3 after line KEY is rendered TRUE and line WOA is thereby rendered TRUE. When line WOA is rendered TRUE, a selected one of motor bar solenoids MBϕ, MB1, or MB2 is energized, depending upon the value of the high-order digit of section 2 of the instruction register. For example, if the high-order digit is either one of the digits "0" or "1," motor bar solenoid MBϕ is energized; if the high-order digit is either one of the digits "2" or "3," motor bar solenoid MB1 is energized, and, if the high-order digit is either one of the digits "4" or "5," motor bar solenoid MB2 is energized thereby. As a result of a selected one of the upper, middle, or lower motor bar solenoids being energized, a cycle of operation of the accounting machine is thus initiated thereby. With reference to FIG. 72, when the accounting machine begins to cycle, the movable arm of switch **540** is deflected to the left, as viewed, so that line (NT)′ is thereby rendered TRUE. Thereafter, each time the accounting machine timing rack is translated from one digital valued position to a successively high-order digital valued position, the state of flipflop **6072** is reversed, so that line PCT experiences a TRUE-to-FALSE reversal of state in the same manner as previously described. Also as previously described, a TRUE-to-FALSE reversal of state of line PCT effectively increments the F-counter by a count of "1."

With reference back to FIG. 95, after the F-counter reaches a count of "9," so that line F9 is rendered TRUE, the state of flipflop **6118** is reversed upon occurrence of the following TRUE-to-FALSE reversal of state of line AN, and, as a result, line WOA is rendered FALSE and thereby causes any previously-energized motor bar solenoid to be de-energized.

When the F-counter first reaches a count of "8," so that line F8 is rendered TRUE, order-hook solenoid OH1 is energized if the low-ordered digit in section 2 of the instruction register is a "1," whereas order-hook solenoid OH2 is energized if the low-order digit in section 2 is a "2." As previously described with respect to FIG. 35, when order-hook solenoid OH1 is energized, all order-hooks from the third to and including the eighth one are unlatched from their respective type sectors. However, when order-hook solenoid OH2 is energized, all order-hooks from the third to and including the eleventh one are unlatched from their respective type sectors.

Also as shown, upper motor bar solenoid MBϕ is de-energized when the F-counter is advanced from a count of "0" to a count of "1," unless the high-order digit in section 2 of the instruction register is a "1"; middle motor bar solenoid MB1 is de-energized when the F-counter is advanced from a count of "0" to a count of "1," unless the high-order digit in section 2 of the instruction register is a "3"; and lower motor bar solenoid MB2 is de-energized when the F-counter is advanced from a count of "0" to a count of "1," unless the high-order digit is a "5." It is evident, therefore, that either a touch or a hold operation of a particular motor bar is accomplished simply by maintaining the corresponding motor bar solenoid energized for a predetermined period of time. However, as previously stated, when line WOA is subsequently rendered FALSE, any previously-energized motor bar solenoid is de-energized thereby. After the F-counter reaches a count of "9," so that line F9 is rendered TRUE, the state of flipflop **6107** is reversed when line WOA is thus rendered FALSE and line STD is rendered TRUE thereby. When line STD is thus rendered TRUE, line TS5 is likewise rendered TRUE. Consequently, a "STD" subinstruction is thereafter executed whereby the contents of section 5 of the instruction register is copied into the word-selecting register, thus terminating the sequence of events executed in accordance with a "MB" subprogram.

77. Detailed Description of STP Instructions

The "stop" instruction is utilized simply to effect cessation of all computation and data-handling activity within the computer until the resume-program-bar is manually depressed.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107H, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "STP" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "07," go to Step 2 of the following subprogram.
2	RPB-0-3	Upon depression of the resume-program-bar, go to Step 3.
3	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "STP" instruction (i.e., "07"), the state of line STP is rendered TRUE in a manner previously described with respect to FIG. 56, and, as a result, a "stop" instruction is thereafter executed in the following manner:

With reference to FIG. 96, there is logically illustrated therein a composite circuit diagram of particular portions of the computer circuitry utilized in executing a "STP" instruction. As shown, when line STP is rendered TRUE, lamp HA is illuminated (see also FIGS. 1 and 2), thus giving a visual indication that depression of the resume program bar is necessary before any computation or data-handling activity will commence. As shown in FIG. 38, when resume-program-bar RPB is depressed, the movable contacts of switch **RP1**—**RP2** are actuated thereby. With reference now to FIG. 84, when the resume-program-bar is depressed, the normally-open contacts of switch **RP2** are closed, and the normally-closed contacts of switch **RP1** are opened thereby. At TIME–1 after the contacts of switch **RP2** are thus closed, the state of flip-flop **6138** is reversed, so that line RP is rendered TRUE thereby. However, after the resume-program-bar is released, so that the contacts of switches **RP1** and **RP2** resume their initial conditions, as shown, the state of flipflop **6138** is again reversed at TIME–1, so that line RP is rendered FALSE thereby. With reference to FIG. 78, when line RP experiences a TRUE-to-FALSE reversal of state, the state of flip-flop **6099** is reversed, and line RPB is thereby rendered TRUE. However, upon a subsequent TRUE-to-FALSE reversal of state of line AN, the state of flipflop **6099** is again reversed, so that line RPB is thereby rendered FALSE. With reference now to FIG. 96, when line RPB thus experiences a TRUE-to-FALSE reversal of state, the state of flipflop **6107** is reversed, and line STD is rendered TRUE thereby. When line STD is thus rendered TRUE, line TS5 is likewise rendered TRUE, and, as a result, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is copied into the word-selecting register, thus terminating the sequence of events executed in accordance with a "STP" subprogram.

78. Detailed Description of ADD Instructions

In executing an "add" instruction, any two words stored in memory addresses $\phi\phi$ through 99 are algebraically added together, and their sum is thereafter stored in any predetermined memory address.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107I, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing an "ADD" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "08," go to Step 2 of the following subprogram.
2	MA-2-3	Copy into address-A the word located in the address in memory specified by the contents of section 2 of the instruction register, and simultaneously copy the word, digit by digit, into the "J" digit-register; thereafter, go to Step 3.
3	RAD-3-4	Add to the word stored in address-A the word stored in the address in memory specified by the contents of section 3 of the instruction register, and thereafter store the sum back in address-A; thereafter, go to Step 4.
4	AM-4-5	Copy the word of address-A into the address in memory specified by the contents of section 4 of the instruction register; thereafter, go to Step 5.
5	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously-initiated "MI" word-cycle, during which time the next instruction word is read out from the address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for an "ADD" instruction (i.e., "08"), the state of line ADD is rendered TRUE in a manner previously described with respect to FIG. 56, and, as a result, an "add" instruction is thereafter executed in the following manner:

Reference is made to FIG. 97, wherein there is logically illustrated a composite circuit diagram of particular portions of the computer circuitry utilized in executing an "ADD" instruction. As previously described, upon completion of the previously-initiated "MI" word-cycle, the state of line MIN experiences a TRUE-to-FALSE reversal of state. When line MIN is thus rendered FALSE, the state of flip flop **6091** is reversed and line MA is rendered TRUE thereby. When line MA is thus rendered TRUE, line TS2 is likewise rendered TRUE. Consequently, a "MA" word-cycle is thereafter executed, whereby the word stored in memory at the address specified by section 2 of the instruction register is read out and stored in address-A. After the word is stored in address-A, line AN experiences a TRUE-to-FALSE reversal of state, so that the state of flip flop **6091** is again reversed and line MA is thereby rendered FALSE. When line MA is thus rendered FALSE, the state of flip flop **6097** is reversed, and line RAD is rendered TRUE thereby. When line RAD is thus rendered TRUE, line TS3 is likewise rendered TRUE. Consequently, a "RAD" subcommand is thereafter executed, whereby the word stored in memory at the address specified by section 3 of the instruction register is added to the word just stored in address-A. Upon completion of the addition operation, line AN experiences a TRUE-to-FALSE reversal of state, so that the state of flip flop **6097** is again reversed and line RAD is rendered FALSE thereby. As a result of line RAD thus being rendered FALSE, the state of flip flop **6077** is reversed, and line AM is thus rendered TRUE. When line AM is thus rendered TRUE, line TS4 is likewise rendered TRUE. Consequently, an "AM" word-cycle is thereafter executed.

whereby the sum now stored in address-A is copied into memory at the address specified by section 4 of the instruction register. After the sum is properly stored in the designated address in memory, line AM is rendered FALSE, so that the state of flip flop **6107** is reversed and line STD is rendered TRUE thereby. When line STD is thus rendered TRUE, line TS5 is likewise rendered TRUE. As a result, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events initiated by the "ADD" instruction.

79. *Detailed Description of SUB Instruction*

During the execution of a "subtract" instruction, a word stored in a predetermined address in memory is subtracted from a second word stored in a second predetermined address, and the remainder resulting from the subtraction operation is thereafter stored in a third designated address in memory.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107J, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing a "SUB" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy the next instruction word from memory into the instruction register, the address in memory of the next instruction word being indicated by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "09," go to Step 2 of the following subprogram.
2	MA-2-3	Copy into address-A the word from the address in memory specified by the contents of section 2 of the instruction register, and simultaneously copy the word digit by digit into the "J" digit-register; thereafter, go to Step 3.
3	RSB-3-4	Subtract the word stored in memory at the address specified by the contents of section 3 of the instruction register from the word just stored in address-A, and thereafter store the remainder back in address-A; thereafter, go to Step 4.
4	AM-4-5	Copy the word stored in address-A into memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 5.
5	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

From a comparison of the flow diagrams relating to an "ADD" and a "SUB" instruction shown in FIGS. 107I and 107J, it is apparent that the two subprograms relating thereto are substantially identical, the exception being that, when line SUB (FIG. 98) is rendered TRUE in order to initiate a "SUB" instruction, line SAS is likewise rendered TRUE. As previously described in detail with respect to the adder-subtracter shown in FIGS. 53 and 54, when line SAS is TRUE, a subsequently initiated "RAD" subcommand causes a subtraction mathematical operation to take place within the adder-subtractor unit rather than an addition mathematical operation, as before. Consequently, in view of the preceding detailed description of an "ADD" instruction, a detailed description of a "SUB" instruction would only result in undue repetition.

80. *Detailed Description of SUM Instructions*

In executing a single "SUM" instruction, a plurality of word stored in sequentially-numbered addresses in memory are algebraically added together, and their sum is thereafter stored in a predetermined address in memory as specified by the contents of section 2 of the instruction register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107K, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing a "SUM" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "10," go to Step 2 of the following subprogram.
2	CA(1)-0-3	Clear address-A; thereafter, go to Step 3.
3	RAD-3-4	Add to the word in address-A the word stored in memory at the address specified by the contents of section 3 of the instruction register; thereafter, go to Step 4.
4	I34-5-6	If the memory addresses specified by the contents of sections 3 and 4 of the instruction register are equal, go to Step 6; otherwise, go to Step 5.
5	IN4-0-3	Increment section 3 of the instruction register; thereafter, go to Step 3.
6	AM-2-7	Copy the word stored in address-A into the address in memory specified by the contents of section 2 of the instruction register; thereafter, go to Step 7.
7	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "SUM" instruction (i.e., "10"), the state of line SUM is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "SUM" instruction is thereafter executed in the following manner:

With reference to the composite logical diagram of a "SUM" instruction shown in FIG. 99, when line MIN is rendered FALSE upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6080** is reversed, so that line CA is thereby rendered TRUE. Consequently, a "CA" word-cycle is thereafter executed, whereby a word of all zeros is stored in address–A. When line CA is rendered FALSE by a subsequent change of state of flipflop **6080** upon completion of the just-initiated "CA" word-cycle, the state of flipflop **6097** is reversed, so that line RAD is thereby rendered TRUE, which, in turn, renders line TS3 TRUE. In response to lines RAD and TS3 being rendered TRUE, a "RAD" subcommand is thereafter executed, whereby the word stored in memory at the address specified by section 3 of the instruction register is algebraically added to the word in address–A.

If it is assumed that the contents of sections 3 and 4 of the instruction register are of unequal magnitude, indicating that there are additional words in memory to be added to the word in address–A, line (I34)′ is rendered FALSE in the manner previously described with respect to FIG. 59. Thus, when line AN is rendered FALSE upon completion of the previously-initiated "RAD" subcommand, the state of line IN4 is likewise rendered FALSE, while the state of flipflop **6097** remains unchanged. Consequently, section **3** of the instruction register is incremented by a count of "1," and thereafter a second "RAD" subcommand is executed. As a result, the word stored in memory at the second address now specified by the incremented contents of section 3 of the instruction register is also algebraically added to the word stored in address–A. If it is now assumed that the contents of section 3 of the instruction register, after being incremented, is equal to the contents of section 4 thereof, line (I34)′ is rendered TRUE, thus indicating that there are no other words in memory to be added to the word in address–A.

Therefore, when line AN is rendered FALSE upon completion of the execution of the second "RAD" subcommand, the state of flipflop **6097** is reversed, and line RAD is thereby rendered FALSE. When line RAD is thus rendered FALSE, the state of flipflop **6077** is reversed, so that line AM is thereby rendered TRUE, which, in turn, renders line TS2 likewise TRUE. In response to lines AM and TS2 being rendered TRUE, an "AM" word-cycle is thereafter executed, whereby the word in address–A is copied into memory at the address specified by section 2 of the instruction register.

Upon completion of the "AM" word-cycle operation, the state of line AM is rendered FALSE by a reversal of state of flipflop **6077**. When line AM is thus rendered FALSE, the state of flipflop **6107** is reversed, and line STD is thereby rendered TRUE; line STD, thus being rendered TRUE, likewise renders line TS5 TRUE. In response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is copied into the word-selecting register, thus terminating the sequence of operations executed in response to a "SUM" instruction.

81. *Detailed Description of APN Instructions*

In executing a single "APN" instruction, a plurality of pairs of words are algebraically added together, and their sums are thereafter stored in consecutively-numbered addresses in memory. For example, if the "APN" instruction is directed toward the four words stored in memory addresses 11 through 14, so that "11" is stored in section 3 and "14" is stored in section 4 of the instruction register, the word stored in address–11 is algebraically added to the word stored in address–14, and their sum is thereafter stored in address–14. Thereafter, sections 3 and 4 of the instruction register are respectively incremented and decremented, after which the word stored in address–12 is algebraically added to the word stored in address–13, and their sum is thereafter stored in address–13.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107L, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing an "APN" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "11," go to Step 2 of the following subprogram.
2	RLR-0-3	Preset the count of the R-counter to correspond to the low-order digit of section 2 of the instruction register; thereafter, go to Step 3.
3	MA-4-4	Copy into address–A the word from the address in memory as specified by the contents of section 4 of the instruction register, and, simultaneously therewith, copy the word, digit by digit, into the "J" digit-register; thereafter, go to Step 4.
4	RAD-3-5	Add to the word stored in address–A the word stored in memory at the address specified by section 3 of the instruction register; thereafter, go to Step 5.
5	AM-4-6	Copy the word stored in address–A into memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 6.
6	DR-0-7	Decrement the R-counter; thereafter, go to Step 7.
7	R-8-10	If the R-counter is at a count of "0," go to Step 10; otherwise, go to Step 8.
8	IN4-0-9	Increment section 3 of the instruction register; thereafter, go to Step 9.
9	DE2-0-3	Decrement section 4 of the instruction register; thereafter, go to Step 3.
10	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for an "APN" instruction (i.e., "11"), the state of line APN is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, an "APN" instruction is thereafter executed in the following manner:

With reference to FIG. 100, wherein there is illustrated a logical diagram of the particular portions of the computer circuitry utilized in executing an "APN" instruction, when line MIN is rendered FALSE upon completion of the previously-initiated "MI" word-cycle, the states of flipflops **6091** and **6088** are simultaneously reversed, whereby lines MA and JHR are respectively rendered TRUE thereby. When line JHR is thus rendered TRUE, line RLR is likewise rendered TRUE. However, at TIME-**1** after line JHR is rendered TRUE, the state of flipflop **6088** is again reversed, so that lines JHR and RLR are both rendered FALSE thereby. As a result of line RLR thus experiencing a TRUE-to-FALSE reversal of state, the R-counter is preset to a count corresponding to the value of the low-order digit of section 2 of the instruction register in the same manner as previously described with respect to FIG. 68. As is evident, the count of the R-counter is thereafter indicative of the number of word-pairs to be added. As line TS4 is likewise rendered TRUE when line MA is initially rendered TRUE, a "MA" word-cycle is thereafter executed, whereby the word stored in memory at the address specified by section 4 of the instruction register is read out and thereafter stored in address-A.

Upon completion of the "MA" word-cycle, the state of flipflop **6091** is again reversed, and line MA is thereby rendered FALSE. When line MA is thus rendered FALSE, the state of flipflop **6097** is reversed, and line RAD is thereby rendered TRUE, which, in turn, renders line TS3 likewise TRUE. In response to lines RAD and TS3 thus being rendered TRUE, a "RAD" subcommand is thereafter executed, whereby the word stored in memory at the address specified by section 3 of the instruction register is added to the word stored in address-A. Upon completion of the "RAD" subcommand, the state of flipflop **6097** is again reversed, and line RAD is thereby rendered FALSE. However, just prior to completion of the previously-initiated "RAD" subcommand, line DR experiences a TRUE-to-FALSE reversal of state and thereby causes the R-counter to be decremented by a count of "1."

When line RAD is rendered FALSE, the state of flipflop **6077** is reversed, so that line AM is thereby rendered TRUE. Line AM, being rendered TRUE, likewise renders line TS4 TRUE. Thus, in response to lines AM and TS4 being rendered TRUE, an "AM" word-cycle is thereafter executed whereby the word in address-A is stored in memory at the address specified by section 4 of the instruction register. Upon completion of the "AM" word-cycle, the state of flipflop **6077** is again reversed, so that line AM is thereby rendered FALSE. When line AM is thus rendered FALSE, lines IN4 and DE2 are likewise rendered FALSE. Consequently, sections 3 and 4 of the instruction register are respectively incremented and decremented by a count of "1."

If, after being decremented, the R-counter is at a count other than "0," indicating that there are additional word-pairs to be added, the state of flipflop **6091** is again reversed when line AM is subsequently rendered FALSE, and, as a result, line MA is again rendered TRUE for the second time. Thereafter, the above-described sequence of events is sequentially repeated until the R-counter is decremented to a count of "0," indicating that there are no other word-pairs to be added. When the R-counter reaches a count of "0," so that line Rϕ is rendered TRUE, the state of flipflop **6107** is reversed when line AM is subsequently rendered FALSE and line STD is thereby rendered TRUE. Line STD, being rendered TRUE, likewise renders line TS5 TRUE. Thus, in response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is copied into the word-selecting register, thus terminating the sequence of operations executed in response to initiation of an "APN" instruction.

82. *Detailed Description of MUS Instructions*

The subprogram for a "multiply-and-shift" instruction essentially comprises five distinct steps or phases. During the first phase, the algebraic signs of the multiplier and of the multiplicand are determined, and the uncomplemented absolute values of the two factors are thereafter derived and stored back in memory at the original addresses thereof. The second phase of the "MUS" subprogram is concerned with the actual multiplication of the two factors. During the multiplication phase, the ten-decimal-digit multiplicand is multiplied by the ten-decimal-digit multiplier, and a twenty-decimal-digit product is derived therefrom. As previously described in detail, a multiplication mathematical operation is carried out by the present computer by the process of repeated additions. That is, the multiplicand is essentially added to a twenty-digit product accumulator by the number of times specified by the value of the first-order multiplier digit, the product accumulator initially having been preset to zero. Thereafter, the thus-derived product is effectively shifted within the accumulator by one digital position to the right, and the multiplicand is again added to the thus-derived product by the number of times specified by the value of the second-order multiplier digit. The product is again shifted one digital position to the right, and successive additions with respect to the third-order multiplier digit are thereafter effected, and, so on, with respect to the remaining higher-order multiplier digits. Thus, upon completion of the multiplication operation, a twenty-decimal-digit product appears in the product accumulator, which essentially comprises addresses A and B operatively connected end to end.

During the third phase of the subprogram, the twenty-digit product is shifted in the direction and by the number of digital positions indicated by the contents of section 2 of the instruction register. The fourth phase of the subprogram is concerned with the copying of the ten low-order digits of the twenty-digit product back in memory at the address specified by section 4 of the instruction register. Due to the fact that the absolute value of the product is stored in memory at this particular time, the fifth and final phase of the subprogram is concerned with determining the algebraic sign of the product and thereafter complementing the product in the event its algebraic sign is negative; however, in the event the sign of the product is positive, the product is not complemented.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107N, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing a "MUS" instruction.

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "13," go to Step 2 in the following subprogram.
2	PFϕ-0-3	Preset the F-counter to "0"; thereafter, go to Step 3.
3	PRϕ-0-4	Preset the R-counter to "0"; thereafter, go to Step 4.
4	MA-3-5	Copy the word from memory into address-A and, simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the word in memory being specified by section 3 of the instruction register; thereafter, go to Step 5.
5	OBM(4)-0-6	Preset the state of line OBM to correspond to algebraic sign of the word just read from memory thereafter, go to Step 6.
6	OBM(2)-7-8	If the state of line OBM is TRUE, go to Step 8 otherwise, go to Step 7.
7	CPA-0-8	Complement the word in address-A thereafter, go to Step 8.
8	CB-0-9	Clear address-B thereafter, go to Step 9.
9	AM-n-10	Copy the word stored in address-A into memory at the address specified by section n of the instruction register, "n" being a "3" if the R-counter is at a count of "0," and being a "4" if the R-counter is at a count of "8" thereafter, go to Step 10.
10	R8-11-15	If the R-counter is at a count of "8," go to Step 15 otherwise, go to Step 11.
11	PR8-0-12	Preset the R-counter to "8" thereafter, go to Step 12.
12	MA-4-13	Copy the word from memory into address-A and, simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the word in memory being specified by section 4 of the instruction register; thereafter, go to Step 13.
13	OBN(4)-0-14	Preset the state of line OBN TRUE if the algebraic sign of the word just read from memory corresponds to the state of line OBM; otherwise, preset the state of line OBN FALSE; thereafter, go to Step 14.
14	J9-8-7	If the digit stored in the "J" digit-register is a "9," go to Step 7; otherwise, go to Step 8.
15	CA(1)-0-16	Clear address-A; thereafter, to to Step 16.
16	PRϕ-0-17	Preset the R-counter to "0"; thereafter, go to Step 17.
17	AD(1)-0-18	Increment the digit-counter; thereafter, go to Step 18.
18	IFD-20-19	If the digit stored in the F-counter is equal to the digit stored in the digit-counter, go to Step 19; otherwise, go to Step 20.
19	RLM-3-20	Preset the R-counter to a count corresponding to the value of the digit whose digital order is specified by the F-counter and whose address in memory is specified by section 3 of the instruction register; thereafter, go to Step 20.
20	D9-17-21	If the digit-counter is at a count of "9," go to Step 21; otherwise, go to Step 17.
21	Rϕ-22-24	If the R-counter is at a count of "0", go to Step 24; otherwise, go to Step 22.
22	RAD-4-23	Add to the word stored in address-A the word stored in memory at the address specified by section 4 of the instruction register; thereafter, go to Step 23.
23	DR-0-21	Decrement the R-counter; thereafter, go to Step 21.
24	SA(2)-0-25	Shift the word of address-A one place to the right; thereafter, go to Step 25.
25	SB(2)-0-26	Shift the word of address-B one place to the right; thereafter, go to Step 26.
26	AF-0-27	Increment the F-counter; thereafter, go to Step 27.
27	F1ϕ-17-28	If the F-counter is at a count of "10," go to Step 28; otherwise, go to Step 17.
28	RLR-0-29	Preset the R-counter to a count corresponding to the low-order digit of section 2 of the instruction register; thereafter, go to Step 29.
29	Rϕ-30-41	If the R-counter is at a count of "0" go to Step 41; otherwise, go to Step 30.
30	H2-31-33	If the high-order digit of section 2 of the instruction register is a "2," go to Step 33; otherwise, go to Step 31.
31	H3-32-35	If the high-order digit of section 2 of the instruction register is a "3," go to Step 35; otherwise, go to Step 32.
32	H4-0-36	If the high-order digit of section 2 of the instruction register is a "4," go to Step 36.
33	SA(1)-0-34	Shift the word of address-A one place to the left; thereafter, go to Step 34.
34	SB(1)-0-40	Shift the word of address-B one place to the left; thereafter, go to Step 40.
35	R1-36-38	If the R-counter is at a count of "1," go to Step 38; otherwise, go to Step 36.
36	SA(2)-0-37	Shift the word of address-A one place to the right; thereafter, go to Step 37.
37	SB(2)-0-40	Shift the word of address-B one place to the right; thereafter, go to Step 40.
38	DAD-0-39	Add the constant 0000000005 to the word stored in address-B; thereafter, to go to Step 39.
39	ARO-0-36	Add a word of all zeros to address-A; thereafter, go to Step 36.
40	DR-0-29	Decrement the R-counter; thereafter, go to Step 29.
41	OBM(2)-42-43	If the state of line OBM is TRUE, go to Step 43; otherwise, go to Step 42.
42	CPM-3-43	Complement the word stored in memory at the address specified by section 3 of the instruction register; thereafter, go to Step 43.
43	BM-4-44	Copy the word of address-B into memory at the address specified by section 4 of the instruction register; thereafter, go to Step 44.
44	PR8-0-45	Preset the R-counter to a count of "8"; thereafter, go to Step 45.
45	OBN(1)-46-47	If the state of line OBN is TRUE, go to Step 47; otherwise, go to Step 46.
46	CPM-4-47	Complement the word stored in memory at the address specified by section 4 of the instruction register; thereafter, go to Step 47.
47	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "MUS" instruction (i.e., "13"), the state of line MUS is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "MUS" instruction is thereafter executed in the following manner:

With reference to FIGS. 102A through 102C, there is illustrated therein a composite logical diagram of various portions of the computer circuitry utilized in executing a "MUS" instruction. With particular reference to FIG. 102B, at TIME-1 after line MI is rendered TRUE at the beginning of the previously-initiated "MI" word-cycle, the states of flipflops **6095** and **6096** are reversed, so that lines OBM and OBN, if not already FALSE, are respectively rendered FALSE thereby. Upon completion of the previously-initiated "MI" word-cycle, line PFϕ experiences a TRUE-to-FALSE reversal of state when line MIN is rendered TRUE. Consequently, when line PFϕ is thus rendered FALSE, the F-counter is thereby caused to be preset to "0." Also as a result of line MIN thus being rendered FALSE, the state of flipflop **6091** is reversed, so that line MA is thereby rendered TRUE. With reference to FIG. 79, the state of flipflop **6107** is reversed at the beginning of the previously-initiated "MI" word-cycle when line (MI)′ is thus rendered FALSE, so that line STD is rendered FALSE thereby. With reference back to FIG. 102C, when line STD is thus rendered FALSE, line PRϕ is likewise rendered FALSE, so that the R-counter is thereby caused to be preset to "0."

With reference to FIG. 102A, as the R-counter is now at a count of "0," so that line Rϕ is TRUE, line TS3 is likewise rendered TRUE when line MA is initially rendered TRUE. As a result of lines MA and TS3 being rendered TRUE, a first "MA" word-cycle is thereafter executed, whereby the multiplier word stored in the address in memory specified by the contents of section 3 of the instruction register is read out and stored in address-A. Simultaneously therewith, the multiplier word just read out from memory is copied digit by digit into the "J" digit-register. Upon completion of the "MA" word-cycle, the state of flipflop **6091** (FIG. 102B) is again reversed, so that line MA is rendered FALSE thereby. If, upon completion of the just-executed "MA" word-cycle, the algebraic sign of the multiplier word just read from memory is negative, so that the tenth-order digit thereof stored in the "J" digit-register is a "9," the state of line OBM remains FALSE. However, if the tenth-order digit stored in the "J" digit-register has a value other than "9," the state of flipflop **6095** is reversed upon completion of the "MA" word-cycle, and line OBM is thereby rendered TRUE. Thus, it is evident, the state of line OBM is rendered TRUE if the algebraic sign of the multiplier word is positive and is rendered FALSE if the algebraic sign of the multiplier word is negative.

If the algebraic sign of the multiplier word just read from memory is negative, so that line J9 is rendered TRUE, the state of flipflop **6083** is reversed upon completion of the previously-initiated "MA" word-cycle, and line CPA is thereby rendered TRUE. As previously described, when line CPA is rendered TRUE, a "CPA" subcommand is thereafter executed, whereby the multiplier word stored in address-A is first read out and complemented, and, thereafter, the absolute negative value thereof is stored back in address-A. Upon completion of the "CPA" subcommand, the state of flipflop **6083** is again reversed, and line CPA is thereby rendered FALSE. It is to be noted that the state of flipflop **6081** is reversed and line CB thereby rendered TRUE, either upon completion of the first "MA" word-cycle if the algebraic sign of the multiplier word is positive, or upon completion of the subsequently-initiated "CPA" subcommand if the algebraic sign of the multiplier word is negative. In either event, in response to line CB thus being rendered TRUE, a "CB" word-cycle is thereafter executed, whereby a word of all zeros is stored in address-B. Upon completion of the "CB" word-cycle, the state of flipflop **6081** is again reversed, so that line CB is thereby rendered FALSE.

With reference to FIG. 102A, in response to line CB thus being rendered FALSE, the state of flipflop **6077** is reversed, and line AM is rendered TRUE thereby. Line AM being rendered TRUE likewise renders line TS3 TRUE. Consequently, in response to lines AM and TS3 being rendered TRUE, an "AM" word-cycle is thereafter executed, whereby the absolute value of the multiplier word stored in address-A is read out and stored in memory at the address specified by the contents of section 3 of the instruction register. Upon completion of the "AM" word-cycle, the state of flipflop **6077** is again reversed, so that line AM is rendered FALSE thereby. As shown in FIG. 102B, line PR8 is also rendered FALSE as a result of line AM thus being rendered FALSE. Line PR8 thus being rendered FALSE causes the R-counter to be preset to a count of "8." Also, when line AM is thus rendered FALSE, the state of flipflop **6091** is again reversed, so that line MA is rendered TRUE for the second time. However, as the R-counter is now at a count of "8," so that line **R8** is TRUE, line **TS4** (FIG. 102C) is likewise rendered TRUE when line MA is rendered TRUE. In response to lines MA and **TS4** thus being rendered TRUE, a second "MA" word-cycle is thereafter executed, whereby the multiplicand word stored in memory at the address specified by the contents of section 4 of the instruction register is read out and stored in address-A. Simultaneously therewith, the multiplicand word just read out from memory is copied digit by digit into the "J" digit-register.

Upon completion of the second "MA" word-cycle, the state of flipflop **6091** (FIG. 102B) is again reversed, so that line MA is rendered FALSE thereby. If, upon completion of the execution of the second "MA" word-cycle, the algebraic sign of the multiplicand word just read out from memory (as indicated by a TRUE state of line J9 if negative and a TRUE state of line (J9)′ if positive) corresponds to the algebraic sign of the multiplier word (as indicated by a TRUE state of line OBM if positive and a TRUE state of line (OBM)′), the state of flipflop **6096** is reversed, so that line OBN is rendered TRUE, indicating that the algebraic sign of the product is positive. However, if such correspondence does not exist, the state of line OBN remains FALSE, indicating that the algebraic sign of the product of the two factors is negative.

In addition, if the algebraic sign of the multiplicand is negative, so that line J9 is rendered TRUE, the state of flipflop **6083** is reversed upon completion of the previously initiated "MA" word-cycle, and line CPA is thereby rendered TRUE. As previously described, when line CPA is rendered TRUE, a "CPA" subcommand is thereafter executed, whereby the word stored in address-A is read out and complemented, and, thereafter, the absolute negative value thereof is stored back in address-A. Upon completion of the "CPA" subcommand, the state of flipflop **6083** is again reversed, and line CPA is thereby rendered FALSE. It is again to be noted that the state of flipflop **6081** is reversed and line CB thereby rendered TRUE, either upon completion of the second "MA" word-cycle if the algebraic sign of the multiplicand is positive, or upon completion of the "CPA" subcommand if the alegbraic sign of the multiplicand is negative. In either event, in response to line CB thus being rendered TRUE, a "CB" word-cycle is thereafter executed, whereby a word of all zeros is stored in address-B. Upon completion of the just-initiated word-cycle "CB", the state of flipflop **6081** is again reversed, so that line CB is thereby rendered FALSE. With reference to FIG. 102A, when line CB is thus rendered FALSE, the state of flipflop **6077** is reversed, and line AM is rendered TRUE thereby for the second time. When line AM is rendered TRUE, line TS4 (FIG. 102C) is likewise rendered TRUE. Consequently, in response to lines AM and TS4 being rendered TRUE, an "AM" word-cycle is thereafter executed, whereby the word in address-A is stored in memory at the address specified by the contents of section 4 of the instruction register. Upon completion of the "AM" word-cycle, the state of flipflop **6077** is again reversed, so that line AM is rendered FALSE thereby.

The steps of the subprogram to this point have been concerned with determining the algebraic signs of the multiplicand and the multiplier, conditioning the state of line OBN to be indicative of the algebraic sign of the product of the two factors, complementing either or both of the factors if the algebraic sign of either or both is negative, and thereafter returning the absolute value of the two factors to their original addresses in memory.

With reference now to FIG. 102A, after the absolute value of the multiplicand is stored in memory at the address specified by the contents of section 4 of the instruction register, via the previously-initiated "AM" word-cycle, the state of flipflop **6080** is reversed when line AM is thus rendered FALSE for the second time, due to the fact that the R-counter is now at a count of "8," as indicated by a TRUE state of line **R8**. When the state of flipflop **6080** is reversed and line CA is rendered TRUE thereby. In response to line CA being rendered TRUE, a "CA" word-cycle is thereafter executed, whereby a word of all zeros is again stored in address-A. Thereafter, the state of flipflop **6080** is again reversed, and line CA is rendered FALSE. As a result of line CA thus being rendered FALSE, the state of flipflop **6066** is reversed, so that line MJ is rendered TRUE thereby. Substantially coincidentally therewith, line PRϕ (FIG. 102C) experiences a TRUE-to-FALSE reversal of state and thereby causes the R-counter to be preset to "0."

When line MJ is thus rendered TRUE, line TS3 (FIG. 102A) is likewise rendered TRUE. Consequently, in response to lines MJ and TS3 being rendered TRUE, a "MJ" word-cycle is thereafter executed, whereby the multiplier stored in memory at the address specified by section 3 of the instruction register is read out and the first-order digit thereof is stored in the R-counter.

The reason that only the first-order multiplier digit is stored in the R-counter via the previously initiated "MJ" word-cycle is that the F-counter was initially preset to zero, and thus line RLM is TRUE only during that time the count of the digit-counter is also equal to zero. Otherwise, line (IFD)′, hence line RLM, remains FALSE. As previously described in detail, a digit is stored in the "J" digit-register from memory only when line RLM is TRUE. After the first-order digit of the multiplier is stored in the R-counter, the state of flipflop **6066** (FIG. 102A) is again reversed, so that line MJ is rendered FALSE thereby. If the first-order multiplier digit now stored in the R-counter has a value other than "0" as indicated by line (Rϕ)′ being TRUE, the state of flipflop **6097** is reversed when line MJ is thus rendered FALSE, and, as a result, line RAD is rendered TRUE thereby. When line RAD is thus rendered TRUE, line TS4 (FIG. 102C) is likewise rendered TRUE.

Consequently, in response to lines RAD and TS4 thus being rendered TRUE, a first "RAD" subcommand is thereafter executed, whereby the multiplicand stored in memory at the address specified by the contents of section 4 of the instruction register is read out and added to the contents of address-A. Upon completion of the first "RAD" subcommand, line DR (FIG. 102C) experiences a TRUE-to-FALSE reversal of state, so that the R-counter is decremented by a count of "1." If, after being decremented, the R-counter is still at a count other than "0," lines RAD and TS4 remain TRUE, so that a second "RAD" subcommand is thereafter executed, whereby the multiplicand is again read out from the address in memory specified by the contents of section 4 of the instruction register and is added to the partial product stored in address-A. After the second "RAD" subcommand is completed, the R-counter is again decremented by a count of "1," and, if, after being decremented for the second time, the R-counter is still at a count other than "0," a third "RAD" subcommand is thereafter executed, and so on.

However, when the R-counter has been decremented to a count of "0," so that line Rϕ is rendered TRUE thereby, indicating that the partial product with respect to the first-order digit of the multiplier word has been derived and stored in address-A, the state of flipflop **6097** (FIG. 102A) is again reversed, so that line RAD is rendered FALSE thereby. As a result of line RAD thus being rendered FALSE, the state of flipflop **6101** is reversed, so that line SA is thereby rendered TRUE. In response to line SA thus being rendered TRUE, a "SA" subcommand is thereafter executed, whereby the partial product now stored in address-A is shifted to the right by one digital position in the same manner as previously described in detail. Upon completion of the previously-initiated "SA" subcommand, the state of flipflop **6101** is again reversed, so that line SA is rendered FALSE thereby. When line SA is thus rendered FALSE, the state of flipflop **6103** is reversed, so that line SB is thereby rendered TRUE. In response to line "SB" being rendered TRUE, a "SB" subcommand is thereafter executed, whereby the word stored in address-B is shifted to the right by one digital position, with the original first-order digit of the word in address-A being stored in the tenth-order digital position of address-B. Upon completion of the just-initiated "SB" subcommand, the state of flipflop **6103** is again reversed, so that line SB is rendered FALSE thereby.

With reference to FIG. 102C, just prior to the completion of the previously-initiated "SB" subcommand, line AF experiences a TRUE-to-FALSE reversal of state when line EG is rendered FALSE; consequently, the F-counter is incremented from a count of "0" to a count of "1." Thus with reference to FIG. 102A, as the F-counter is at a count other than "10," so that line **(F1ϕ)′** remains TRUE, the state of flip-flop **6066** is again reversed upon completion of the just-initiated "SB" subcommand, so that lines MJ and TS3 are both rendered TRUE for the second time. With reference to FIG. 102C, it is again to be noted that, due to the fact that the F-counter is now at a count of "1," line (IFD)′, hence line RLM, is rendered TRUE only during the time the digit-counter is also at a count of "1"; otherwise, line RLM remains FALSE. As a result of lines MJ and TS3 again being rendered TRUE, a second "MJ" digit-cycle is thereafter executed, whereby the second-order digit of the multiplier word is read out from the address specified by section 3 of the instruction register and is thereafter stored in the R-counter. Thereafter, a succession of "RAD" subcommands is executed, whereby the multiplicand is added to the partial product now stored in address-A by the number of times equal to the digital value of the second-order multiplier digit stored in the R-counter. Thereafter, the twenty-digit word stored in combined addresses A and B is shifted to the right by one digital position in the same manner as just described.

The above-described sequence of events is sequentially repeated until the tenth-order digit of the multiplier is stored in the R-counter, the multiplicand is again added to the contents of address-A by the number of times dictated by the digital value of the tenth-order multiplier digit, and the twenty-digit product stored in combined addresses A and B is shifted to the right for the tenth time by one digital position. As the F-counter is now at a count of "10," as indicated by line **F10** being rendered TRUE, the multiplication portion of the "MUS" instruction is now completed.

During the next phase of the "MUS" instruction, a succession of "SA" and "SB" subcommands is executed, whereby the twenty-digit product is shifted either to the right or to the left by the number of digital positions indicated by the contents of section 2 of the instruction register, all in essentially the same manner as previously described in detail with respect to that portion of the preceding description entitled "Description of SHF Instruction" and also with respect to that portion of the preceding description relating to the "SA" and "SB" subcommands.

With reference to FIG. 102B, if the state of line (OBM)′ was previously rendered TRUE, indicating that the algebraic sign of the multiplier is negative, the state of flipflop **6084** is reversed upon completion of the last "SB" subcommand, during which time lines **F1ϕ** and **Rϕ** are both rendered TRUE. Consequently, the state of line CPM is rendered TRUE in order to effect complementing of the multiplier stored in memory at the address specified by section 3 of the instruction register. However, if the algebraic sign of the multiplier is positive, the state of line (OBM)′ thus being FALSE, a subsequently-initiated "CPM" subcommand is prevented. As shown in FIG. 102C, the state of flipflop **6079** is reversed and line BM thereby rendered TRUE upon completion of the last-initiated "SB" subcommand, if the state of line OBM is TRUE, indicating that the algebraic sign of the multiplier is positive; otherwise, the state of flipflop **6079** remains unchanged and line BM thereby remains FALSE until the last-initiated "CPM" subcommand is completed. However, when line BM is finally rendered TRUE due to a reversal of state of flipflop **6079**, line TS4 is likewise rendered TRUE.

In response to lines BM and TS4 thus being rendered TRUE, a "BM" word-cycle is thereafter executed, whereby the word is address-B is read out and stored in memory at the address specified by the contents of section 4 of the instruction register. Upon completion of the "BM" word-cycle, the state of flipflop **6079** is again reversed, so that line BM is rendered FALSE thereby. With reference to FIG. 102B, when line BM is thus rendered FALSE, the state of line PR8 is reversed from TRUE to FALSE, thus causing the R-counter to be preset to a count of "8," so that line R8 is rendered TRUE thereby. If the state of line (OBN)′ was previously rendered TRUE, indicating that the algebraic sign of the product is negative, the state of flipflop **6084** is reversed when line BM is thus rendered FALSE, and, as a result, line CPM is rendered TRUE thereby. When line CPM is thus rendered TRUE, line TS4 (FIG. 102C) is likewise rendered TRUE. Thus, in response to lines CPM and TS4 being rendered TRUE, a "CPM" subcommand is thereafter executed, whereby the product stored in memory at the address specified by the contents of section 4 of the instruction register is complemented thereby. Upon completion of the just-initiated "CPM" subcommand, the state of flipflop **6084** (FIG. 102B) is again reversed, so that line CPM is rendered FALSE thereby.

As shown, the state of flip-flop **6107** is reversed and line STD rendered TRUE upon completion of the just-initiated "BM" subcommand if the state of line OBN is TRUE, indicating that the algebraic sign of the product is positive; otherwise, the state of flipflop **6107** is not so reversed, and line STD is thereby rendered TRUE until the previously-initiated "CPM" subcommand is completed. However, when line STD is finally rendered TRUE, line TS5 is likewise rendered TRUE. In response to lines STD and TS5 thus being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events initiated in accordance with the "MUS" instruction.

83. *Detailed Description of MDD Instructions*

A "MDD" instruction is essentially a special case of the just-described "MUS" instruction. For example, in

executing a "MDD" instruction, the code number "32" is effectively utilized during the "shift product" phase of the subprogram instead of the number stored in section 2 of the instruction register, as in the previously-described "MUS" instruction. In addition, in executing a "MDD" instruction, the address in memory of the multiplier is specified by the contents of section 2 of the instruction register instead of section 4, as in the execution of the previously-described "MUS" instruction.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107M, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "MDD" instruction:

Step	Subinstructions	Description
1	MI-0-1	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "12," go to Step 2 of the following subprogram.
2	PFϕ-0-3	Preset the F-counter to "0"; thereafter, go to Step 3.
3	PRϕ-0-4	Preset the R-counter to "0"; thereafter, go to Step 4.
4	MA-3-5	Copy the next word from memory into address-A, and, simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the next word in memory being specified by the contents of section 3 of the instruction register; thereafter, go to Step 5.
5	OBM(4)-0-6	Preset the state of line OBM to correspond to the algebraic sign of the word just read from memory; thereafter, go to Step 6.
6	OBM(2)-7-8	If the state of line OBM is TRUE, go to Step 8; otherwise, go to Step 7.
7	CPA-0-8	Complement the word stored in address-A; thereafter, go to Step 8.
8	CB-0-9	Clear address-B; thereafter, go to Step 9.
9	AM-*n*-10	Copy the word stored in address-A into memory at the address specified by section *n* of the instruction register, "*n*" being a "3" if the R-counter is at a count of "0", and being a "4" if the R-counter is at a count of "8"; thereafter, go to Step 10.
10	R8-11-15	If the R-counter is at a count of "8", go to Step 15; otherwise, go to Step 11.
11	PR8-0-12	Preset the R-counter to "8"; thereafter go to Step 12.
12	MA-2-13	Copy the next word from memory into address-A, and, simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the next word in memory being specified by the contents of section 2 of the instruction register; thereafter, go to Step 13.
13	OBN(4)-0-14	Preset the state of line OBN TRUE if the algebraic sign of the word just read from memory corresponds to the state of line OBM; otherwise, preset the state of line OBN FALSE; thereafter, go to Step 14.
14	J9-8-7	If the digit stored in the "J" digit register is a "9", go to Step 7; otherwise, go to Step 8.
15	CA(1)-0-16	Clear address-A; thereafter, go to Step 16.
16	PRϕ-0-17	Preset the R-counter to "0"; thereafter, go to Step 17.
17	AD(1)-0-18	Increment the digit-counter; thereafter, go to Step 18.
18	IFD-20-19	If the digit stored in the F-counter is equal to the digit stored in the digit counter, go to Step 19; otherwise, go to Step 20.
19	RLM-3-20	Store in the R-counter the digit whose digital position is specified by the F-counter and whose address in memory is specified by the contents of section 3 of the instruction register; thereafter, go to Step 20.
20	D9-17-21	If the digit-counter is at a count of "9," go to Step 21; otherwise, go to Step 17.
21	Rϕ-22-24	If the R-counter is at a count of "0," go to Step 24; otherwise, go to Step 22.
22	RAD-4-23	Add the word stored in memory at the address specified by section 4 of the instruction register to the word stored in address-A; thereafter, go to Step 23.
23	DR-0-21	Decrement the R-counter; thereafter, go to Step 21.
24	SA(2)-0-25	Shift the word of address-A one place to the right; thereafter, go to Step 25.
25	SB(2)-0-26	Shift the word address-B one place to the right; thereafter, go to Step 26.
26	AF-0-27	Increment the F-counter; thereafter, go to Step 27.
27	F1ϕ-17-28	If the F-counter is at a count of "10", go to Step 28; otherwise, go to Step 17.
28	PR2-0-29	Preset the R-counter to a count of "2"; thereafter, go to Step 29.
29	Rϕ-30-36	If the R-counter is at a count of "0", go to Step 36; otherwise, go to Step 30.

Step	Subinstructions	Description
30	R1-31-33	If the R-counter is at a count of "1", go to Step 33; otherwise, go to Step 31.
31	SA(2)-0-32	Shift the word of address-A one place to the right; thereafter, go to Step 32.
32	SB(2)-0-35	Shift the word of address-B one place to the right; thereafter, go to Step 35.
33	DAD-0-34	Add the constant 0000000005 to the word stored in address-B; thereafter, go to Step 34.
34	ARO-0-31	Add a word of all zeros to address-A; thereafter, go to Step 31.
35	DR-0-29	Decrement the R-counter; thereafter, go to Step 29.
36	OBM(2)-37-38.	If the state of line OBM is TRUE, go to Step 38; otherwise, go to Step 37.
37	CPM-3-38	Complement the word stored in memory at the address specified by the contents of section 3 of the instruction register; thereafter, go to Step 38.
38	BM-4-39	Copy the word of address-B into memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 39.
39	PR8-0-40	Preset the R-counter to a count of "8"; thereafter, go to Step 40.
40	OBN(2)-41-42.	If the state of line OBN is TRUE, go to Step 42; otherwise, go to Step 41.
41	CPM-4-42	Complement the word stored in memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 42.
42	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

As previously described with respect to FIG. 56, if the two-decimal-digit number stored in section 1 of the instruction register is either a "12" or a "13," line MUS is thereby rendered TRUE. Line MDD is rendered TRUE, however, only when the number "12" is stored in section 1 of the instruction register. Thus, upon completion of the previously-initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and is thereafter stored in the instruction register, if the contents of section 1 of the instruction register corresponds to the code designation for a "MDD" instruction (i.e., "12"), the states of lines MUS and MDD are both rendered TRUE in the manner just stated. In response to both of lines MUS and MDD simultaneously being rendered TRUE, a "MDD" instruction is thereafter executed in essentially the same manner as just described with respect to the "MUS" instruction. This is evident from a comparison of the "MDD" and "MUS" flow diagrams shown in FIGS. 107M and 107N, respectively, together with the accompanying step-by-step descriptions thereof previously given in chart form. This is further exemplified by the fact that the combination of the logical diagrams shown in FIGS. 101 and 102A through 102C form a composite logical diagram of particular portions of the computer circuitry utilized in executing a "MDD" instruction.

The primary differences between the "MDD" and "MUS" instructions are as follows: During the execution of the second "MA" word-cycle, during which time the multiplicand is read out from a particular address in memory, and also during which time the R-counter is at a count of "8," the state of line TS2 (FIG. 101) is rendered TRUE, whereas line TS4 (FIG. 102C) is rendered FALSE by logical AND **1521**. Consequently, during the execution of the second "MA" word-cycle portion of the "MDD" subprogram, the multiplicand to be thus stored in address-A is read out from the address in memory specified by the contents of section 2 of the instruction register, rather than being read out from the address in memory specified by the contents of section 4 of the instruction register, as in the execution of the previously-described "MUS" instruction. Thus, upon completion of the first-initiated "SB" subcommand portion of the "MDD" subprogram, during which time the F-counter is at a count of "9," line PR2 (FIG. 101) is rendered FALSE, so that the R-counter is preset to "2," rather than having stored therein the low-order digit of section 2 of the instruction register via execution of a "RLR" subinstruction, as in the execution of the "MUS" instruction. Also, as line (MDD)′ is rendered FALSE during the execution of the "MDD" instruction, lines H2 and H4 (FIG. 101) are likewise rendered FALSE. However, when line MDD is initially rendered TRUE, line H3 is likewise rendered TRUE. Therefore, as the R-counter is at a count of "2," and as line H3 is TRUE during the execution of the secondly-initiated "SA" and "SB" subcommand portions of the "MDD" subprogram, the twenty-digit product stored in addresses A and B is shifted one place to the right, thereafter rounded and again shifted one place to the right, all in the same manner as previously described. After the final shifting operation is completed, the "MDD" subprogram is thereafter terminated in exactly the same manner as the previously-described "MUS" subprogram.

84. *Detailed Description of DIV Instructions*

The subprogram for the "DIV" instruction essentially comprises three distinct steps or phases. During the first phase, the divisor is read out from the address in memory specified by the contents of section 3 of the instruction register and is stored in address-A. The algebraic sign of the divisor is thereafter determined, and its uncomplemented absolute value is derived and stored back in the original address in memory thereof. Thereafter, the dividend is read out from the address in memory specified by the contents of section 2 of the instruction register and is stored in address-A. The algebraic sign of the dividend is determined, and its uncomplemented absolute value is thereafter derived and stored back in address-A. During the division phase of the subprogram, the ten-decimal-digit dividend in address-A is effectively divided by the ten-decimal digit divisor, and a ten-decimal-digit quotient is derived from it and stored in address-B.

As previously stated, a division mathematical operation is carried out within the computer by the process of repeated subtractions. That is, the divisor is successively subtracted from the contents of address-A until the magnitude of the contents of address-A is less than the magnitude of the divisor, the word 0000000001 being added to the contents of address-B upon completion of each subtraction operation. After the word in address-A becomes less than the divisor, the contents of addresses A and B are each effectively shifted to the left by one digital position. Thereafter, the divisor is again successively subtracted from the contents of address-A until the magnitude thereof again becomes less than the magnitude of the divisor, the word 0000000001 again being added to the contents of address-B upon completion of each subtraction operation. Thereafter, the contents of addresses A and B are again effectively shifted to the left by one digital position and the sequence of events again repeated until the contents of addresses A and B are shifted to the left by the total number of nine digital positions. The third phase of the subprogram is concerned with the copying of the contents of address-B back in memory at the address specified by the contents of section 4 of the instruction register. Due to the fact that the absolute value of the quotient is now stored in memory, the fourth and final phase of the subprogram is concerned with determining the algebraic sign of the quotient and thereafter complementing the quotient in the event its algebraic sign is negative; however, in the event the algebraic sign of the quotient is positive, a subsequent complementing operation is not performed.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107P, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "DIV" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "14," go to Step 2 in the following subprogram.
2	PRϕ-0-3	Preset the R-counter to "o"; thereafter, go to Step 3.
3	MA-3-4	Copy the word from memory into address-A, and, simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the word in memory being specified by the contents of section 3 of the instruction register; thereafter, go to Step 4.
4	OBM(4)-0-5	Preset the state of line OBM to correspond to the algebraic sign of the word just read from memory; thereafter, go to Step 5.
5	OBM(2)-6-7	If the state of line OBM is TRUE, go to Step 7; otherwise, go to Step 6.
6	CPA-0-7	Complement the word stored in address-A; thereafter, go to Step 7.
7	Rϕ-14-8	If the R-counter is at a count of "0," go to Step 8; otherwise, go to Step 14.
8	CB-0-9	Clear address-B; thereafter, go to Step 9.
9	AM-4-10	Copy the word stored in address-A into memory at the address specified by the contents of the section 4 of the instruction register; register; thereafter, go to Step 10.
10	PR8-0-11	Preset the R-counter to a count of "8"; thereafter, go to Step 11.
11	MA-2-12	Copy the word stored in memory at the address specified by the contents of section 2 of the instruction register into address-A, and, simultaneously therewith, copy the word digit by digit into the "J" digit-register; thereafter, go to Step 12.
12	OBN(4)-0-13	Preset the state of line OBN TRUE if the algebraic sign of the word just read from memory corresponds to the state of line OBM; otherwise, preset the state of line OBN FALSE; thereafter, go to Step 13.
13	J9-14-6	If the digit stored in the "J" digit-register is a "9," go to Step 6; otherwise, go to Step 14.
14	RSB-4-15	Subtract the word stored in memory at the address specified by the contents of section 4 of the instruction register from the word stored in address-A; thereafter, go to Step 15.
15	EAS-17-16	If the algebraic sign of the difference is positive, go to Step 16; otherwise, go to Step 17.
16	DAD-0-14	Add the constant 0000000001 to the word stored in address-B; thereafter, go to Step 14.
17	RAD-4-18	Add the word stored in memory at the address specified by the contents of section 4 of the instruction register to the word stored in address-A; thereafter, go to Step 18.
18	DR-0-19	Decrement the R-counter; thereafter, go to Step 19.
19	Rϕ-20-22	If the R-counter is at a count of "O," go to Step 22; otherwise, go to Step 20.
20	SA(1)-0-21	Shift the word stored in address-A one digital position to the left; thereafter, go to Step 21.
21	SB(1)-0-14	Shift the word stored in address-B one digital position to the left; thereafter, go to Step 14.
22	BM-4-23	Copy the word of address-B into memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 23.
23	OBN(1)-24-25	If the state of line OBN is TRUE, go to Step 25; otherwise, go to Step 24.
24	CPM-4-25	Complement the word stored in memory at the address specified by the contents of section 4 of the instruction register; thereafter, go to Step 25.
25	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section **1** of the instruction register thereafter corresponds to the code designation for a "DIV" instruction (i.e. "14"), the state of line DIV is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "DIV" instruction is thereafter executed in the following manner:

With reference to FIGS. 103A and 103B, there is illustrated therein a composite logical diagram of particular portions of the computer circuit utilized in executing a "DIV" instruction. With particular reference to FIG. 103B, at TIME-1 after line MI is rendered TRUE at the beginning of the previously-initiated "MI" word-cycle, the states of flipflops **6096** are reversed, so that lines OBM and OBN are respectively rendered FALSE thereby. Upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6091** is reversed when line MIN is subsequently rendered FALSE, so that line MA is rendered TRUE thereby. With reference to FIG. 79, at the beginning of the previously initiated "MI" word-cycle, the state of flipflop **6107** is reversed, when line (MI)′ is thus rendered FALSE, so that line STD is rendered FALSE thereby. With reference back to FIG. 103B, when line STD is thus rendered FALSE, line PRϕ is likewise rendered FALSE, so that the R-counter is thereby caused to be preset to "0."

With reference to FIG. 103A, as the R-counter is now at a count of "0," so that line Rϕ is TRUE, line **TS3** is rendered TRUE as a result of line MA being rendered TRUE. Consequently, a first "MA" word-cycle is thereafter executed, whereby the divisor stored in memory at the address specified by the contents of section 3 of the instruction register is read out and stored in address-A. Simultaneously therewith, the divisor is copied digit by digit into the "J" digit-register. Upon completion of the "MA" word-cycle, the state of flipflop **6091** (FIG. 103B) is again reversed, so that line MA is rendered FALSE thereby. If upon completion of the "MA" word-cycle the algebraic sign of the divisor just read from memory is negative, so that the tenth-order digit thereof stored in the "J" digit-register is a "9," the state of line OBM remains FALSE. However, if the tenth-order digit of the divisor now stored in the "J" digit-register has a value other than "9," the state of flipflop **6095** is reversed on completion of the "MA" word-cycle, and line OBM is thereby rendered TRUE. Thus, it is evident, the state of line OBM is rendered TRUE if the algebraic sign of the divisor is positive and is rendered FALSE if the algebraic sign of the divisor is negative.

If the algebraic sign of the divisor just read from memory is negative, so that line **J9** is TRUE, the state of flipflop **6083** is reversed upon completion of the previously-initiated "MA" word-cycle, and line CPA is thereby rendered TRUE. In response to line CPA thus being rendered TRUE, a "CPA" subcommand is thereafter executed, whereby the divisor now stored in address-A is read out and complemented, and its absolute negative value thereafter stored back in address-A. Upon completion of the "CPA" subcommand, the state of flipflop **6083** is again reversed, and line CPA is thereby rendered FALSE. It is evident, therefore, that the state of flipflop **6081** is reversed and line CB thereby rendered TRUE, either upon completion of the "MA" word-cycle, if the algebraic sign of the divisor is positive, or upon completion of the "CPA" subcommand if the algebraic sign of the divisor is negative. In either event, as a result of line CB thus being rendered TRUE, a "CB" word-cycle is thereafter executed, whereby a word of all zeros is stored in address-B. Upon completion of the "CB" word-cycle, the state of flipflop **6081** is again reversed, so that line CB is thereby rendered FALSE.

With reference to FIG. 103A, the state of flipflop **6077** is reversed as a result of line CB thus being rendered FALSE, and line AM is rendered TRUE thereby. When line AM is rendered TRUE, line **TS4** is likewise rendered TRUE. Consequently, in response to lines AM and **TS4** being rendered TRUE, an "AM" word-cycle is thereafter executed, whereby the divisor now stored in address-A is read out and stored in memory at the address specified by the contents of section 4 of the instruction register. Upon completion of the "AM" word-cycle, the state of flipflop **6077** is again reversed, so that line AM is rendered FALSE thereby.

When line AM is thus rendered FALSE, line **PR8** (FIG. 103B) is likewise rendered FALSE and thereby causes the R-counter to be preset to a count of "8." Also, the state of flipflop **6091** is again reversed as a result of line AM thus being rendered FALSE, and line MA is again rendered TRUE thereby. As the R-counter is now at a count of "8," so that line **R8** is TRUE, line

TS2 (FIG. 103A) is likewise rendered TRUE as a result of line MA being rendered TRUE. In response to lines MA and TS2 thus being rendered TRUE, a second "AM" word-cycle is thereafter executed, whereby the dividend stored in memory at the address specified by the contents of section 2 of the instruction register is read out and stored in address-A; simultaneously therewith, the dividend is copied digit by digit into the "J" digit-register.

Upon completion of the second "MA" word-cycle, the state of flipflop **6091** (FIG. 103B) is again reversed, so that line MA is rendered FALSE thereby. If upon completion of the "MA" word-cycle the algebraic sign of the dividend, as indicated by the states of lines J9 and (J9)′, corresponds to the algebraic sign of the divisor, as indicated by the states of lines OBM and (OBM)′, the state of flipflop **6096** is reversed, so that line OBN is rendered TRUE thereby. However, if the algebraic signs of the dividend and of the divisor are of opposite signs, the state of line OBN remains FALSE. Thus it is evident that the state of line OBN is rendered TRUE if the algebraic sign of the dividend corresponds to the algebraic sign of the divisor, and is rendered FALSE if the algebraic sign of the dividend does not correspond to the algebraic sign of the divisor. Also, if the algebraic sign of the dividend just read from memory is negative, so that line J9 is rendered TRUE, the state of flipflop **6083** is reversed upon completion of the previously-initiated "MA" word-cycle, and line CPA is thereby rendered TRUE.

As previously described, when line CPA is rendered TRUE, a "CPA" subcommand is thereafter executed, whereby the word stored in address-A is read out and complemented, and its absolute value is thereafter stored back in address-A. Upon completion of the "CPA" subcommand t,he state of flipflop **6083** is again reversed, and line CPA is thereby rendered FALSE. It is evident, therefore, that the state of flipflop **6081** is reversed and line CB thereby rendered TRUE, either upon completion of the secondly-initiated "MA" word-cycle, if the algebraic sign of the dividend is positive, or upon completion of the previously-initiated "CPA" subcommand if the algebraic sign of the dividend is negative. In either event, when line CB is finally rendered TRUE, a "CB" word-cycle is thereafter executed, whereby a word of all zeros is stored in address-B. Upon completion of the "CB" word-cycle, the state of flipflop **6081** is again reversed, so that line CB is thereby rendered FALSE.

With reference to FIG. 103A, the state of flipflop **6100** is reversed as a result of line CB thus being rendered FALSE, and, as a result, line RSB is thereby rendered TRUE. When line RSB is thus rendered TRUE, line TS4 is likewise rendered TRUE. Consequently, in response to lines RSB and TS4 thus being rendered TRUE, a "RSB" subcommand is thereby executed, whereby the divisor stored in memory at the address specified by the contents of section 4 of the instruction register is read out and subtracted from the contents of address-A. Upon completion of the "RSB" subcommand, the state of flipflop **6100** is again reversed, so that line RSB is thereby rendered FALSE. As previously described in detail with respect to that portion of the adder-subtractor circuitry shown in FIG. 54, decimal-carry output line EAS is rendered TRUE upon completion of the "RSB" subcommand if the algebraic sign of the remainder is negative; however, if the algebraic sign of the remainder is positive, line EAS is rendered FALSE.

If it be assumed that the algebraic sign of the remainder now in address-A is positive, so that line (EAS)′ is TRUE, the state of flipflop **6085** (FIG. 103A) is reversed upon completion of the previously-initiated "RSB" subcommand, and line DAD is thereby rendered TRUE. When line DAD is thus rendered TRUE, a "DAD" subcommand is thereafter executed, whereby the constant 0000000001 is added to the word now stored in address-B, which, in this instance, is a word of all zeros. Upon completion of the "DAD" subcommand, the state of flipflop **6085** is again reversed, so that line DAD is rendered FALSE thereby. As a result of line DAD thus being rendered FALSE, the state of flipflop **6100** is again reversed, so that line RSB is rendered TRUE for the second time. Consequently, a second "RSB" subcommand is thereafter executed, whereby the divisor stored in memory at the address specified by the contents of section 4 of the instruction register is again subtracted from the contents of the address-A. Thereafter, the state of flipflop **6100** is again reversed, so that line RSB is rendered FALSE thereby. When line RSB is thus rendered FALSE, the state of flipflop **6085** is again reversed, so that line DAD is rendered TRUE for the second time, provided that the algebraic sign of the difference now in address-A is still positive. However, if it be assumed that the algebraic sign of the difference is now negative, so that line (EAS)′ is rendered FALSE, the state of flipflop **6085** is thereafter prevented from being reversed, and, consequently, line DAD remains FALSE. Therefore, as it has been assumed that the algebraic sign of the remainder is now negative, so that line EAS is rendered TRUE, the state of flipflop **6097** is reversed upon completion of the previously-initiated "RSB" subcommand, and, as a result, line RAD is rendered TRUE thereby. It is evident, therefore, that the subtraction operation is continued until line EAS is rendered TRUE, indicating that the algebraic sign of the remainder is negative.

However, when line RAD is thus rendered TRUE, line TS4 is likewise rendered TRUE. Consequently, a "RAD" subcommand is thereafter executed, whereby the divisor stored in memory at the address specified by the contents of section 4 of the instruction register is added to the contents of address-A, which is thereafter storing a remainder whose algebraic sign is now positive; i.e., greater than "0." Just prior to completion of the "RAD" subcommand, the state of line DR (FIG. 103B) experiences a TRUE-to-FALSE reversal of state and thereby causes the R-counter to be decremented from a count of "8" to a count of "7." Upon completion of the previously-initiated "RAD" subcommand, the state of flipflop **6097** (FIG. 103A) is again reversed, so that line RAD is thus rendered FALSE. As the R-counter is now at a count other than "0," so that line (Rϕ)′ is rendered TRUE, the state of flipflop **6101** is reversed when line RAD is thus rendered FALSE, and, consequently, line SA is rendered TRUE thereby. In response to line SA thus being rendered TRUE, a "SA" subcommand is thereafter executed, whereby the remainder stored in address-A is shifted to the left by one digital position. Thereafter, the state of flipflop **6101** is again reversed, so that line SA is rendered FALSE thereby. When line SA is thus rendered FALSE, the state of flipflop **6103** is reversed and line SB rendered TRUE thereby. In response to line SB being rendered TRUE, a "SB" subcommand is thereafter executed, whereby the partial quotient stored in address-B is also shifted to the left by one digital position. Upon completion of the just-initiated "SB" subcommand, the state of flipflop **6103** is again reversed, and line SB is rendered FALSE thereby.

As a result of line SB thus being rendered FALSE, the state of flipflop **6100** is again reversed, so that lines RSB and TS4 are again rendered TRUE. Consequently, a "RSB" subcommand is thereafter executed, whereby the divisor stored in memory at the address specified by section 4 of the instruction register is subtracted from the shifted remainder now stored in address-A. Thereafter, a succession of "RSB" subcommands is executed until the remainder stored in address-A has been reduced to zero or until the algebraic sign thereof becomes negative, at which time the states of flipflops **6097** and **6100** are both reversed, so that lines RAD and RSB are thereby rendered TRUE and FALSE, respectively. In response to lines RAD and TS4 now being TRUE, a "RAD" subcommand is thereafter executed, whereby the divisor stored in memory at the address specified by section 4 of the instruction register is added to the remainder stored in address-A. Thereafter, a "SA" subcommand followed by a "SB" subcommand is sequentially executed, whereby the words stored in addresses A and B are both shifted one digital position to the left. As shown in FIG. 103B, just prior to the completion of the "RAD" subcommand, the state of line DR experiences a TRUE-to-FALSE reversal of state and thereby causes the R-counter to be decremented from a count of "7" to a count of "6." The above-described sequence of subtraction operations is sequentially repeated until the R-counter is decremented from a count of "1" to a count of "0."

Upon completion of the last-initiated "RAD" subcommand, after the R-counter reaches a count of zero, as indicated by line Rφ being rendered TRUE, the state of flipflop **6079** is reversed, so that lines BM and TS4 are both rendered TRUE. In response to lines BM and TS4 thus being rendered TRUE, a "BM" word-cycle is thereafter executed, whereby the word stored in address-B is read out and stored in memory at the address specified by the contents of section 4 of the instruction register. Upon completion of the "BM" word-cycle, the state of flipflop **6079** is again reversed, so that line BM is rendered FALSE thereby. With reference to FIG. 103B, when line BM is reversed from TRUE to FALSE, the state of line **PR8** is likewise reversed from TRUE to FALSE, and, as a result, the R-counter is again preset to a count of "8," so that line **R8** is rendered TRUE thereby. If it is assumed that line (OBN)′ was previously rendered TRUE, indicating that the algebraic sign of the quotient is negative, the state of flipflop **6084** is reversed upon completion of the previously-initiated "BM" word-cycle, and line CPM is rendered TRUE thereby. As line CPM is thus rendered TRUE, line TS4 (FIG. 103A) remains TRUE. Thus, in response to lines CPM and TS4 being TRUE, a "CPM" subcommand is thereafter executed, whereby the quotient now stored in memory at the address specified by the contents of section 4 of the instruction register is read out and complemented, and the complement thereof stored in the memory address specified by section 4 of the instruction register. Upon completion of the just-initiated "CPM" subcommand, the state of flipflop **6084** is again reversed, so that lines CPM and **TS4** are both rendered FALSE thereby.

As shown in FIG. 103B, if the state of line OBN is TRUE, indicating that the algebraic sign of the quotient is positive, the state of flipflop **6107** is reversed, and line STD is thereby rendered TRUE upon completion of the previously-initiated "BM" subcommand; otherwise, the state of flipflop **6107** is not so reversed until the previously initiated "CPM" subcommand is completed. However, when line STD is finally rendered TRUE, line **TS5** is likewise rendered TRUE. Consequently, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events initiated in response to a "DIV" instruction.

85. *Detailed Description of CFM Instructions*

During the execution of a "CFM" instruction, the values of the two words stored in memory at the addresses specified by the contents of sections 2 and 3 of the instruction register are compared with each other. If the value of the word stored in memory at the address specified by section 2 of the instruction register is equal to or larger than the value of the word stored in memory at the address specified by section 3 of the instruction register, the address of the next instruction word is specified by section 4 of the instruction register. However, if the absolute value of the word stored in memory at the address specified by section 2 of the instruction register is less than the value of the word stored in memory at the address specified by section 3 of the instruction register, the address of the next instruction word is that which is specified by section 5 of the instruction register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107Q, is a step-by-step description of the various previously-described subinstructions to which the computer is sequentially responsive in executing a "CFM" instruction:

Step	Subinstruction	Description
1	MI-0-2	Copy into the instruction register the next instruction word which is stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "15," go to Step 2 in the following subprogram.
2	MA-2-3	Copy the next word from memory into address-A, and, simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the next word from memory being specified by the contents of section 2 of the instruction register; thereafter, go to Step 3.
3	OBM(4)-0-4	Preset the state of line OBM to correspond to the algebraic sign of the word just read from memory; thereafter, go to Step 4.
4	MJ-3-5	Copy the word stored in memory at the address specified by the contents of section 3 of the instruction register digit by digit into the "J" digit register; thereafter, go to Step 5.
5	OBN(4)-0-6	Preset the state of line OBN TRUE if the algebraic sign of the word just read from memory corresponds to the state of line OBM; otherwise, preset the state of line OBN FALSE; thereafter, go to Step 6.
6	OBN(1)-9-7	If the state of line OBN is TRUE, go to Step 7; otherwise, go to Step 9.
7	RSB-3-8	Subtract the word stored in memory at the address specified by the contents of section 3 of the instruction register from the word stored in address-A; thereafter, go to Step 8.
8	EAS-11-10	If the algebraic sign of the difference is positive, go to Step 10; otherwise, go to Step 11.
9	OBM(2)-11-10	If the state of line OBM is TRUE, go to Step 10; otherwise, go to Step 11.
10	STD-4-*	Copy the contents of section 4 of the instruction register into the word-selecting register.
11	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next insruction word is read out from the particular address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "CFM" instruction (i.e., "15"), the state of line CFM is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "CFM" instruction is thereafter executed in the following manner:

With reference to the composite logical diagram shown in FIG. 104, at TIME–1 after line MI is rendered TRUE at the beginning of the previously-initiated "MI" word-cycle, the states of flipflops **6095** and **6096** are reversed, so that lines OBM and OBN are respectively rendered FALSE thereby. When line MIN is subsequently rendered FALSE upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6091** is reversed, and line MA is thereby rendered TRUE. When line MA is thus rendered TRUE, line TS2 is likewise rendered TRUE. Consequently, a "MA" word-cycle is thereafter executed whereby the word stored in memory at the address specified by section 2 of the instruction register is read out and stored in address-A; simultaneously therewith, the word being read out from memory is copied digit by digit into the "J" digit-register. Upon completion of the "MA" word-cycle, if the tenth-order digit of the word just read out from memory is a "9," indicating that the algebraic sign of the word is negative, the state of line OBM remains FALSE, indicative of the negative algebraic sign of the word just read from memory. However, if the algebraic sign of the word just read from memory is positive, so that line (J9)′ is TRUE, the state of flipflop **6095** is reversed just prior to completion of the previously-initiated "MA" word-cycle, so that line OBM is rendered TRUE, indicating that the algebraic sign of the word just read from memory is positive.

Upon completion of the previously-initiated "MA" word-cycle, as indicated by line MA being rendered FALSE, the state of flipflop **6066** is reversed, and lines MJ and TS3 are both rendered TRUE thereby. Consequently a "MJ" word-cycle is thereafter executed, whereby the word stored in memory at the address specified by section 3 of the instruction register is read out and copied digit by digit into the "J" digit-register. Again, if the tenth-order digit of the word just read from memory and copied digit by digit into the "J" digit-register is a "9," indicating that the algebraic sign of the word is negative, the state of line J9 is rendered TRUE; otherwise, line (J9)′ is rendered TRUE. Therefore, if the algebraic signs of the two words just read from memory are the same, the state of flipflop **6096** is reversed upon completion of the previously-initiated "MJ" word-cycle, and, as a result, line OBN is rendered TRUE thereby. However, if the algebraic signs of the two words just read from memory are opposite, the state of line OBN remains FALSE. If it is assumed that the algebraic signs of the two words stored in memory at the addresses specified by sections 2 and 3 of the instruction register are of opposite signs, so that line (OBN)′ is rendered TRUE, the state of flipflop **6107** is reversed upon completion of the previously-initiated "MJ" word-cycle, and, as a result, line STD is rendered TRUE thereby. When line STD is thus rendered TRUE, line TS4 is rendered TRUE if the algebraic sign of the word stored in memory at the address specified by section 2 of the instruction register is positive, as indicated by line OBM being rendered TRUE. However, if the word stored in memory at the address specified by section 2 of the instruction register is negative, so that line (OBM)′ is rendered TRUE instead of line OBM, line TS5 is rendered TRUE thereby. As previously described, when line TS5 is rendered TRUE simultaneously with line STD, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register. However, if line TS4 is rendered TRUE simultaneously with line STD, a "STD" subinstruction is thereafter executed whereby the contents of section 4 of the instruction register is stored in the word-selecting register.

If it be assumed that the words stored in memory at the addresses specified by sections 2 and 3 of the instruction register are of like algebraic signs, so that line OBN is rendered TRUE, the state of flipflop **6100** is reversed upon completion of the previously-initiated "MJ" word-cycle, and, as a result, lines RSB and TS3 are both rendered TRUE thereby. Consequently, a "RSB" subcommand is thereafter executed whereby the word stored in memory at the address specified by section 3 of the instruction register is subtracted from the word stored in address-A Upon completion of the "RSB" subcommand, the state of flipflop **6100** is again reversed, so that lines RSB and TS3 are both rendered FALSE. If, upon completion of the subtraction operation, the word stored in memory at the address specified by section 2 of the instruction register is of greater magnitude than the value of the word stored in memory at the address specified by section 3 of the instruction register, line (PCI)′ is rendered TRUE in the same manner as previously described with respect to FIG. 53. However, if the word stored in memory at the address specified by section 2 of the instruction register is less than the value of the word stored in memory at the address specified by section 3 of the instruction register, line PCI is rendered TRUE instead. Thus, it is seen, if line PCI is rendered TRUE simultaneously with line STD, line TS5 is rendered TRUE in order to effect the storage of the contents of section 5 of the instruction register into the word-selecting register. However, if line (PCI)′ is rendered TRUE simultaneously with line STD, line TS4 is rendered TRUE in order to effect the storage of the contents of section 4 of the instruction register into the word-selecting register.

86. *Detailed Description of CFE Instructions*

During the execution of a "CFE" instruction, the two words stored in memory at the addresses specified by the contents of sections 2 and 3 of the instruction register are first compared for equality. Thereafter, if the values of the two words are of equal magnitude, the contents of section 5 of the instruction register is thereafter stored into the word-selecting register; however, if the values of the two words are of unequal magnitudes, the contents of section 4 of the instruction register is stored into the word-selecting register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107R, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "CFE" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "16," go to Step 2 in the following subprogram.
2	MA-2-3	Copy the next word from memory into address-A, and simultaneously therewith, copy the word digit by digit into the "J" digit-register, the address of the next word from memory being specified by the contents of section 2 of the instruction register; thereafter, go to Step 3.
3	RSB-3-4	Subtract the word stored in memory at the address specified by the contents of section 3 of the instruction register from the word stored in Address-A; thereafter, go to Step 4.
4	OBN(1)-6-5	If the state of line OBN is TRUE, go to Step 5; otherwise, go to Step 6.
5	STD-4-*	Copy the contents of section 4 of the instruction register into the word-selecting register.
6	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and thereafter stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for a "CFE" instruction (i.e., "16"), the state of line CFE is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "CFE" instruction is thereafter executed in the following manner:

With reference to the composite logical diagram shown in FIG. 105, at TIME–1 after line MI is rendered TRUE at the beginning of the previously-initiated "MI" word-cycle the state of flipflop **6096** is reversed, so that line OBN is rendered FALSE thereby. Upon completion of the previously initiated "MI" word-cycle, as indicated by line MIN thereafter being rendered FALSE, the state of flipflop **6091** is reversed, so that lines MA and TS2 are both rendered TRUE thereby. In response to lines MA and TS2 being rendered TRUE, a "MA" word-cycle is thereafter executed during which the word stored in memory at the address specified by section 2 of the instruction register is read out and stored in address-A; simultaneously therewith, the word is also copied digit by digit into the "J" digit-register. Upon completion of the "MA" word-cycle, the state of flipflop **6091** is again reversed, and lines MA and TS2 are both rendered FALSE thereby. When line MA is thus rendered FALSE, the state of flipflop **6100** is reversed, so that lines RSB and TS3 are both rendered TRUE thereby. Consequently, a "RSB" subcommand is thereafter executed, whereby the word stored in memory at the address specified by section 3 of the instruction register is subtracted from the word stored in address-A. Upon completion of the "RSB" subcommand, the state of flipflop **6100** is again reversed, and lines RSB and TS3 are both rendered FALSE thereby.

As previously described with respect to FIG. 54, line ZNϕ from the adder-subtracter is rendered FALSE if the remainder resulting from a subtraction operation is zero; otherwise, line ZNϕ is rendered TRUE

Therefore, if it is assumed that the value of the word stored in address-A and the value of the word stored in memory at the address specified by section 3 of the instruction register are of unequal magnitudes, so that line ZNϕ is rendered TRUE upon completion of the "RSB" subcommand, the state of flipflop **6096** is reversed, and line OBN is rendered TRUE thereby. Also, when line RSB is thus rendered FALSE, the state of flipflop **6107** is reversed, so that line STD is rendered TRUE thereby. In response to lines STD and OBN simultaneously being TRUE, line TS4 is likewise rendered TRUE. Consequently, a "STD" subinstruction is thereafter executed, whereby the contents of section 4 of the instruction register is stored in the word-selecting register. However, if it is assumed that the values of the two words stored in memory at the addresses specified by sections 2 and 3 of the instruction register are of equal magnitudes, as indicated by line ZNϕ being rendered FALSE, the state of line OBN remains FALSE upon completion of the previously-initiated "RSB" subcommand. Consequently, when line STD is rendered TRUE simultaneously with line (OBN)′, the state of line TS5 is likewise rendered TRUE. Consequently, a "STD" subinstruction is thereafter executed whereby the contents of section 5 of the instruction register is stored in the word-selecting register.

87. *Detailed Description of EPT Instructions*

By means of the execution of but a single "EPT" instruction, from one to one hundred words read from punched paper tape are stored in sequentially-numbered addresses in memory. During the first phase of the "EPT" subprogram, the paper tape is caused to be translated in a forward direction, during which time data information is read therefrom if the value of the low-order digit of section 2 of the instruction register is a "0"; however, if the low-order digit is a "1," the paper tape is caused to be translated in a reverse direction (i.e., rewound), during which time only symbol information is read therefrom.

As previously described with respect to FIG. 88C, the total number of words read from the paper tape, to be subsequently stored in sequentially-numbered addresses in memory, is determined by the number of ten-digit words punched in the paper tape between "end-of-frame" (EOF) or "alternate-instruction" (AI) symbols. That is, the words sequentially read from the tape are stored in sequentially numbered addresses in memory until either an "EOF" symbol or an "AI" symbol is detected. When an "EOF" symbol is detected, zeros are automatically stored in the remaining higher-order digital positions of the last address, and the computer thereafter proceeds to carry out the dictates of the next instruction word stored in memory at the address specified by the contents of section 5 of the instruction register. During the tape-reading operation, the first digit read from the tape is stored in the first-order digital position of the address in memory specified by the contents of section 3 of the instruction register; the next digit read is stored in the second-order digital position thereof; and so on, until the address specified by the contents of section 3 of the instruction register is completely filled. Thereafter, section 3 of the instruction register is incremented by a count of "1," and the next ten digits read from the tape are stored in the next-higher-order address in memory, the above-described sequence of events being sequentially repeated until an "EOF" symbol is detected.

However, when an "alternate-instruction" (AI) symbol is detected, zeros are first stored in any remaining digital positions of the last operable address when the "AI" symbol is first detected, and, immediately thereafter, the computer proceeds to carry out the dictates of the next instruction word stored in memory at the address specified by the contents of section 4 of the instruction register. If, however, the digit stored in the low-order digital position of section 2 of the instruction register is a "1" instead of a "0," the paper tape is caused to be translated in a reverse direction until an alternate-instruction symbol is detected, at which time the tape is brought to a standstill, and the computer immediately proceeds to carry out the dictates of the next instruction word stored in memory at the address specified by the contents of section 5 of the instruction register.

Listed below in somewhat tabular form, and also graphically illustrated by the flow diagram of FIG. 107S, is a step-by-step description of the various previously described subinstructions to which the computer is sequentially responsive in executing a "EPT" instruction:

Step	Subinstructions	Description
1	MI-0-2	Copy into the instruction register the next instruction word stored in memory at the address specified by the contents of the word-selecting register; thereafter, if the contents of section 1 of the instruction register is "17," go to Step 2 in the following subprogram.
2	Lϕ-4-3	If the low-order digit stored in section 2 of the instruction register is a "0," go to Step 3; otherwise, go to Step 4.
3	PTF-0-6	Translate the paper tape in a forward direction; thereafter, go to Step 6.
4	LI-0-5	If the low-order digit stored in section 2 of the instruction register is a "1," go to Step 5.
5	PTR-0-6	Translate the paper tape in a reverse direction; thereafter, go to Step 6.
6	TCL-0-7	When the output of the paper tape clock is rendered TRUE, go to Step 7.
7	EOF-9-8	If the digit just read from the punched paper tape is an end-of frame symbol, go to Step 8; otherwise, go to Step 9.
8	PTS-0-18	Stop the paper tape; thereafter, go to Step 18.
9	TCL-0-10	When the output of the paper tape clock is rendered TRUE, go to Step 10.
10	AI-12-11	If the digit just read from the paper tape is an alternate-instruction symbol, go to Step 11; otherwise, go to Step 12.
11	OBM(5)-0-8	Preset the state of line OBM TRUE; thereafter, go to Step 8.
12	Lϕ-6-13	If the low-order digit of section 2 of the instruction register is a "0," go to Step 13; otherwise, go to Step 6.
13	TDA-6-14	If the character just read from the paper tape is a data digit, go to Step 14; otherwise, go to Step 6.
14	AD(1)-0-15	Increment the digit-counter; thereafter, go to Step 15.
15	TDS-3-16	Store the data digit just read from the paper tape into the next higher order digital position of the memory address specified by the contents of section 3 of the instruction register; thereafter, go to Step 16.
16	D9-6-17	If the digit-counter is at a count of "9," go to Step 17; otherwise, go to Step 6.
17	IN4-0-6	Increment section 3 of the instruction register; thereafter, go to Step 6.
18	D9-20-19	If the digit-counter is at a count of "9," go to Step 19; otherwise, go to Step 20.
19	OBM(2)-22-23.	If the state of line OBM is TRUE, go to Step 23; otherwise, go to Step 22.
20	AD(1)-0-21	Increment the digit-counter; thereafter, go to Step 21.
21	ROS-3-18	Store a zero in the memory address specified by the contents of section 3 of the instruction register, the digital position being indicated by the count of the digit-counter; thereafter, go to Step 18.
22	STD-5-*	Copy the contents of section 5 of the instruction register into the word-selecting register.
23	STD-4-*	Copy the contents of section 4 of the instruction register into the word-selecting register.

Upon completion of the execution of the previously initiated "MI" word-cycle, during which time the next instruction word is read out from the particular address in memory specified by the contents of the word-selecting register and stored in the instruction register, if the contents of section 1 of the instruction register thereafter corresponds to the code designation for an "EPT" instruction (i.e., "17"), the state of line EPT is rendered TRUE in the manner previously described with respect to FIG. 56, and, as a result, a "EPT" instruction is thereafter executed in the following manner.

It is to be noted at the outset, however, that the particular sequence of operations carried out during the execution of an "EPT" instruction is dependent, not only upon the contents of section 2 of the instruction register, but also upon the condition of the paper-tape reader when the "EPT" instruction is first initiated, as previously described with respect to that portion of the preceding description relating to the format of an "EPT" instruction word. For example, a different sequence of operations is executed when the "EPT" instruction is first initiated if the contents of section 2 of the instruction register is "00" and a data digit is positioned over the tape reading head; if the contents of section 2 is "00" and an "alternate-instruction" symbol digit is positioned over the reading head; if the contents of section 2 is "00" and an "end-of-frame" symbol digit is positioned over the reading head; if the contents of section 2 is "01" and an "end-of-frame" symbol digit is positioned over the reading head; or if the tape unit is in a rewinding cycle of operation.

Reference is now made to FIGS. 106A and 106B, wherein there is illustrated a composite logical diagram of particular portions of the computer circuitry utilized in executing an "EPT" instruction, and additional reference is also made to FIG. 88C, wherein a fragmentary portion of the particular perforated tape utilized by a conventional paper-tape reader is diagrammatically illustrated.

It is again to be noted that the particular paper-tape-reading mechanism utilized by the present computer is a well-known photo-electric type, commercially available as Model No. 903, at present manufactured by Potter Instrument Company, Inc. The circuitry contained within the paper-tape reader is conventionally connected in such a way that lines T*a* through T*d* are respectively rendered TRUE during a reading operation in response to a perforation located in a corresponding one of data channels #1 through #4; line TDA is rendered TRUE in response to a lack of a perforation in channel #6, indicating the presence of a data digit over the reading head; line EOF is rendered TRUE in response to a perforation located in channel #8; line AI is rendered TRUE in response to a perforation located in channel #7; line TCL is rendered TRUE when a perforation is sensed in clock channel CLK; and line REW is rendered TRUE each time the tape unit is in a rewinding cycle of operation.

At the first TIME–1 after the "EPT" instruction is initiated, the state of flipflop **6061** is reversed, and line TH is thereby rendered TRUE; at the second TIME–1, the state of flipflop **6061** is again reversed, and line TH is thereby rendered FALSE; at the third TIME–1, the state flipflop **6061** is aagin reversed, so that line TH is thereby rendered TRUE; and so on. Each time line TH is rendered TRUE, lines TS3 and JM are likewise rendered TRUE. Therefore, if it is now assumed that "00" is stored in section 2 of the instruction register and that one of the data digits "0" through "9" is positioned over the reading head when the "EPT" instruction is first initiated, the output of gate **1570** experiences a TRUE-to-FALSE reversal of state upon completion of the previously-initiated "MI" word-cycle when line MIN is thereafter rendered FALSE. As a result, the state of flipflop **6087** is reversed, so that line GC is rendered TRUE thereby. However, approximately forty microseconds later, at TIME–1, the state of flipflop **6087** is reversed by gate **1571**, so that line GC is rendered FALSE thereby. Also, when line MIN is rendered FALSE upon completion of the previously-initiated "MI" word-cycle, the state of flipflop **6098** is reversed, so that line ROS is thereby rendered TRUE. After line ROS is rendered TRUE, the state of flipflop **6061** is reversed when line GC is rendered FALSE, and, as a result, line TH is rendered TRUE thereby and remains TRUE for a period of approximately two hundred microseconds, at which time the state of flipflop **6061** is again reversed, so that line TH is thereby rendered FALSE. As shown in FIG. 61, the state of line AD is rendered FALSE at TIME–3 after line TH is rendered TRUE, and thereby causes the digit-counter to be incremented from a count of "9" to a count of "0."

During the two hundred-microsecond interval during which time line TH is TRUE, line MXW (FIG. 60) is rendered TRUE each time the binary bit indicating output of the serializer is indicative of a binary "1" as evidenced by line TDS being rendered TRUE. Consequently, in response to lines TS3 and JM both being rendered TRUE when TH is rendered TRUE, a "JM" digit-cycle is initiated, so that the data digit read from the tape is thereafter stored directly into memory at the first-order digital position of the address indicated by the contents of section 3 of the instruction register. Thereafter, the state of flipflop **6061** (FIG. 106A) is reversed and lines TH, TS3, and JM are rendered FALSE thereby.

It is to be noted that, even though line JM is thus rendered TRUE, any digit previously stored in the "J" digit-register is not stored in memory in the normal manner via the "JM" digit-cycle, due to the fact that the output of gate **1090** is rendered FALSE all during the execution of an "EPT" instruction. Consequently, line JM is rendered TRUE only to cause a digit cycle to be initiated, so that the digit read from the tape is directly stored in memory via gate **1092**, which is controlled by the "EPT" instruction. It is also to be noted that, even though lines PTF and PTS are rendered TRUE and FALSE, respectively, when line ROS is rendered TRUE, thus releasing the tape-braking mechanism and thereby permitting the tape to thereafter be translated in a forward direction at constant velocity, sufficient time elapses before the tape actually begins to move, due to the inertia of the moving parts of the tape reader, to thus allow the data digit initially positioned over the reading head to be read and thereafter stored in the designated address in memory before being moved away from the reading head.

However, when the tape is translated in a forward direction to the point where the second-order digit is positioned over the reading head, line (TCL)′ from the clock channel output circuitry experiences a TRUE-to-FALSE reversal of state. As a result, the state of flipflop **6087** is reversed due to a reversal of state of the output of gate **1569**, and line GC is again rendered TRUE thereby. At TIME–1 thereafter, the state of flipflop **6087** is again reversed, and line GC is rendered FALSE. When line GC is thus rendered FALSE, the state of flipflop **6061** is again reversed and line TH thereafter rendered TRUE for a second two hundred-microsecond period, during which time the second-order digit read from the tape is stored in the second-order digital position of the address in memory specified by the contents of section 3 of the instruction register. As shown in FIG. 61, at TIME–3 after line TH is rendered TRUE for the second time, the state of line AD is again rendered FALSE and thereby causes the digit-counter to be incremented from a count of "0" to a count of "1."

As long as line ROS remains TRUE, the above-described sequence of events is sequentially repeated until the tenth digit read from the tape is stored in the tenth-order digital position of the address in memory specified by the contents of section 3 of the instruction register, at which time the digit-counter is incremented to a count of "9" as indicated by line D9 being rendered TRUE. However, following the third bit-time period after line TH is rendered TRUE, the state of line IN4 is reversed from TRUE to FALSE. As a result, section 3 of the instruction register is incremented by a count of "1." Thereafter, the next group of ten digits read from the tape is stored in memory at the address now specified by the contents of section 3 of the instruction register.

After an end-of-frame or an alternate-instruction symbol is detected, the state of flipflop **6098** is reversed due to a reversal of state of a corresponding one of lines EOF and AI. Consequently, line ROS is rendered FALSE and thereby prevents line MXW (FIG. 60) from thereafter being rendered TRUE. After line (ROS)′ is thus rendered TRUE, the state of flipflop **6061** is continued to be reversed at each TIME–1 and thereby causes line TH to be rendered TRUE for successive two hundred-microsecond periods until the digit-counter reaches a count of "9." However, in the event the digit-counter is already at a count of "9" when a symbol is detected, indicating that the address specified by the contents of section 3 of the instruction register is completely filled with significant digits, the state of flipflop **6061** is prevented by line (D9)′ from being so reversed. Consequently, a zero is stored in each of the higher-order digital positions remaining unfilled in the last-designated address when a symbol is detected.

When line ROS is thus rendered FALSE, line PTF is rendered FALSE and line PTS is simultaneously rendered TRUE to thereby cause immediate energization of the forward tape-breaking mechanism, so that the paper tape is brought to a sudden standstill. Also after line (ROS)′ is rendered TRUE, the state of flipflop **6107** is reversed when line AN is subsequently rendered FALSE, indicating completion of the previously-initiated "JM" digit-cycle. When line STD is thus rendered TRUE, one of lines TS4 or TS5 is simultaneously rendered TRUE, depending upon whether the symbol just detected is an alternate-instruction symbol or an end-of-frame symbol. If the character just detected is an alternate-instruction symbol, the state of flipflop **6095** is reversed, and line OBM is rendered TRUE. However, if the character just detected is an end-of-frame symbol, the state of line OBM remains FALSE.

It is evident, therefore, that when line STD is rendered TRUE, line TS4 is likewise rendered TRUE if an alternate-instruction symbol is detected, whereas line TS5 is rendered TRUE if an end-of-frame symbol is detected. In response to lines STD and TS4 being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 4 of the instruction register is stored in the word-selecting register, whereas, in response to lines STD and TS5 being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register, thus terminating the sequence of events initiated in response to the "EPT" instruction.

It is next assumed that the contents of section 2 of the instruction register is "00" and an alternate-instruction symbol is positioned over the reading head when the "EPT" instruction is first initiated. As before, the state of flipflop **6098** is reversed, so that line ROS is rendered TRUE upon completion of the previously-initialed "MI" word-cycle. As a result of line ROS thus being rendered TRUE, line PTF is rendered TRUE, whereas line PTS is rendered FALSE thereby. Consequently, the forward braking mechanism of the paper-tape reader is de-energized, so that the paper tape is thereafter translated in a forward direction at a constant velocity. However, as soon as the tape is translated forward, so that the alternate-instruction symbol is no longer disposed over the reading head, line AI is rendered FALSE, and, as a result, the state flipflop **6098** is again reversed, so that line ROS is rendered FALSE thereby. In addition, when line AI is thus rendered FALSE, the state of flipflop **6095** is also reversed, so that line OBM is rendered TRUE thereby. In response to line (ROS)′ initially being rendered FALSE, the state of flipflop **6107** is reversed, so that line STD is rendered TRUE thereby. Therefore, when line OBM is subsequently rendered TRUE as a result of the alternate-instruction symbol being moved away from the reading head, line TS4 is likewise rendered TRUE. In response to lines STD and TS4 being rendered TRUE, a "STD" subinstruction is thereafter executed, whereby the contents of section 4 of the instruction register is stored in the word-selecting register. In addition, line PTF is rendered FALSE and line PTS is rendered TRUE as a result of line ROS being rendered FALSE. Consequently, the forward braking mechanism is again energized in a manner such that the paper tape is brought to a sudden standstill.

It is next assumed that, when the "EPT" instruction is first initiated, the contents of section 2 of the instruction register is "00" and an end-of-frame symbol is disposed over the reading head. As before, the state of flipflop **6098** is reversed when line MIN is rendered FALSE upon completion of the previously-initiated "MI" word-cycle, so that line ROS is rendered TRUE thereby. As a result of line ROS thus being rendered TRUE, line PTF is likewise rendered TRUE, whereas line PTS is rendered FALSE. Consequently, the forward braking mechanism is de-energized, so that the tape is thereafter caused to be translated in a forward direction at a constant speed, during which time the data information read from the tape is stored in memory in the same manner as previously described. When an alternate-instruction symbol is subsequently detected, the states of both of flipflops **6095** and **6098** are reversed, so that line OBM is rendered TRUE and line ROS is rendered FALSE thereby. As a result of line ROS being rendered FALSE, line PTS is rendered TRUE, whereas line PTF is rendered FALSE. Consequently, the forward braking mechanism is energized, so that the tape is brought to a standstill. When line AN is subsequently rendered FALSE after line (ROS)′ is rendered TRUE, the state of flipflop **6107** is reversed, so that line STD is rendered TRUE. As line OBM is also TRUE at this time, line TS4 is rendered TRUE when line STD is thus rendered TRUE. Thereafter, a "STD" subinstruction is executed, whereby the contents of section 4 of the instruction register is stored in the word-selecting register. It is to be noted that, if an end-of-frame symbol had been detected, instead of an alternate-instruction symbol, the state of flipflop **6098** is reversed, as before, and thereby causes the forward braking mechanism to be energized. However, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register, instead of section 4, is stored in the word-selecting register.

It is next assumed that, when the "EPT" instruction is first initiated, the contents of section 2 of the instruction register is "01" and a data digit is disposed over the reading head. Therefore, when line ROS is rendered TRUE upon completion of the previously-initialed "MI" word-cycle, as before, line PTR is rendered TRUE and line PTS is rendered FALSE. Consequently, the reverse tape braking mechanism is de-energized, so that the tape is thereafter translated in a reverse direction at a maximum constant speed. After the tape is being translated in a reverse direction, line GC is sequentially being rendered TRUE and FALSE by flipflop **6087**, as before. However, due to the fact that the low-order digit of section 2 of the instruction register is now a "1", so that the state of line Lϕ is FALSE, the state of line TH remains FALSE and thereby prevents any data read from the tape from being stored in memory at the address specified by section 3 of the instruction register. When an end-of-frame symbol is subsequently detected, the state of flip-flop **6098** is again reversed, and line ROS is rendered FALSE thereby. When line ROS is thus rendered FALSE, line PTR is likewise rendered FALSE, while line PTS remains FALSE and the tape continues to be rewound. Approximately forty microseconds after line (ROS)′ is rendered TRUE, the state of flipflop **6107** is reversed, and lines STD and TS5 are both rendered TRUE thereby. Consequently, a "STD" subinstruction is thereafter executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register.

As previously mentioned, line REW is rendered TRUE each time the tape is being translated in a reverse direction. Consequently, while the tape is being translated in a reverse direction and the computer is in the process of executing another instruction than "EPT," if an alternate-instruction symbol is subsequently detected, so that line AI is rendered TRUE, line PTS is rendered TRUE to effect immediate energization of the reverse braking mechanism and thereby cause the tape to be brought to a sudden standstill.

When the "EPT" instruction is first initiated, if the contents of section 2 of the instruction register is "01" and an end-of-frame symbol is disposed over the reading head, line ROS is rendered TRUE, as before, upon completion of the previously-initiated "MI" word-cycle. When line ROS is thus rendered TRUE, line PTR is likewise rendered TRUE, and, as a result, the reverse braking mechanism is de-energized, so that the paper tape is thereafter translated in a reverse direction, as before. When an end-of-frame symbol is thereafter detected, the state of flipflop **6098** is again reversed, and lines ROS and PTR are both rendered FALSE. Approximately forty microseconds after line ROS is rendered FALSE, lines STD and TS5 are rendered TRUE by a reversal of state of flipflop **6107**. As a result, the contents of section 5 of the instruction register is stored in the word-selecting register. The paper tape, however, continues to be translated in a reverse direction until an alternate-instruction symbol is detected, at which time line PTS is rendered TRUE. As previously described, as a result of lines PTS and PTR being rendered TRUE and FALSE, respectively, the reverse braking mechanism is energized and thereby causes the paper tape to be brought to a sudden standstill.

It will next be assumed that, when the "EPT" instruction is first initiated, the contents of section 2 of the instruction register is "**01**" and an alternate-instruction symbol is disposed over the reading head. As before, line ROS is rendered TRUE by a reversal of state of flipflop **6098** when line MIN is rendered FALSE upon completion of the previously-initiated "MI" word-cycle. When line ROS is rendered TRUE, line PTR is rendered TRUE and thereby causes de-energization of the reverse braking mechanism, so that the paper tape is thereafter translated in a reverse direction.

When an end-of-frame symbol is subsequently detected, the state of flipflop **6098** is reversed, so that lines ROS and PTR are both rendered FALSE thereby. Approximately forty microseconds after line ROS is rendered FALSE, the state of flipflop **6107** is reversed, so that lines STD and TS5 are both rendered TRUE. Thereafter, a "STD" subinstruction is executed, whereby the contents of section 5 of the instruction register is stored in the word-selecting register. Upon subsequent detection of an alternate-instruction symbol, line PTS is rendered TRUE and thereby causes energization of the reverse braking mechanism, whereby the tape is brought to a sudden standstill.

Finally, it is assumed that the paper tape mechanism is in a rewinding cycle of operation when the "EPT" instruction is first initiated and "00" is stored in section 2 of the instruction register. As before, the state of flipflop **6098** is reversed and line ROS rendered TRUE when line MIN is subsequently rendered FALSE upon completion of the previously-initiated "MI" word-cycle. When an alternate-instruction symbol is thereafter detected, the states of flipflop **6098** and **6095** are both reversed, so that lines ROS and OBM are rendered FALSE and TRUE, respectively. However, approximately forty microseconds after line ROS is rendered FALSE, the state of flipflop **6107** is reversed, so that lines STD and TS4 are both rendered TRUE thereby. As a result, the contents of section 4 of the instruction register is thereafter stored in the word-selecting register. Also, when the alternate-instruction symbol is detected, line PTS is rendered TRUE and thereby causes the reverse braking mechanism to be energized and the tape to be brought to a sudden standstill.

88. *Detailed Description of Paper Tape Punching Operation*

There have previously been described in detail various modes of operations capable of being executed by the computer whereby words indexed into the accounting machine keyboard, or taken from the totalizers therein, are subsequently stored in selected addresses in memory via execution of an "EKW" instruction; whereby words stored in selected addresses in memory are subsequently printed out on recording media via execution of a "POW" instruction; whereby words stored in selected addresses in memory are magnetically recorded on a ledger card via execution of a "ROC" instruction; whereby words are magnetically read from a ledger card and subsequently stored in selected addresses in memory via execution of an "ECW" instruction; and whereby words are read from punched paper tape and subsequently stored in selected addresses in memory via execution of an "EPT" instruction.

There will now be described another mode of operation capable of being executed by the computer whereby words stored in selected addresses in memory, or either indexed into the keyboard or taken from the totalizers of the accounting machine, are subsequently recorded on punched paper tape.

It is assumed that, prior to the initiation of a punching cycle of operation, the carriage of the accounting machine is stationary and is located in a columnar position such that plate **214** (FIG. 13), hence switch-operating studs **215** and **216**, is positioned adjacent switch plungers **217** and **218**, as illustrated by the dotted lines therein. It is further assumed that the paper-tape-punching mechanism (FIG. 44H) is properly loaded with a suitable supply of paper tape, and that switch **ST43** (FIG. 86) has previously been closed by the operator, thus causing the direct-current power supply, which supplies the necessary unidirectional operating potential to the control circuitry of the paper-tape-punching mechanism, to be energized thereby in a well-known manner. Upon the machine carriage reaching the just-assumed position, the forward extension of plate **214** (FIG. 13) engages the roller portion of the arm **153** in a manner such the arm **153** is deflected counter-clockwise, as viewed, to the extent that switch **SC41** is actuated thereby. When switch **SC41** is thus actuated, contacts **SC41—1** and **SC41—2** (FIG. 86) are closed, and contacts **SC41—3** (FIG. 82) are opened thereby.

It will now be assumed that a cycle of operation of the accounting machine has just been initiated in response to an enter-keyboard-words (EKW), print-out-words (POW), or motor bar (MB) instruction, so that the amount racks are subsequently differentially positioned indicative of the word read out from memory, or just entered in the keyboard, of taken from a totalizer of the accounting machine.

With continued reference to FIG. 86 and also to the timing chart shown in FIG. 87C, after a cycle of operation of the accounting machine is initiated, so that the main cam shaft **588** of the machine is rotated clockwise, as viewed in FIG. 44B, through an angular distance of approximately 126 degrees from the position shown, the contacts of switch **SC49** are closed due to the engagement of the cam **592** with the actuator thereof. Closure of the contacts of switch **SC49** completes the circuit through solenoid **L47** (FIG. 86) and thus effects energization thereof. With reference to FIG. 13, as a result of solenoid **L47** thus being energized, its armature is caused to be rotated counter-clockwise, as viewed, thereby causing the stud **226** to deflect the lever **223** clockwise and thus shift the switch basket upwardly until selected ones of the plungers **217** and **218** are respectively engaged with predetermined ones of the studs **215** and **216**. As a result of such engagement, predetermined ones of the plungers **218** in the rightmost row and predetermined ones of the plungers **217** in the leftmost row are simultaneously depressed and latched in their downwardly-deflected positions, and, additionally, remaining ones of any previously-depressed and -latched plungers are caused to be released simultaneously therewith. After the main cam shaft of the machine is rotated through an angular distance of approximately 168 degrees, the movable arm of switch SC**48** (FIG. 44B) is effectively deflected downwardly due to the engagement of the cam **592** with its actuator. When the movable arm of switch SC**48** is deflected downwardly from the position shown in FIG. 86, solenoid L47 is de-energized, whereas solenoids L**45** and L**46** are both caused to be energized thereby. With reference back to FIG. 13, when solenoid L47 is thus de-energized, the switch basket is spring-urged back to its initial starting position, as shown. As a result of the basket return, the electrically-non-conductive tip **228**, carried on the lowermost end of each of the plungers **217** and **218**, engages and thus deflects downwardly the topmost one of a corresponding stack of switch blades **229**, which stacks are arranged in twenty rows, with each row corresponding to a different one of the plungers **217** and **218**. It is appreciated, of course, that there are ten of the plungers **217** arranged in a leftmost row and there are ten of the plungers **218** arranged in a rightmost row. When the uppermost one of the switch blades **229** is deflected downwardly, all of the switch blades located in that particular stack are electrically connected together until the solenoid L47 is again energized and thereby again causes the switch basket to be moved upwardly in the same manner as previously described.

With reference to FIG. 45D, there is illustrated therein a block diagram of the electrical circuitry which is utilized by the paper-tape-punching mechanism and which is substantially identical with that shown and described in co-pending United States patent application Serial No. 634,260, filed by Richard C. Simmerman and Melvin T. Roudebush on January 15, 1957, and assigned to the present assignee, now United States Patent No. 2,922,141, issued January 19, 1960. The tape-punch control circuitry, the rear form bar circuitry, and the rack read-out circuitry are herein shown and described in detail with respect to FIGS. 86, 13, and 3B, respectively. Due to the fact that a detailed description of the program selection, serializer, and punch magnet control circuitry is to be found in the just-mentioned co-pending application, a further detailed description thereof is not deemed necessary to again be given herein. Suffice it to say that the pattern of selective closure of the switches **229** (FIG. 13) causes a predetermined tape-punching program within the paper tape recorder to be selected in the same manner as described in the just-mentioned co-pending application.

Just prior to de-energization of solenoid L47, the various amount racks are assumed to be differentially positioned indicative of a word just read from memory, just entered from the accounting machine keyboard, or just taken from a selected one of the totalizers of the machine, all in the same manner as previously described with respect to the "EKW," "POW," and "MB" instructions, so that the aligner bar **72** (FIG. 3A) is in engagement with the aligner notches formed on the lower end of the member **58**. Thus, with reference now to FIGS. 3B and 9, when the solenoids L**45** and L**46** are energized, the shaft **197** is rotated clockwise, as viewed in FIG. 3B, so that the switch basket **171** is lowered thereby to a position sufficient to cause engagement of the lowermost end of one of the spring clip members **184** in each of the ten columns thereof, as illustrated in FIG. 9, with a corresponding raised cam surface **205**, which is formed on the topmost side of the rightmost end of each of the extensions of the amount racks. Consequently, due to such engagement of the spring clip members **184** with the cams **205**, corresponding ones of the members **184**, one in each column, are latched in an upwardly-deflected position, all in the manner previously described. Thereafter, the latched clip members collectively represent the differentially-set digital valued positions of the amount racks.

After the main cam shaft of the accounting machine is rotated an angular distance of approximately 182 degrees, the actuator of the switch SC49 (FIG. 44B) is released from engagement with the cam **592**. As a result of the release of the actuator of the switch SC**49**, the contacts thereof are thereby opened. With reference back to FIG. 86, when the contacts of switch SC**49** are opened, as shown, solenoids L**45** and L**46** are de-energized thereby. As a result of solenoids L**45** and L**46** thus being de-energized, the lower switch basket **171** (FIG. 3B) is caused to be spring-urged upwardly, so that all latched ones of the clipmembers **184** are thereby brought into engagement with, and thus short-circuit, corresponding ones of the spring-clip members **177**, all in the same manner as previously described with respect to FIG. 12.

When the main cam shaft of the accounting machine is rotated an angular distance of approximately 213 degrees, the cam **593** (FIG. 44G) engages the actuator of switch SC**42**, so that its normally-opened contacts are closed thereby. With reference to FIG. 86, due to the fact that relay solenoid K**3** is normally energized, so that contacts K**3**—**3** are closed thereby, relay solenoid K**1** is caused to be energized when the contacts of switch SC**42** are thus closed. When relay solenoid K**1** is energized, contacts K**1**—**1** are opened and thereby cause a 300-ohm resistor to be connected in series with relay solenoid K**1**, contacts K**1**—**2** are closed, contacts K**1**—**3** are opened, contacts K**1**—**4** are closed, contacts K**1**—**5** are opened and thereby cause relay solenoid K**3** to be de-energized, contacts K**1**—**6** are closed and thereby maintain relay solenoid K**1** energized, and contacts K**1**—**7** (FIG. 82) are opened and thereby cause the state of line GO to be rendered FALSE, thus indicating that a punching cycle of operation is just initiated. At approximately 264 degrees of rotation of the main cam shaft, the actuator of switch SC**42** (FIG. 44G) is released by the cam **593**, so that the contacts of switch SC**42** are allowed to open as illustrated in FIG. 86.

After relay solenoid K**1** is energized, the accounting machine independently continues to complete its cycle of operation in the same manner as previously described. However, in response to relay solenoid K**1** thus being energized, a cycle of operation of the punching mechanism is thereafter executed in the same manner as described in detail in the before-referred-to co-pending application Serial No. 634,260, whereby the amount collectively represented by the short-circuited conditions of the spring clip members **177** is caused to be recorded on the paper tape via the paper-tape-punching mechanism shown and described in detail with respect to FIG. 44H.

If it is assumed that the word indexed into the accounting machine keyboard, as collectively represented by the short-circuited conditions of the spring clip members **177** (FIG. 3B), is a negative amount, as indicated by a depression of the reverse-entry key, the movable arm of switch SC46 (FIG. 44A) is actuated, so that its normally opened contacts are closed upon depression of the reverse entry key REV. With reference to FIG. 86, as the contacts of switch SC**46** are thus closed prior to the closure of the contacts of switch SC**42**, relay solenoid K**5** is energized simultaneously with relay solenoid K**1** when the contacts of switch SC**42** are subsequently closed in the manner just described. When relay solenoid K**5** is thus energized, contacts K**5**—**1** are opened, whereas contacts K**5**—**2** are closed and thereby maintain relay solenoid K**5** energized even though contacts SC42 are subsequently opened. In response to relay solenoid K5 thus being energized, a symbol representing an opposite algebraic sign to that which was program-selected is punched in the paper tape.

However, if it is assumed that the word stored in memory to be subsequently read out and recorded on punched paper tape is the complement of a negative number, a different sequence of events from that just described is initiated. As previously described with respect to FIG. 37, a credit-balance cycle of operation is automatically initiated when solenoid CBS is selectively energized. Also, when credit balance key "CR. BAL." (FIG. 2) is depressed, a credit-balance operation is carried out when a cycle of operation of the accounting machine is subsequently initiated.

As previously described with respect to FIGS. 44D and 44F, when a credit-balance cycle of operation is initiated and the main cam shaft thereafter rotated an angular distance of approximately thirty degrees, the overdraft cam shaft **601** is coupled through the gear **604** to the main cam shaft **588** in such a manner that the overdraft cam shaft is thereafter rotated at exactly one half the speed of the main cam shaft. Thus, when a machine cycle is initiated and the overdraft cam shaft **601** thereafter is rotated an angular distance of approximately ten degrees, the normally-closed contacts of switch SC45 (FIG. 44C) are released by the cam **612** and are thus opened thereby. With reference to FIG. 86, when the contacts of switch SC45 are opened, relay solenoid K1 is prevented from being energized and thereby initiating a tape-recording operation during the first cycle of the credit-balance operation of the accounting machine. However, after the overdraft cam shaft is rotated through an angular distance of approximately 165 degrees, during which time the main cam shaft is rotated one complete revolution, the contacts of switch SC45 are caused to be closed by the cam **612** and remain closed during the second cycle of the credit-balance operation of the accounting machine. Thus, relay solenoid K1 is allowed to be energized in the manner previously described only during the second cycle of operation of the accounting machine when a credit-balance operation is being carried out.

After the overdraft cam shaft **601** is rotated an angular distance of approximately 260 degrees, the normally-opened contacts of switch SC47 (FIG. 44E) are caused to be closed by the cam **613**. With reference to FIG. 86, when the contacts of switch SC47 are closed, relay solenoid K5 is thereafter permitted to be energized when contacts SC42 are subsequently closed and thereby cause energization of relay solenoid K1 in the same manner as previously described. Energization of relay solenoid K5 causes contacts K5—2 to be closed and thereby maintain relay solenoid K5 energized, as before. After relay solenoid K5 is energized, the cycle of operation subsequently carried out is exactly the same as just described with respect to the "reverse-entry" cycle of operation. After the overdraft shaft is rotated an angular distance of approximately 290 degrees, relay solenoid K5 remains energized even though the contacts of switch SC47 are caused to be opened by the cam **613**.

89. *Operator Controls*

As previously mentioned with respect to FIG. 1, located on the front of the right-hand portion of the computer cabinet is a control panel **15**, having mounted thereon four push button controls, Sϕ1—Sϕ2, S11—S12, SP1—SP2, and RS1, each of which, when depressed, initiates a particular cycle of operation of the computer. That is, when either of the four just-mentioned push buttons is depressed, the particular cycle of operation associated with that particular push button is thereafter carried out, regardless of the particular portion of the program the computer is currently in the process of executing.

For example, when push button Sϕ1—Sϕ2 is depressed, the word-selecting register is preset to "00" via line PWϕ. Immediately thereafter, the word stored in memory address "ϕϕ" is read out and stored in the instruction register, and the particular cycle of operation dictated by that particular instruction word is thereafter executed. When push button S11—S12 is depressed, the word-selecting register is preset to "01," via line PW1, and the particular cycle of operation dictated by the word stored in memory address "ϕ1" is immediately executed thereafter. However, when push button SP1—SP2 is depressed, the instruction register is preset to 0000009900. Immediately thereafter, the particular cycle of operation dictated by the contents of the instruction register is executed; i.e., "Enter-keyboard-words into memory addresses ϕϕ through 99 with an upper motor bar 'touch' operation and a normal decimal-point lamp illumination." When reset push button RS1 is depressed, the states of substantially all of the control-counter flipflops are simultaneously so conditioned that the reference outputs thereof are thereafter rendered FALSE via line (RS)'. Thereafter, the computer remains in a standby or neutral condition until one of push buttons Sϕ1—Sϕ2, S11—S12, or SP1—SP2 is subsequently depressed. Also, as previously described, when the computer is first turned "ON," a neutral or standby condition is effected by line (RS)'. After the computer is in a neutral or standby condition, depression of one of push buttons Sϕ1—Sϕ2, S11—S12, or SP1—SP2 is necessary in order to initiate a subsequent cycle of operation thereof.

Located on the right side of the control panel **15** are two additional push buttons, AT1—AT2 and MN1—MN2, which, when depressed, respectively condition the computer for a subsequent automatic or a manual mode of operation. For example, when automatic-mode push button AT1—AT2 is depressed and locked, the computer thereafter successively carries out the dictates of one instruction word after another in a fully automatic manner. That is, upon the execution of each instruction, the next successive instruction is executed without substantially any delay therebetween. When manual-mode-selection push button MN1—MN2 is depressed, the automatic-mode-selection push button, if depressed, is released, and the word stored in memory at the address specified by the contents of the word-selecting register is thereafter read out and stored in the instruction register, and the particular cycle of operation dictated by that particular instruction word is thereafter executed. Upon the execution of that particular instruction, all subsequent operations within the computer cease until the manual or automatic mode-selection push button is thereafter depressed. Thus, the manual mode-selection push button is capable of being utilized whenever it is desired for the computer to proceed through a program in a step-by-step manner.

Also located on the control panel **15** are five columns of lamps, with four lamps per column, which provide a visual indication, in binary-coded-decimal form, of the two-decimal-digit numbers stored in sections 1 and 5 of the instruction register. For example, counting from left to right, the lowermost lamps in columns #1 and #2 are simultaneously illuminated whenever the contents of section 1 of the instruction register is "11"; the lowermost lamp in column #1 and the uppermost lamp in column #2 are simultaneously illuminated whenever the contents of section 1 of the instruction register is "18"; the two uppermost lamps in columns #4 and #5 are simultaneously illuminated whenever the contents of section 5 of the instruction register is "88"; the two uppermost and the two lowermost lamps in columns #4 and #5 are simultaneously illuminated whenever the contents of section 5 of the instruction register is "99"; and so on.

With the exception of push button RS1, which has previously been shown and described in detail with respect to FIG. 76, there is logically illustrated in FIG. 84 the electrical circuitry associated with each of the just-described push buttons. When either one of the push buttons is depressed, the state of a corresponding one of flipflops **6134** through **6139** is reversed, so that the state of the reference output line therefrom is rendered TRUE. However, when that particular push button is released, the state of the corresponding flipflop is again reversed, so that the reference output therefrom is rendered FALSE. In view of the foregoing detailed description of the logical circuitry associated with each of the various instructions, it is readily determinable that, when the reference output of any one of flipflops **6134** through **6139** experiences a TRUE-to-FALSE reversal of state, the particular cycle of operation associated with the corresponding push button is thereafter executed, whereby the before-stated end results are attained; consequently, a further detailed description of the cycle of operation initiated by depression of each of the foregoing push buttons is not deemed necessary in order to insure a full and complete comprehension of the various novel aspects of the present invention.

90. *Detailed Description of a Payroll Program Application*

In order to facilitate a more complete understanding and appreciation of the overall features and functions of the various aspects of the present computer as to its adaptability and applicability to be effectively utilized in substantially any of the various record-keeping systems, an exemplary application with respect to a typical payroll record-keeping and check-writing operation will next be described.

It is first assumed that a typical employee's earnings record or ledger card, such as that illustrated in FIG. 88A, is to be utilized and, in addition to the two posting lines of information previously printed on its face, as illustrated, also has magnetically recorded on the ferromagnetic strip **306** the employee's clock number, earnings rate, number of dependents, union dues, community chest deduction, bond deduction, credit union deduction, insurance deduction, annuity deduction, total gross earnings to date, total Federal tax to date including income tax and F.I.C.A., total city tax to date, total annuity to date, total F.I.C.A. deductions to date, total earnings to date at the end of the previous quarter, and "proof total." It is further assumed that it is desired for the computer to electronically calculate the information identified by the columnar headings of the ledger card and to automatically cause the information to be printed out on the card below the second illustratively-shown posting line. It is also assumed that it is desired for the above information to be simultaneously posted on a journal sheet and that the net amount of the employee's computed pay be automatically printed out on a payroll check; after the results are calculated and printed out, the ledger card is automatically ejected, and the computer is thereafter properly conditioned so as to be ready to receive the next employee's ledger card, which is manually fed thereinto.

It is further assumed, in accordance with standard programming techniques, that the particular arithmetic operations necessary to derive the necessary mathematical results of such a payroll application are determined; that all arithmetic data-input and data-output format-control steps are arranged in a logical sequence as exemplified in the following chart; and that the literal instructions are encoded into computer language in the form of corresponding instruction words, also exemplified in the following chart, with each instruction word having a format corresponding to that previously described in detail.

Instruction word symbol	Literal instruction	Memory address location	Instruction and data words				
SHF	Access the initializing routine.	00	04	00	00	00	70
ECW	Store in memory the words read from the ledger card.	01	02	63	70	85	02
SUM	Summarize the words read from the ledger card for proof.	02	10	95	70	84	03
CFE	Compare the sum with the proof total.	03	16	95	85	01	04
EKW	Enter clock number from time card.	04	00	40	95	95	51
CFE	Compare clock numbers	05	16	85	70	06	07
MB	Return to home position	06	06	30	00	00	01
POW	Print out Program Number	07	01	20	94	94	08
MDD	Multiply hourly earnings rate by overtime factor.	08	12	71	63	99	52
EKW	Enter regular hours, overtime hours, other earnings, miscellaneous deductions.	09	00	20	95	98	10
MUS	Multiply earnings rate by regular hours.	10	13	35	71	95	11
MUS	Multiply overtime rate by overtime hours.	11	13	35	99	96	12
POW	Print out regular earnings and overtime earnings.	12	01	20	95	96	13
MDD	Multiply number of dependents by 13.00.	13	12	72	55	96	14
EKW	Print out gross earnings (sub-total "X" totalizer).	14	00	23	95	95	15
ADD	Add gross earnings to total gross earnings to date.	15	08	95	79	79	16
SUB	Subtract tax credit from gross earnings.	16	09	95	96	96	17
CFM	Determine if adjusted gross is equal to or greater than zero.	17	15	96	62	19	18
SHF	Copy .00 for withholding tax.	18	04	00	62	96	20
MDD	Multiply adjusted gross by tax rate.	19	12	96	56	96	20
SHF	Copy gross earnings	20	04	00	95	97	21
MUS	Multiply gross earnings by F.I.C.A. rate.	21	13	35	57	95	22
ADD	Add F.I.C.A. deduction to F.I.C.A. to date.	22	08	83	95	99	23
CFM	Determine if new F.I.C.A. to date is equal to or greater than $120.00.	23	15	99	58	24	25
SUB	Subtract F.I.C.A. to date from $120.00.	24	09	58	83	95	25
ADD	Add F.I.C.A. deduction to F.I.C.A. to date.	25	08	95	83	83	26
ADD	Add withholding tax to F.I.C.A. deduction.	26	08	96	95	96	27
POW	Print out Federal tax deduction.	27	01	20	96	96	28
ADD	Add Federal tax to Federal tax to date.	28	08	96	80	80	29
MUS	Multiply gross earnings by city tax rate.	29	13	35	59	97	30
POW	Print out city tax	30	01	20	97	97	31
ADD	Add city tax deduction to city tax to date.	31	08	97	81	81	31
POW	Print out union dues and community chest deductions.	32	01	20	73	74	36
MB	Move accounting machine carriage to column #18.	33	06	40	00	00	35
POW	Print out bond and credit union deductions.	34	01	20	75	76	38
POW	Print out insurance and annuity deductions.	35	01	20	77	78	40
ADD	Add union dues deduction to union dues to date.	36	08	73	86	86	37
ADD	Add Community Chest deduction to Community Chest deductions to date.	37	08	74	87	87	35
ADD	Add bond deduction to bond deductions to date.	38	08	75	88	88	39
ADD	Add credit union deduction to credit union deductions to date.	39	08	76	89	89	35
ADD	Add insurance deduction to insurance deductions to date.	40	08	77	90	90	41
ADD	Add annuity deduction to annuity deductions to date.	41	08	78	91	91	42
POW	Print out miscellaneous deduction.	42	01	20	98	98	43
ADD	Add miscellaneous deduction to miscellaneous deductions to date.	43	08	98	92	92	44
ADD	Add annuity deduction to annuity deductions to date.	44	08	78	82	82	45
POW	Print out four totals to date	45	01	20	79	82	46
ADD	Add advance factor to check number.	46	08	93	61	93	47
MB	Print out amount of net pay (total "X" totalizer).	47	06	20	00	00	48
POW	Print out check number	48	01	10	93	93	49
SUM	Summarize words for proof total.	49	10	85	70	84	50

Instruction word symbol	Literal instruction	Memory address location	Instruction and data words				
ROC----	Records words on ledger card.	50	03	00	70	85	01
SHF-----	Shift clock number two places to the right.	51	04	42	95	85	05
SHF-----	Shift clock number eight places to the left.	52	04	28	95	95	53
CFE----	Determine if the exception alert digit is equal to zero.	53	16	95	62	67	54
SHF-----	Copy forty hours into temporary storage address.	54	04	00	60	95	65
	Withholding tax exemption factor; i.e., $13.00.	55	00	00	00	13	00
	Withholding tax rate; i.e., 18%.	56	00	00	00	00	18
	F.I.C.A. rate; i.e., 2.5%-----	57	00	00	00	25	00
	F.I.C.A. maximum; i.e., $120.00.	58	00	00	00	120	00
	City tax rate; i.e., 0.5%------	59	00	00	00	05	00
	Constant of 40 hours--------	60	00	00	00	40	00
	Check number advance factor; i.e., 1.	61	00	00	00	01	00
	Constant "0"---------------	62	00	00	00	00	00
	Overtime factor; i.e., 1.5-----	63	00	00	00	01	50
CMA----	Clear miscellaneous deduction address.	64	05	00	98	98	10
POW----	Print out forty-hour constant.	65	01	20	60	60	66
MB-----	Move accounting machine carriage to column #11.	66	00	43	96	96	64
CFE-----	Determine if exception alert digit is equal to "1."	67	16	95	68	09	69
	Constant "1"---------------	68	10	00	00	00	00
EKW---	Enter regular hours and overtime hours into keyboard.	69	00	40	95	96	64
EKW---	Enter last check number and program number into keyboard.	70	00	00	93	94	71
SHF-----	Shift the program number two places to the right.	71	04	42	94	95	72
ADD----	Add program number to contents of address-31.	72	08	95	31	31	73
CMA----	Clear accumulation addresses.	73	05	00	86	92	01

Memory Addresses:

Words read from ledger card and stored in memory:

Address	Contents
70	Clock No.
71	Rate.
72	No. of Dep.
73	Union Dues Ded.
74	Comm. Chest Ded.
75	Bond Ded.
76	Credit Union Ded.
77	Insurance Ded.
78	Annuity Ded.
79	Earnings to date.
80	Fed. Tax to date.
81	City Tax to date.
82	Annuity to date.
83	F.I.C.A. to date.
84	Prev. Qtr. Earnings to date.
85	Proof Total.
86	Union Dues Accum.
87	Comm. Chest Accum.
88	Bond Accum.
89	Credit Union Accum.
90	Insurance Accum.
91	Annuity Accum.
92	Misc. Accum.
93	Last Check No.
94	Program No.
95	Temporary Storage.
96	Temporary Storage.
97	Temporary Storage.
98	Temporary Storage.
99	Temporary Storage.

The instruction words initially stored in memory addresses 7ϕ through 73 are utilized only one time in the program; i.e., at the beginning thereof. After the contents of memory addresses 7ϕ through 73 are once utilized, the addresses are thereafter utilized for subsequent storage of data words therein, as illustrated above.

It is further assumed that the journal sheet and the pay-check-statement combination (FIGS. 88F and 88G) are properly positioned in the accounting machine carriage, ready for a posting operation thereon, as illustrated in FIGS. 88D and 88E; that the just-mentioned forms either are composed of suitable carbonless paper, such as that currently manufactured by the present assignee as "N C R" paper, or have disposed therebetween suitable transfer means in accordance with conventional practice; that, in accordance with the teachings of the previously-referred-to United States Patent No. 2,626,-749, the various carriage stops are each assembled in the manner diagrammatically illustrated in FIGS. 88D and 88E and are properly positioned on the front form bar of the accounting machine corresponding to the columnar format of the journal and pay-check-statement disposed in the carriage, and also corresponding to the columnar format of the ledger card which is subsequently to be additionally positioned in the carriage; and that an automatic mode of operation of the computer is desired, as evidenced by push button AT1—AT2 (FIG. 1) being initially depressed by the operator.

Assuming that all of the above-mentioned conditions are satisfied, the carriage return key "CAR. RET." (FIG. 2) is thereafter depressed in order to initiate a tabulation cycle of operation of the accounting machine, whereby the carriage is returned to the home position stop, the home position stop being the leftmost stop on the form bar as viewed from the front of the machine and also as diagrammatically illustrated in FIG. 88D, disposed directly above column #1. After the carriage has reached home position, start-preset push button SP1—SP2 (FIG. 1) is depressed. In response to the start-preset push button being depressed, the instruction register is preset to 0000009900, and, in addition, enter-keyboard-words lamp "EK" (FIGS. 1 and 2) is illuminated to provide a visual indication to the operator that the computer is now properly conditioned to sequentially receive into memory each of the seventy-four instruction and data words shown in the just-preceding chart. Thereafter, the just-mentioned words, starting with 0400000070 and ending with 0500869201, are sequentially indexed into the keyboard, and resume-program-bar *rpb* is depressed after each word is indexed into the keyboard. Consequently, as a result of each of the seventy-four instruction and data words having been sequentially indexed into the keyboard, with each being followed by a resume-program-bar depression, the program is thereby stored in memory addresses $\phi\phi$ through 73, the first instruction word in the program (0400000070) being stored in address-$\phi\phi$, and the last instruction word in the program (0500869201) being stored in address-73.

After the payroll program is properly stored in memory addresses $\phi\phi$ through 73, push button Sϕ1—Sϕ2 (FIG. 1) is depressed and thereby causes the word-selecting register to be preset to "00." Immediately thereafter, the instruction word in address-$\phi\phi$ is read out and stored in the instruction register. In response to a "04" thus being stored in section 1 of the instruction register, a shift instruction (SHF) is thereafter executed whereby the word stored in address-$\phi\phi$ is simply read out and copied back in address-$\phi\phi$ and the contents of section 5 of the instruction register—i.e., "70"—is thereafter stored in the word-selecting register in the same manner as previously described. Immediately thereafter, the instruction word stored in address-7ϕ is read out and stored in the instruction register. As the contents of section 1 of the instruction register is now "00," enter-keyboard-word lamp "EK" is again illuminated. After the enter-keyboard-word lamp is illuminated, the preceding payroll check number, say 50230, is indexed into the keyboard and the resume-program-bar subsequently depressed. As a result, the word 0000050230 is stored in address-93, and enter-keyboard-word lamp "EK" is again illuminated, indicating that another word is to be entered into the keyboard.

As illustrated in the top leftmost portion of the exemplary ledger card shown in FIG. 88A, if the fixed deductions with respect to the employee for this particular pay period are union dues, community chest, insurance, and annuity, the program number is indicated by the numeral "1"; if the fixed deductions for this particular pay period are insurance and annuity only, the program number is indicated by the numeral "2"; however, if the fixed deductions for this particular pay period are bond, credit union, insurance, and annuity, the program number is indicated by the numeral "3."

Therefore, if it is assumed that the fixed deductions relating to this particular employee for this particular pay period are bond, credit unit, insurance, and annuity, a "3" is indexed into the "dollar" amount row of the keyboard. Upon subsequent depression of the resume-program-bar, the word 0000000300 is stored in address-94, and, immediately thereafter, the instruction word stored in address-71 is read out and stored in the instruction register. In response to the word now stored in the instruction register, a "SHF" instruction is thereafter executed whereby the contents of address-94 is shifted two digital positions to the right, and the result—i.e., 0000000003—is stored in address-95. Immediately thereafter, the word stored in address-72 is read out and stored in the instruction register. In response to the word now stored in the instruction register, an "ADD" instruction is thereafter executed, whereby the word stored in address-95—i.e., 0000000003—is added to the word stored in address-31—i.e., 0897818131—and the result—i.e., 0897818134—is stored back in address-31, which, in effect, causes the numeral "3" to be added to the low-order digit of the fifth section of the instruction word stored in address-31; thereafter the instruction word stored in address-73 is read out and stored in the instruction register. In response to the word now stored in the instruction register, a "CMA" instruction is thereafter executed, whereby memory addresses 86 through 92 are cleared and the instruction word stored in address-ϕ1 is thereafter read out and stored in the instruction register.

Due to the fact that the two-decimal-digit number now stored in section 1 of the instruction register is "02," enter-card-words lamp "EC" (FIGS. 1 and 2) is illuminated, thus indicating that the computer is, at this time, properly conditioned and ready to receive the information magnetically recorded on the employee's ledger card, which, in this instance, comprises the employee's clock number, rate of pay, number of dependents, union dues deduction, community chest deduction, bond deduction, credit union deduction, insurance deduction, annuity deduction, earnings to date, Federal tax to date, city tax to date, annuity to date, F.I.C.A. to date, earnings to date at the end of the previous quarter, and proof total.

After the operator inserts the employee's ledger card into the card-handling portion of the computer in the manner previously described, the above-mentioned card words are magnetically picked up one after the other and stored in memory addresses 7ϕ through 85, as illustrated in the just-preceding chart. Thereafter, the instruction word stored in address-ϕ2 is read out and stored in the instruction register. In response to the word now stored in the instruction register, a "SUM" instruction is thereafter executed, whereby the words magnetically picked up from the ledger card, with the exception of the "proof total" amount, are added together, and the result is thereafter stored in address-95. Upon completion of the summation operation, the sum thus derived is compared with the proof total stored in address-85. If the contents of addresses 85 and 95 are of equal magnitude, the word stored in address-ϕ4 is subsequently read out and stored in the instruction register. However, if the sum and the proof total are not in agreement, indicating that the information of the magnetic ledger card was not accurately read into the computer, the word stored in address-ϕ1 is subsequently read out and stored in the instruction register. Thereafter, enter-card-words lamp "EC" is again illuminated, indicating that a reading error has occurred and that the ledger card just read should be removed manually and again be inserted into the card-handling portion of the computer, as before.

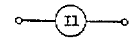
If it is assumed that the summation of the card words is in agreement with the proof total, so that the word stored in address-ϕ4 is read out and stored in the instruction register, an "EKW" instruction is executed, whereby the employee's clock number, indexed in amount rows #3 through #7 of the keyboard, is stored in address-95. Thereafter, the clock number stored in address-95 is shifted two places to the right and stored in address-85. Upon completion of the shifting operation, the clock numbers stored in addresses 7ϕ and 85 are compared for equality. If the clock numbers are equal, indicating that the time card and ledger card each relates to the same employee, the computer looks to address-ϕ7 for the next instruction; otherwise, the computer executes the instruction dictated by the word stored in address-ϕ6. In carrying out the dictates of the word stored in address-ϕ6, the carriage is returned to home position, and enter-card-word lamp "EC" is again illuminated, indicating that either the wrong time card or the wrong ledger card was utilized by the operator. However, in carrying out the dictates of the word stored in address ϕ7, a "POW" instruction is executed, whereby the program number stored in address-94 is printed out in column #7, as illustrated in FIG. 88D, the low-order cent and dime digits thereof being suppressed and thus not printed.

With the aid of the just-preceding portion of the mode of operation of the computer in executing a typical payroll program being described in detail to this point, a detailed comprehension of the remaining portion of the program can readily be had by following the step-by-step description thereof shown in the preceding chart, in exactly the same manner as just described. Consequently, a further detailed description thereof would be superfluous and, accordingly, is not deemed necessary in order to insure a full and complete comprehension and appreciation thereof. It is, however, to be appreciated that, while a specific mode of operation and application of the present computer has been described in detail with respect to a typical payroll operation, it is not intended to be implied or inferred that its adaptability and applicability are in any way limited to any specific accounting operation. Being of a general-purpose variety, the present computer is readily adaptable to be utilized in a highly effective manner in substantially any of the multitude of different present-day accounting operations.

91. *Component Chart*

Shown below in the following chart is an itemized listing of the values and other identifications of the various components and operating potentials utilized by each of the computer building blocks schematically illustrated in FIGS. 46 through 51 and previously described in detail. In each instance, the logical symbols shown in the chart below correspond to like building block symbols; the reference characters located to the left of the chart correspond to like reference characters used throughout the logical drawings with respect to the various building blocks; and the reference characters located at the top of the columns correspond to like reference characters used throughout the schematic representations of the various building blocks shown in FIGS. 46 through 51.

	650
(4759) (4762)	2N109
(4732) (4761)	2N173



	650	652	663
(4241–4242)	2N113	Output	CRA
(3268–3287)	2N113	Output	CRX
(3288–3307)	2N113	Output	CRY

I2

	650	653	654
(4664–4673)	2N109	1N128	–25

I3

	650	659
(4710–4715)	2N109	41

I4

	650	655	656
(4001) (4008) (4011) (4017) (4020) (4023) (4025) (4027) (4029) (4031) (4033) (4040) (4081) (4084) (4086) (4088) (4090) (4095) (4108) (4213) (4262) (4317–4320) (4325–4334) (4345–4348) (4353–4362) (4465)	2N140	5.6K	–6
(4004) (4253–4254) (4257) (4444) (4461–4463) (4476) (4485–4487) (4491) (4507–4510) (4532–4536) (4538–4544) (4546–4553) (4555–4556) (4557–4565) (4567–4571) (4674–4675)	2N140	20K	–6
(4005) (4168) (4255) (4266) (4454) (4468–4470) (4481) (4493) (4515) (4572–4591) (4593–4608) (4676–4677) (4850–4852)	2N140	1K	–6

I5

	650	655	656
(4070) (4093) (4101) (4105) (4111)	2N109	5.1K	–6
(4072) (4079)	2N109	3.9K	–6
(4074) (4225)	2N109	1.8K	–6
(4077) (4135) (4537) (4545) (4566)	2N109	2.0K	–6
(4098)	2N109	1.6K	–6
(4103)	2N109	3.0K	–6
(4233–4236) (4258–4261) (4442–4443) (4445–4449) (4451) (4474–4475) (4477–4478) (4498) (4524–4527) (4652) (4776–4777) (4780)	2N140	10K	–12
(4264)	2N140	2K	–12
(4265)	2N140	3.9K	–12
(4279) (4517)	2N140	4.7K	–6
(4414)	2N140	1.5K	–6
(4280)	2N140	620	–12
(4432) (4681–4684) (4751) (4753)	2N109	1.5K	–12
(4217)	2N140	15K	–12

I5

	650	655	656
(4409–4413)	2N140	8.2K	–6
(4472)	2N140	3K	–12
(4215) (4221–4222) (4245) (4876)	2N140	1K	–12
(4068)	2N140	5.6K	–12
(4256)	2N140	820	–6
(4036) (4554)	2N140	4.7K	–12
(4038) (4163) (4165)	2N140	10K	–6
(4492) (4494)	2N140	510	–6

I5

I6

	650	660
(4730–4731)	2N277	47
(4395–4396)	2N140	6.2K

	650	655	656	663
(4634)	2N109	200	–12	6

I7

	650	655	656	660
(4285)	2N140	270	–12	2.4K
(4703–4708)	2N109	820	–6	15K
(4702)	2N109	120K	6	22K
(4724–4726) (4752)	2N109	820	–6	27K
(4754) (4757)	2N109	1.5K	–6	22K

I8

	650	660	661	662
(4393–4394)	2N140	7.5K	3.6K	6
(4788–4793)	2N109	5.1K	30K	6

I9

	650	655	656	661	662
(4685–4637)	2N109	1.5K	–12	56K	6

I10

	650	655	656	664
(4737–4740) (4804) (4805)	2N109	1.8K	–6	39

I11

	650	655	656	658
(4237–4240) (4373–4382)	2N113	540	–18	300
(3248–3267) (4228–4231)	2N113	430	–18	240
(3228–3247)	2N113	340	–18	240

I12

	650	653	654	655	656
(4071) (4094) (4102) (4106) (4226)	2N140	1N128	–6	430	–12
(4104)	2N140	1N128	–6	240	–12
(4075)	2N140	1N128	–6	150	–12
(4099)	2N140	1N128	–6	130	–12
(4073) (4080) (4267–4270)	2N140	1N128	–6	330	–12
(4078)	2N140	1N128	–6	180	–12
(4865)	2N140	1N128	–6	2.2K	–12
(4069) (4091) (4530–4531)					
(4592)	2N140	1N128	–6	510	–12
(4112)	2N140	1N128	–6	390	–12
(4164) (4166) (4246–4249) (4452–4453) (4455–4460) (4479–4480) (4482–4483) (4528–4529) (4653) (4778–4779) (4781) (4860–4863) (4868) (4871) (4874)	2N140	1N128	–6	1.0K	–12
(4272–4273)	2N140	1N128	–6	200	–12
(4296)	2N113	1N128	–6	220	–12
(4313)	2N140	1N128	–6	5.1K	–12
(4415–4420)	2N140	1N128	–6	1.2K	–12
(4421)	2N140	1N128	–6	270	–12
(4211–4212)	2N140	1N128	–6	5.6K	–12
(4516)	2N140	1N128	–6	3.9K	–12
(4742–4744) (4806) (4307)	2N109	1N128	–6	820	–12

I13

	650	653	654	657
(4695–4701) (4760) (4763)	2N277	1N128	–18	47

I14

	650	655	656	660	661	662
(4003) (4010) (4013) (4019) (4022) (4035)	2N140	820	−6	10K	47K	6
(4014)	2N140	2K	−6	4.3K	51K	6
(4785–4787)	2N109	1K	−6	5.1K	30K	6
(4214)	2N140	1K	−12	5.6K	62K	6
(4335–4344)	2N140	2K	−6	5.1K	62K	6
(4363–4372)	2N140	3.9K	−12	10K	62K	6
(4405)	2N140	2K	−12	47K	200K	6

I15

	650	653	654	655	656	660
(4718) (4828)	2N109	1N128	−6	820	−12	15K
(4440)	2N140	1N128	−6	2.7K	−12	10K
(4439)	2N140	1N128	−6	1.2K	−12	10K
(4438)	2N140	1N128	−6	3.3K	−12	10K
(4048) (4049) (4051)	2N140	1N128	−6	750	−12	7.5K
(4050) (4056)	2N140	1N128	−6	1.0K	−12	9.1K
(4052) (4053)	2N140	1N128	−6	820	−12	9.1K
(4054)	2N140	1N128	−6	2.2K	−12	22K
(4055)	2N140	1N128	−6	1.5K	−12	13K
(4057)	2N140	1N128	−6	620	−12	5.6K
(4089) (4109) (4719) (4782) (4816) (4830)	2N109	1N128	−6	820	−12	5.1K

I16

	650	653	654	655	656	660
(4082) (4085)	2N109	1N128	−6	1.2K	−12	4.3K
(4087) (4096)	2N109	1N128	−6	2.0K	−12	10K
(4097) (4110)	2N109	1N128	−6	1.0K	−12	15K
(4083) (4107)	2N109	1N128	−6	2.9K	−12	33K
(4784)	2N109	1N128	−6	1.0K	−12	7.5K
(4076)	2N109	1N128	−6	3.0K	−12	39K
(4100)	2N109	1N128	−6	620	−12	9.1K
(4092)	2N109	1N128	−6	2.0K	−12	39K
(4113)	2N109	1N128	−6	5.1K	−12	5.1K
(4309)	2N140	1N128	−6	910	−12	6.2K
(4311)	2N140	1N128	−6	680	−12	6.2K
(4435)	2N140	1N128	−6	1K	−12	10K
(4436)	2N140	1N128	−6	1K	−12	6.8K
(4437)	2N140	1N128	−6	1.5K	−12	10K
(4717)	2N109	1N128	−6	820	−12	10K

I16

	650	655	656	663	667	668
(4169)	2N140	2.4K	−6	6	1000	8.2K
(4315)	2N140	820	−6	6	500	39K

I17

	650	655	656	660	667
(4271)	2N140	3.3K	−12	22K	560
(4286–4289)	2N140	3.9K	−12	39K	33
(4620)	2N140	430	−12	2K	1,000
(4621) (4632) (4641) (4650) (4678–4679)	2N140	1K	−12	2K	1,000
(4633) (4635–4638) (4642)	2N140	510	−12	2K	1,000
(4651)	2N140	300	−12	2K	1,000

I18

	649	650	655	656	660	661	662
(4797–4799)	470	2N109	3.9K	−12	5.1K	3.5K	6

I19

	650	653	654	655	656	660	661	662
(4002) (4009) (4012) (4018) (4021) (4024) (4026) (4028) (4030) (4032) (4034) (4041)	2N140	1N128	−6	1.2K	−12	6.8K	62K	6
(4015)	2N140	1N128	−6	470	−12	1K	20K	6
(4016)	2N140	1N128	−6	470	−12	2K	20K	6
(4039)	2N140	1N128	−6	1K	−12	2.2K	47K	GND
(4115–4130) (4187–4194)	2N109	1N128	−6	2.7K	−12	10K	47K	6
(4141) (4143) (4471)	2N140	1N128	−6	2.0K	−12	19K	100K	6
(4136)	2N140	1N128	−6	1.0K	−12	10K	100K	6
(4308) (4310)	2N140	1N128	−6	3K	−12	15K	82K	6
(4720) (4808) (4829)	2N109	1N128	−6	820	−12	5.1K	47K	6
(4721)	2N109	1N128	−6	510	−12	5.1K	15K	6
(4749–4750) (4783) (4819–4820) (4824–4825)	2N109	1N128	−6	820	−12	15K	47K	6
(4745) (4747)	2N109	1N128	−6	820	−12	5.1K	15K	6
(4741)	2N109	1N128	−6	820	−12	10K	39K	6
(4748) (4801) (4826)	2N109	1N128	−6	1K	−12	15K	47K	6

I20

	650	655	656	660	661	662	667
(4232)	2N140	20K	−12	22K	100K	6	180
(4219–4220)	2N140	20K	−12	39K	75K	6	180
(4314)	2N140	3.9K	−6	27K	39K	6	500
(4321–4324) (4349–4352)	2N140	1K	−6	13K	62K	6	2,000
(4495)	2N140	1K	−12	5.1K	100K	6	680
(4496)	2N140	5.6K	−12	10K	100K	6	56
(4853–4855)	2N140	3.9K	−18	10K	47K	6	100
(4497) (4519–4520)	2N140	5.6K	−12	10K	100K	6	47
(4521)	2N140	3K	−12	10K	100K	6	47

I21

	650	653	654	655	656	663	667	668
(4522)	2N140	1N128	−6	1K	−12	6	.01MF	33K

I22

	650	653	654	655	656	660	667
(4006–4007)	2N109	1N128	−6	1K	−12	5.1K	250
(4046)	2N140	1N128	−6	1K	−12	9.1K	250
(4290–4293)	2N140	1N128	−6	620	−12	3.9K	630
(4441)	2N140	1N128	−6	1.5K	−12	8.2K	150
(4207)	2N113	1N128	−6	240	−12	1.8K	470
(4434)	2N140	1N128	−6	2.2K	−12	15K	2,000
(4312)	2N140	1N128	−6	1K	−12	24K	2,000
(4422)	2N140	1N128	−6	560	−12	2.2K	2,000
(4401–4404)	2N140	1N128	−6	750	−12	5.6K	330
(4423)	2N140	1N128	−6	560	−12	3.3K	500
(4433)	2N140	1N128	−6	470	−12	4.7K	250

I23

	650	653	654	655	656	660	667
(4450) (4611–4613) (4615–4616) (4618)	2N140	1N128	−6	510	−12	2K	1,000
(4484) (4614–4645) (4647)	2N140	1N128	−6	390	−12	2K	1,000
(4505–4506) (4610) (4614) (4622) (4630–4631) (4639) (4643)	2N140	1N128	−6	1K	−12	2K	1,000
(4609) (4627)	2N140	1N128	−6	200	−12	2K	1,000
(4617)	2N140	1N128	−6	500	−12	2K	1,000
(4623–4624) (4640)	2N140	1N128	−6	300	−12	2K	1,000
(4619) (4625–4626) (4648–4649)	2N140	1N128	−6	390	−12	2K	500
(4628)	2N140	1N128	−6	510	−12	2K	500
(4629) (4646)	2N140	1N128	−6	200	−12	2K	500
(4680)	2N109	1N128	−6	2K	−12	15K	500
(4764) (4767–4768) (4771–4772) (4774–4775)	2N140	1N128	−6	470	−12	2K	1,000
(4765) (4770)	2N140	1N128	−6	1K	−12	27K	1,000
(4766) (4769)	2N140	1N128	−6	200	−12	2K	2,000

I23

	650	653	654	655	656	660	661	662	671
(4746)	2N109	1N128	−6	820	−12	5.1K	1K	6	1N128

I24

	650	653	654	655	656	660	661	662	667
(4037)	2N140	1N128	−6	470	−12	20K	47K	6	500
(4216)	2N140	1N128	−6	1K	−12	5.1K	47K	6	220
(4877)	2N140	1N128	−6	1K	−12	6.8K	33K	6	470
(4406)	2N140	1N128	−6	2K	−12	5.1K	100K	6	100
(4869) (4872) (4866)	2N140	1N128	−6	1K	−12	10K	47K	6	220
(4473)	2N140	1N128	−6	5.6K	−12	20K	47K	6	56
(4878)	2N140	1N128	−6	1K	−12	7.5K	5.1K	GND	470
(4818) (4827)	2N109	1N128	−6	820	−12	15K	47K	6	1,000

I25

	650	655	656	661	662	665	667	668	669
(4503)	2N109	30K	−12	180	6	.022 MF	.05 MF	75	27K

I26

I27

	650	655	656	667	668	670	672
(4809) (4810) (4821)	2N109	1K	−18	50	680	27K	.10 MF

	650	655	656	658	664	665	666	667	668	670	672	680
(4399–4400)	2N140	150	−18	10K	1K	15 MF	15 MF	.05 MF	1.5K	18K	.05 MF	.05 MF

I28

	650	655	656	678
(4864)	2N140	47K	−6	1K

I29

	650	656	664
(4223)	2N140	6	1.3K
(4218)	2N140	6	2.0K
(4263)	2N140	6	10K
(4722)	2N109	−6	10K

E1

	650	656	663	664
(4042–4045)	2N140	−6	6	27K
(4424–4431)	2N140	−12	6	39K
(4047) (4114) (4167)	2N109	−6	6	5.1K
(4716) (4723)	2N109	−6	6	2.7K
(4131) (4132) (4138) (4139) (4464) (4489–4490) (4511)	2N140	−6	6	2.0K
(4133) (4137)	2N140	−6	6	1.3K
(4466–4467) (4499–4500)	2N140	−6	6	68K
(4140–4145) (4162) (4186) (4274)	2N140	−6	6	1.0K
(4146–4161) (4170–4185) (4224) (4275–4278) (4300–4307)	2N140	−6	6	5.6K
(4281–4284)	2N109	−6	6	22K
(4227) (4294) (4488)	2N113	−6	6	1.5K
(4502)	2N140	GND	6	1.0K
(4512)	2N140	−6	6	3K
(4513)	2N140	−6	6	15K
(4800) (4880)	2N140	−6	6	0
(4514)	2N140	−6	6	690

E2

	650	656	664
(4243–4244) (4250–4253) (4383–4392)	2N113	−12	.5

E3

	650	656	663	664	673
(4504)	2N109	−6	6	2.7K	1.5K
(4654–4662)	2N140	−6	6	10K	360

E4

	650	656	661	662	664
(4709)	2N109	−6	1.8K	−12	62

E5

	650	655	656	663	664
(4727–4728)	2N109	100	−6	6	10K

E6

	650	656	663	664	666	673
(4295)	2N113	−6	6	4.7K	2000	1.5K

E7

	650	655	656	664	666
(4794–4796)	2N109	150	−12	270	.25 MF

E8

	650	655	656	663	664	673
(4729) (4756)	2N109	30	−6	6	2.7K	47

E9

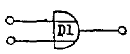
	650	655	656	660	661	662	663	664
(4688–4694) (4755) (4758)	2N109	33	−6	1.5K	5.1K	6	6	1K

E10

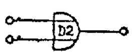
	650	655	656	661	662	664	667	668
(4822)	2N109	470	−18	30K	−18	47	50MF	10K
(4813) (4814)	2N109	470	−18	30K	−18	470	2MF	10K

EI

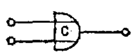
	679	689	690
(3000) (3003–3004) (3006–3013) (3016) (3112)	1N128	−6	120
(3353)	1N128	−6	8.2K
(993) (3002) (3014–3015) (3033–3037) (3039) (3043) (3078) (3095) (3121) (3123) (3134–3137) (3139–3140) (3147–3150) (3153–3154) (3157–3158) (3324–3326) (3343) (3346–3348) (3350–3351) (3357–3359) (3368–3370) (3372–3373) (3375–3378) (3387–3388) (3403) (3407–3429) (3432–3443) (3446–3452) (3486) (3507–3508) (3510–3512) (3517) (3520) (3553)	1N128	−6	47K
(3015) (3128) (3163–3164) (3327–3334)	1N128	−12	5.1K
(3017) (3113–3116) (3165)	1N128	−12	20K
(3044) (3117–3120) (3122) (3125–3127) (3129) (3145–3146) (3497–3506) (3522)	1N128	−12	10K
(3138)	1N128	−12	47K
(3133)	1N128	−12	2K
(3151) (3404–3406) (3415)	1N128	−6	27K
(3152)	1N128	−6	12K
(3005) (3132) (1758)	1N128	−6	10K
(3159–3162)	1N128	−12	4.3K
(3172–3181)	1N128	−12	1.5K
(3182–3183)	1N128	−12	4.7K
(3184–3187) (3206–3209)	S-4896	−6	82K
(3196–3205) (3218–3227)	S-4896	−6	120K
(3316–3319)	1N128	−12	18K
(3320)	1N128	−12	82K
(3321) (3513–3514) (3516) (3518–3519) (3521) (3523)	1N128	−12	27K
(3344–3345)	1N128	−6	6.2K
(3352) (3374)	1N128	−6	4.7K
(3061) (3354–3356) (3391) (3401–3402)	1N128	−6	62K



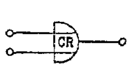
	679	689	690	691
(3028) (3038) (3042)	1N128	−6	47K	33K
(3041)	1N128	−6	47K	30K
(3030) (3040)	1N128	−6	47K	24K
(3029) (3031–3032)	1N128	−6	47K	15K
(3322)	1N128	−6	51K	33K
(3323)	1N128	−6	47K	20K



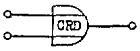
	683
(3001) (3087) (3090–3091) (3094) (3389)	68
(3018) (3020) (3022) (3024) (3026) (3053–3060) (3104–3111)	250
(3070–3077) (3123–3124) (3130–3131) (3168) (3550–3552)	180
(3141) (3188–3191) (3210–3213) (3312–3315) (3335–3342) (3360–3367) (3379–3386)	150
(3143–3144) (3392) (3394) (3396) (3398)	220
(3554)	470
(3487–3496)	120
(3430)	.015MF
(3444–3445)	500



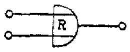
	683	687
(3062–3068) (3079–3085)	180	5.6K
(3069) (3086)	{ 180 68 }	5.6K



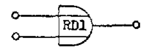
	679	683	687
(3088–3089) (3092–3093)	1N128	68	10K
(3142)	1N128	250	10K
(3155–3156) (3167) (3169–3171)	1N128	80	7.5K
(3349)	1N128	120	10K
(3393) (3395) (3397) (3399)	1N128	220	10K



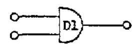
	691
(3019) (3021) (3023) (3025) (3027) (3045–3052) (3096–3103)	5.1K
(3476–3485)	10K



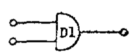
	679	691
(3192–3195) (3214–3217) (3371–3390) (3453–3454) (3456–3466) (3468) (3470–3473)	S-489G	10K
(3455) (3467) (3469) (3474–3475)	1N128	7.5K



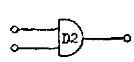
	679	689	690
(905–906)	1N128	6	51K
(960–965) (1115–1118) (1137) (1150) (1159–1160) (1452) (1743) (1750) (1754–1757) (1759) (1763–1765) (1772–1777) (1798–1802) (1804) (1808–1811)	1N128	6	10K
(1014) (1016) (1018) (1020) (1022) (1024) (1026) (1028) (1039) (1042) (1046) (1048) (1050) (1052) (1054) (1056)	1N128	6	68K
(966–970)	1N128	6	39K
(1013) (1015) (1017) (1019) (1021) (1023) (1025) (1027) (1029) (1040) (1043) (1047) (1049) (1051) (1053) (1055) (1057)	1N128	6	75K
(972–976) (982) (984) (986–988) (994–1004) (1006–1012) (1044) (1058–1065) (1119–1120) (1373–1375) (1382) (1392) (1397) (1401) (1403) (1414) (1424) (1429–1445) (1470–1473) (1534) (1538) (1542) (1546) (1549) (1558) (1561) (1564) (1578) (1598) (1605) (1669) (1673) (1689–1690) (1701–1702)	1N128	6	47K
(1088–1093) (1346–1361)	1N128	6	22K
(1094–1095) (1139–1142) (1169–1176)	1N128	6	56K
(1097–1098)	1N128	6	4.7K
(1131–1134) (1398) (1400) (1402) (1404)	1N128	6	5.1K
(1151–1158)	1N128	6	910
(1177–1196)	1N128	6	620



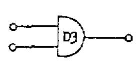
	679	689	690
(1399) (1447–1450)	1N128	6	100K
(1222–1225) (1257–1260)	S-489G	6	47K
(1323–1324)	1N128	6	220K
(1161–1162)	1N128	6	7.5K
(1321–1322)	1N128	6	470K
(1041)	1N128	6	120K
(1565) (1566) (1602) (1658) (1741–1742) (1782–1793)	1N128	6	15K
(1724–1733)	1N128	6	91K
(1744–1749)	1N128	6	3K
(1751–1753)	1N128	6	18K
(1778–1781)	1N128	6	20K
(1807)	1N128	6	27K



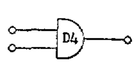
	679	689	690	691
(980)	1N128	6	100K	22K
(1045)	1N128	6	100K	10K
(1066–1067)	1N128	6	10K	5.1K
(1113–1114)	1N128	6	10K	2.0K
(1121)	1N128	6	20K	2.0K
(1198)	1N128	6	10K	3K
(1330)	1N128	6	47K	22K
(1345)	1N128	6	27K	7.5K



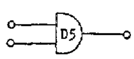
	679	689	690	691	695
(1096)	1N128	6	10K	5.1K	1000



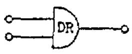
	679	689	700	701	702
(1197)	1N128	−6	2N140	1K	6



	679	689	690	692	693
(1325–1328)	1N128	6	39K	18K	−12

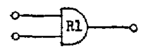


	679	696
(1687–1688)	1N128	10K

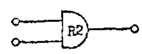


	689	690	696
(900–904) (907–914) (920–937) (940–950) (956–959) (1202–1205) (1207–1210) (1212–1215) (1217–1220) (1236–1239) (1241–1245) (1247–1250) (1252–1255)	6	10K	4.7K
(1135–1136)	6	20K	5.1K
(1005) (1363–1364) (1383–1391) (1393–1396) (1417–1423) (1425–1428) (1446) (1451) (1453–1469)	6	15K	5.6K
(938) (939) (951–955) (977–979) (981) (983) (985) (989–990) (1030–1038) (1103–1107) (1122–1130) (1143–1149) (1199) (1376–1382) (1415–1416) (1615–1617) (1723) (1760–1762) (1795–1797) (1405–1408) (1410–1413) (1814–1818)	6	10K	5.1K
(1084–1087) (1365–1372) (1474–1482) (1484–1533) (1535–1537) (1539–1541) (1543–1545) (1547–1548) (1550–1556) (1559–1560) (1562–1563) (1567–1577) (1579–1597) (1599–1601) (1603–1604) (1606–1608) (1610–1614) (1618–1657) (1659–1668) (1670–1672) (1674–1686) (1692–1700) (1703–1718) (1734–1740) (1720–1722) 1812–1813)	6	10K	5.6K

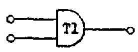
	689	690	696
(1099–1102) (1108–1111) (1281–1320)	6	18K	4.3K
(1226–1235) (1261–1270) (1331–1344) (1362)	6	9.1K	4.7K
(1271–1280)	6	11K	4.3K
(915–919)	6	120K	4.7K
(1112)	6	6.8K	5.1K
(1068–1083)	6	18K	4.7K



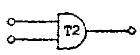
	689	690	691	696
(991–992)	6	27K	100	5.1K



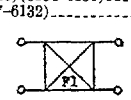
	689	690	700	703	704	705	706	707	708	709	710	711	712	713	714
(1803) (1805–1806)	−12	750	2N109	50	27K	.015 MF	50	750	10K	2N109	2K	5.1K	470	3.5K	6



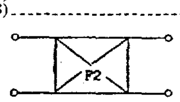
	664	666	689	690	700	703	709	710	712	713	714
(1163–1164)	1K	M'F	−6	5.1K	2N109	.05 MF	2N109	150K	5.1K	10K	6



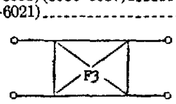
	722	723	724	725	726	727	728	729	734	735	736	737
(6061) (6066) (6072) (6077–6085) (6088–6089) (6091) (6093) (6097) (6100–6101) (6104)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	7.5K	500	7.5K	500
(6086–6087) (6090) (6092) (6094–6096) (6098–6099) (6103) (6105–6107) (6118–6119) (6134–6139)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	5.1K	500	5.1K	500
(6127–6132)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	5.1K	1000	5.1K	1000



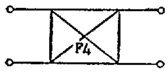
	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737
(6038) (6041) (6046)	2N113	2N113	−12	1K	−12	1K	5.1K	5.1K	−6	1N128	−6	1N128	7.5K	120	7.5K	120
(6047–6054) (6057–6060) (6108–6117) (6062–6065) (6067–6070) (6073–6076)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	−6	S489G	−6	S489G	7.5K	500	7.5K	500
(6055–6056)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	−6	1N128	−6	1N128	10K	500	10K	500
(6071)	2N109	2N109	−12	1K	−12	1K	5.1K	5.1K	−6	1N128	−6	1N128	8.2K	470	8.2K	470
(6102)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	−6	1N128	−6	1N128	5.1K	500	5.1K	500
(6133)	2N140	2N140	−12	1K	−12	1K	5.1K	5.1K	−6	1N128	−6	1N128	5.1K	1,000	5.1K	1,000



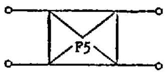
	722	723	724	725	726	727	729	734	735	736	737
(6001–6013) (6030–6037)	2N109	2N109	−6	1K	−6	1K	5.1K	15K	1,000	15K	1,000
(6014–6021)	2N140	2N140	−6	1K	−6	1K	5.6K	15K	500	15K	500



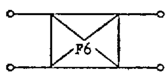
	722	723	724	725	726	727	729	730	731	732	733	734	735	736	737
(6039–6040)	2N140	2N140	−12	1K	−12	1K	5.1K	−6	1N128	−6	1N128	5.1K	180	5.1K	180



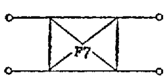
	722	723	724	725	726	727	728	734	735	736	737
(6022–6025)	2N140	2N140	−6	1K	−6	1K	5.6K	15K	500	15K	500



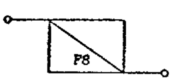
	722	723	724	725	726	727	728	730	731	732	733	734	735	736	737
(6026–6029)	2N140	2N140	−12	1K	−12	1K	5.6K	−6	1N128	−6	1N128	15K	500	15K	500



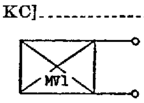
	722	723	724	725	726	727	728	729	734	736
(6122)	2N109	2N109	−6	820	−6	820	5.1K	5.1K	15K	15K



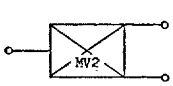
	722	723	724	725	726	727	734	735	737	738	739	752
(6120) [.25 Sec.]	2N109	2N109	−6	1K	−6	1K	27K	500	10MF	−6	27K	470



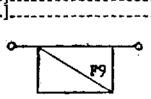
	722	723	724	725	726	727	735	737	741	742	743	744	745	746	748	749	750
(6125) [50 KC]	2N140	2N140	−6	10K	−6	10K	1,120	1,120	−6	15K	6	10K	−12	10K	1N128	−6	15K



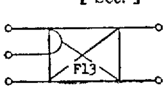
	722	723	724	725	726	727	735	737	741	742	749	750	752	753	754
(6126) [400 cps]	2N140	2N140	−6	1K	−6	1K	.145 MF	.145 MF	−6	18K	−6	18K	.01 MF	.01 MF	2K



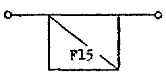
	722	723	724	725	726	727	729	735	736	749	750	755
(6124) [.04 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	20MF	15K	−6	15K	5.1K
(6142) [.06 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	2MF	15K	−6	27K	2.4K
(6141) [.06 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	4MF	15K	−6	18K	2.4K



	722	723	724	725	726	727	735	736	741	742	749	750	755	757	758	762	763
(6145) [.00124 Sec.]	2N109	2N109	−6	820	−6	820	.1 MF	15K	6	2.7K	−6	22K	3.9K	27K	2N109	560	150

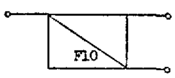


	722	723	724	725	726	727	734	735	736	737
(6000)	2N109	2N109	−12	1K	−12	1K	10K	500	10K	500



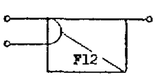
	722	723	724	725	726	727	734	737	741	742	752
(6147–6148) [3μ Sec.]	2N109	2N109	−6	680	−6	1K	750	330	−6	15K	68
(6149–6150) [3μ Sec.]	2N109	2N109	−6	1K	−6	1K	0	33	−6	15K	220

	722	723	724	725	726	727	729	735	736	749	750	755	756
(6121) [.25 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	25 MF	15K	−6	15K	3.9K	.5 MF
(6143) [.075 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	5 MF	15K	−6	15K	2.4K	.5 MF
(6140) [.01–.04 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	3 MF	15K	−6	15K	100K	.5 MF
(6123) (.12 Sec.]	2N109	2N109	−6	820	−6	820	5.1K	11 MF	15K	−6	18K	2.4K	.5 MF



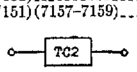
	722	723	724	725	726	727	734	735	736	737	749	750	757	758	760	761	759
(6144)	2N109	2N109	−6	820	−6	820	10K	.1 MF	10K	.1 MF	6	3.2K	15K	2N109	150	5.1K	5.1K

	722	723	724	725	726	727	734	735	736	737	749	750	757	758	759	760	761
(6146)	2N109	2N109	−6	820	−6	820	15K	.1 MF	15K	.1 MF	6	3.2K	5.1K	2N109	5.1	150	5.1K

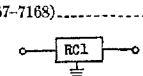


	766	767	768	780
(7007)	120	1N128	91K	6v.
(7008–7010) (7015) (7026–7089) (7118–7129) (7132–7143)	500	1N128	10K	GND
(713–7131)	.01MF	1N128	56K	GND
(7156)	.04MF	1N128	15K	GND
(7154–7155)	.10MF	1N128	1.5K	GND

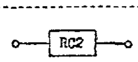
	766	767	768	769	770	771
(7101) (7153)	1MF	1N128	1M	6	2.7K	1K
(7112–7114) (7117)	1MF	1N128	15K	6	2.7K	1K
(7150)	1MF	1N128	15K	6	1K	2.7K
(7152)	.1MF	1N128	15K	6	2.7K	680
(7151) (7157–7159)	.1MF	1N128	15K	6	2.7K	710



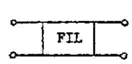
	772	773
(7160–7165) (7167–7168)	1K	.05MF



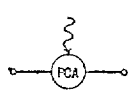
	766	772
(7166) (7169)	470	200K



	785	786	787	788	789	790	791
(7144–7149)	220	220	.1 MF	.1 MF	4.7K	4.7K	−12



	160	794	795	796	797	799	800
(4501)	Clairex CL3	100	1N128	6	10K	.10 MF	7K



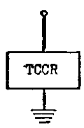
	802	803	804	805	806	807	808	809	810
(4733–4736) (4802–4803)	2N140	−6	6	2K	Clairex CL3	50 MF	5.1K	5.1K	−12

	650	655	656	661	662	681	684
(4856–4859) (4867) (4870) (4873) (4875)	2N140	2.2K	−12	100K	6	1N77A	−12

	813	814	815	816	817	818	819	820	821	822	823	824	825
(4811-4812) (4815) (4817) (4823)	2N109	−6	6.8K	2.7K	150	.1MF	7.5	1N128	560	2.7K	2.7K	6	5MF

	827	829	830	831	832	833	834	835	836	837	838	839
(7000-7002)	2N109	1.5K 70° Thermister	1.5K 70° Thermister	270	−12	652 ZENER DIODES	620	2.4K	1.2K	652 ZENER DIODES	1.1K	−12

840



	828	844	845	846	847	848	849	850	851	860
841	2N140	2N158	.5	.5	1N128	1N128	25	11	50	100MF

	852	853	854	855	856	857	858	859	866	861	862	863	864	865
842	2N173	2N109	−6	680	−12	2.5	−8	620	25MF	500	150	680	−18	−12

92. *Signal Origination and Destination Chart*

Shown below in the following chart is an alphabetical listing of the various signal lines shown throughout the logical diagrams, wherein the number in the center column corresponds to the figure number of the origination of that particular line; and wherein the number, or numbers, in the right-hand column corresponds to the logical diagram figure or figures to which the particular line is connected as an input line.

Signal	Origination	Destination
(A1-A2)	83	83
AAD	60	52B
ABG	60	52B
AD	61	53, 63, 70, 71
ADD	56	56, 73, 74, 75, 77, 79
AF	72	72
AI	77	76, 77
AJ	71	60, 71
AK	69	60, 70
AM	74	58, 60, 68, 73, 74, 75, 79
AN	61	58, 68, 71, 72, 74, 75, 76, 77, 78, 79, 80, 81, 84
(AN*a*-AN*d*)	54	54
APN	56	58, 68, 73, 74, 75, 77, 79
ARO	74	53, 69, 71, 74, 78
ASA	67	60, 67, 70, 71
AUT	84	76, 84
AYW	60	60
B	83	83, 85
B*a*	62	54, 55, 58, 60, 67, 68, 69, 70, 71, 76
BB	85	85
(B*b*-B*c*)	62	54, 55, 58, 60, 67, 68, 70, 71
B*d*	62	54, 55, 58, 60, 61, 67, 68, 69, 70, 71, 74, 77
(B*a*L-B*d*L)	55	55, 57, 58, 59
(B*a*M-B*d*M)	62	60, 62
BBD	60	52B
(BC*a*-BC*b*)	62	61, 62
BJ	71	60, 71
BK	69	60, 70
BM	74	60, 68, 73, 74, 79
BYW	60	60
C12	84	61
C13	84	84
C14	84	60, 62
C23	84	61, 64, 71, 76, 81
C24	84	61, 65, 84
C31	84	60, 61, 68, 69, 70, 71, 75, 76, 77, 79, 84
C34	84	55, 61, 71
C41	84	55, 58, 60, 61, 62, 69, 74, 75, 77, 79, 81, 84
C42	84	67, 84
C43	84	67
C	83	81, 83, 85
CA	74	60, 68, 71, 74, 77, 78
CB	74	60, 74, 78
CCR	85	75
CFE	56	73, 75, 77, 78, 79
CFF	61	60, 61, 64, 69, 79
CFM	56	69, 71, 73, 75, 76, 77, 78, 79
CLC	82	82
CM	74	60, 69, 73, 74, 77, 79
CMA	56	73, 74, 79
CO	82	82

Signal	Origination	Destination
CPA	74	53, 69, 74
CPM	74	53, 60, 69, 73, 74, 79
CRA	60	60
CRX	66	66
CRY	66	66
CSA	85	71
CYC	61	62
D	82	81, 83, 82
Dϕ	63	59, 67, 69, 71, 75, 76
DϕM	63	66
D1	63	59, 66, 67, 80
(D2-D5)	63	58, 66, 67, 80
(D6-D7)	63	57, 66, 67, 80
D8	63	55, 66, 67, 80
D9	63	53, 55, 58, 61, 67, 69, 70, 74, 77, 79, 80
D9L	63	61, 64, 70
D9M	63	66
D*a*	63	63, 68, 72
(D*b*-D*d*)	63	63, 72
DAD	75	60, 69, 71, 74, 75, 78
DBD	62	62, 63
DDF	61	54, 61, 79
DE2	58	58
DIV	56	56, 71, 73, 74, 75, 77, 78
DR	68	68
E	83	82, 83, 85
EAS	54	53, 54, 75, 77
ECW	56	58, 60, 61, 68, 69, 71, 73, 74, 75, 76, 77, 79, 81, 82, 83
EG	61	53, 68, 72, 76, 77
EKL	81	81
EKW	56	58, 60, 73, 74, 75, 78, 79, 81
EOC	82	64, 65, 66, 82
EOF	77	75, 77
EOW	71	60, 79
EPT	56	58, 60, 69, 73, 75, 76, 77, 79
EVG	80	80
Fϕ	72	71, 81
F8	72	75, 77, 78, 81
F9	72	68, 74, 76, 79, 81
F1ϕ	72	62, 68, 70, 71, 72, 74, 75, 78
(F*a*-F*d*)	72	72
FG	75	58, 61, 64, 65, 75, 76, 77
FO1	69	60, 69
FO2	69	60, 69, 71
FOO	69	53, 61, 69, 71, 79
G	82	81, 82, 83, 85
GC	75	61, 69, 75, 76, 77, 79
GO	82	71, 72, 81, 82
GOO	62	61, 62, 79
(Hϕ-H1)	57	81
H2	57	70, 78, 81
H3	57	62, 70, 74, 75, 78, 81
H4	57	62, 70, 78, 81
H5	57	62, 78, 81
H6	57	77, 79
HC	82	82
HH	83	83, 85
i	82	82
(Iϕ*a*-Iϕ*d*)	59	65
((I1*a*-I1*d*)	59	64
(I2*a*-I2*d*)	58	58, 59, 65
I2ϕ	59	58
(I3*a*-I3*c*)	58	58, 59, 64
I3*d*	58	59, 64
I34	59	58, 69, 74, 75, 76, 77, 79
I49	59	58
(I4*a*-I4*d*)	58	58, 59, 65
(I5*a*-I5*c*)	58	58, 59, 64

Signal	Origination	Destination
I5d	58	59, 64
(I6a–I6b)	57	57, 65, 68
I6c	57	57, 65, 68, 82
I6d	57	57, 65, 68, 76, 82
(I7a–I7d)	57	57, 64
(I8a–I8d)	55	56
I9a	55	56
ICC	84	75
ICR	56	56
IFD	72	68
IFJ	72	80
IN4	58	58
IW	61	61
J9	71	74, 76, 77
JA	71	60, 71
Ja	71	53, 54, 71, 72
(Jb–Jc)	71	53, 71, 72
Jd	71	54, 71, 72
JB	71	60, 71
JHR	75	58, 61, 68, 69, 71, 74, 75, 76, 77
JJF	71	71
JL	71	71
JM	60	60, 73
JS	71	60
Kφ	70	69, 75, 76
KA	69	60, 69
Ka	70	53, 54, 70
(Kb–Kc)	70	53, 70
Kd	70	54, 70
KB	69	60, 69
KEY	75	61, 72, 75, 76, 77, 79, 81
KK	82	82
KKF	69	70
KL	70	70
KS	70	60, 77
L	85	82, 83, 85
Lφ	57	69, 75, 76, 77, 81, 85
L1	57	76, 77, 81, 85
L2	57	81, 85
L3	57	81
LFA	85	68
LIN	72	72
LK	75	79
LL	85	85
M14	85	68
MA	75	60, 68, 71, 73, 74, 75, 76, 77, 78
MAN	84	76, 84
MB	56	75, 79, 81
MDD	56	57, 68, 73
MI	76	55, 56, 58, 60, 61, 64, 69, 72, 75, 76, 77, 79
MIC	76	61, 69, 76, 77
MIN	72	68, 74, 75, 76, 77, 78
MJ	71	60, 68, 71, 73, 74, 76, 77, 78, 79, 80, 81
MK	69	60, 70, 73
MOD	56	68, 74, 75, 76, 77, 79
MOO	60	61, 64
MOU	76	58, 61, 69, 75, 76, 79, 82
MSA	67	55, 58, 60, 67, 68, 70, 71
MUS	56	53, 56, 62, 68, 70, 71, 72, 73, 74, 75, 77, 78
MV	84	84
MXW	60	60
MYW	60	60
N	83	83
(N1–N2)	85	85
NN	85	85
NT	72	72, 81
OBM	76	69, 70, 73, 74, 75, 76, 77, 81
OBN	77	71, 73, 74, 76, 77, 78, 79, 85
P	83	82, 83, 85
pc	53	53
PC	61	56, 61, 64, 72, 75, 76
PCT	72	72, 75, 80
PFφ	72	72
PCI	53	53, 54, 73
PIφ	55	55, 57, 58, 59
PJφ	71	71, 80
PJ1	71	71
PJ5	71	71
PKφ	70	70
PK9	70	70
POW	56	60, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81
PRB	58	58
PT	72	72
PTF	76	Input to Tape Reader Terminal (FWD)
PTR	76	Input to Tape Reader Terminal (REV.)
PTS	76	Input to Tape Reader Terminal (STOP)
PRφ	68	68
PR2	68	68
PR8	68	68
PWφ	64	64, 65
Q	85	82
q	53	54, 82
qc	53	53
r	53	54
Rφ	68	68, 73, 74, 75, 76, 77, 78, 79, 85
R1	68	74, 75, 78
R8	68	73, 74, 75, 77, 78, 79
(RAφ–RA9)	80	80
(Ra–Rd)	68	68
rc	53	54
RAD	77	58, 60, 68, 69, 73, 74, 77, 78
RCC	85	85
RDS	85	85
REC	82	82, 83, 85
RED	82	81, 82, 85
REV	74	74
REW	77	69, 76, 77
RF	81	81
RLR	68	68
RLM	68	68
ROC	56	58, 61, 62, 69, 73, 75, 76, 77, 79, 82, 85
ROS	77	60, 69, 75, 76, 77, 79, 81, 85
RP	84	77, 78, 84
RPB	78	78, 79, 81, 82
RS	76	55, 61, 62, 63, 69, 71, 72, 74, 75, 76, 77, 78, 79, 81
(RSa–RSd)	67	67
RSB	78	53, 60, 69, 73, 75, 77, 78, 79
RSS	67	60
s	54	54
S4L	58	58
SA	78	62, 68, 69, 70, 71, 78, 79
SAM	79	69, 75, 77, 79, 85
SAS	53	53, 54
SB	79	62, 68, 69, 70, 71, 72, 74, 75, 78, 79
SCD	85	76, 85
SES	79	58, 61, 69, 75, 76, 77, 79, 85
SHA	79	54, 60, 61, 69, 70, 71, 79
SHB	79	69, 71, 79
SHF	56	62, 68, 70, 71, 73, 74, 75, 76, 78, 79
SMC	60	60
SOO	60	61, 64
SP	84	58, 75, 76, 81, 84
SQ	81	76, 81
STφ	84	76, 79, 81, 84
ST1	84	65, 76, 79, 81, 84
STD	79	60, 61, 68, 71, 73, 76, 78, 79, 81, 82
STP	56	79, 81
SUB	56	53
SUM	56	58, 73, 74, 77, 79
t	54	54, 85
(Ta–Td)	60	60
TCL	75	75
TDA	69	69, 75
TDS	60	60
TH	69	58, 60, 61, 64, 69, 73, 76, 79
(TI2–TI5)	61	64, 65
(TS2–TS5)	73	61
(Wφφ–Wφ9)	65	65
U	83	82, 83, 85
(Wφa–Wφd)	65	65
(W1a–W1d)	64	64
(W1φ–W19)	64	66
WDF	61	55, 61
WOA	81	72, 74, 75, 76, 77, 78, 79, 80, 81
WR	62	60, 66
(XDa–XDd)	60	52a
XDW	60	60
(XGφ–XG9)	66	55A, 52B
YDW	60	65
(YDφ–YD9)	65	52A, 52B
(YGφ–YG5)	66	52A
(YG-6–YG9)	66	52B
Z	54	60
ZA	60	60
ZB	60	60
ZM	60	60
ZNO	54	77

While particular embodiments of the invention have been shown and described in detail, it will be obvious to those skilled in the art that changes and modifications may be made without departing from the invention in its broader aspects, and, therefore, the aim in the appended claims is to cover all such changes and modifications as fall within the true spirit and scope of the invention.

What is claimed is:

1. An electronic digital computing machine comprising: registering means differentially positionable to different registering positions; means for differentially positioning said registering means; a storage device of a type providing a plurality of addressable storage locations, each storage location being capable of having a word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any desired storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and for reproducing a word manifestation previously stored at the particular storage location dictated by said address selecting means, means for providing information indicative of the position at which said registering means was differentially positioned; means operatively connected to said informa-

tion providing means and to said read-write means for causing a word manifestation indicative of said registering position to be stored in said storage device at the selected storage location dictated by said address selecting means; means for receiving ledger card record media each having information both magnetically recorded and printed thereon; signal reproducing means for sensing the information magnetically recorded on each ledger card record medium; means operatively connected to said signal reproducing means and to said read-write means for causing word manifestations indicative of said magnetically recorded information to be stored in said storage device at the particular storage locations selected by said address selecting means; means settable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced word manifestations as indicated by said pattern of control; and means associated with said signal reproducing means for thereafter magnetically recording on a selected one or ones of said ledger card record media a selected portion or portions of the results of said computational operation while effectively erasing the information previously recorded magnetically thereon.

2. An electronic digital computing machine in accordance with claim 1 which further includes means for suppressing all non-significant zeros of said computational results during said recording operation.

3. An electronic digital computing machine comprising: registering means differentially positionable to different registering positions; setting means for differentially positioning said registering means; a storage device of a type providing a plurality of addressable storage locations, each storage location being capable of having a word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any desired storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and for reproducing a word manifestation previously stored at the particular storage location dictated by said address selecting means; means for providing information indicative of the position at which said registering means was differentially positioned; means operatively connected to said information providing means and to said read-write means for causing a word manifestation indicative of said registering position to be stored in said storage device at the selected storage location dictated by said address selecting means; means for receiving ledger card record media each having information both magnetically recorded and printed thereon; signal reproducing means for sensing the information magnetically recorded on each ledger card record medium; means operatively connected to said signal reproducing and to said read-write means for causing word manifestations indicative of said magnetically recorded information to be stored in said storage device at the particular storage locations selected by said address selecting means; means settable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with said read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced word manifestations as indicated by said pattern of control; means associated with said signal reproducing means for thereafter magnetically recording on a selected one or ones of said ledger card record media a selected portion or portions of the results of said computational operation while effectively erasing the information previously recorded magnetically thereon; and means operatively connected to said setting means for causing said registering means to be differentially positioned indicative of a selected portion or portions of said computational results.

4. An electronic digital computing machine comprising: a plurality of differentially positionable printing means capable of printing on an associated medium a plurality of word manifestations comprising a plurality of characters of various kinds disposed in different order positions along a recording line transversely oriented with respect to said medium, the unique and independent differential position of each of said printing means determining the specific kind of character to be printed by that particular printing means; an actuator associated with each of said printing means for differentially positioning the corresponding printing means, each of said actuators being independently movable in at least one direction; means for moving all of said actuators in said direction; stop means associated with each actuator for selectively immobilizing said actuator independently of the other actuators at any one of a plurality of differential positions, the differential position at which the indivdual actuators are immobilized determining the differential positioning of their associated printing means; a storage device of a type printing a plurality of addressable storage locations, each storage location being capable of having a word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any desired storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and for reproducing a word manifestation previously stored at the particular storage location dictated by said address selecting means; means for providing information indicative of the positions at which said printing means were differentially positioned, means operatively connected to said information providing means and to said read-write means for causing a word manifestation indicative of the differentially-set positions of said printing means to be stored in said storage device at the selected storage location dictated by said address selecting means; means for receiving ledger card record media each having information both magnetically recorded and printed thereon; signal reproducing means for sensing the information magnetically recorded on each ledger card record medium; means operatively connected to said signal reproducing and to said read-write means for causing word manifestations indicative of said magnetically recorded information to be stored in said storage device at the particular storage locations selected by said address selecting means; means settable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with said read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced word manifestations as indicated by said pattern of control; means associated with said moving means and said stop means for causing said printing means to be differentially positioned indicative of a selected portion or portions of the results of said computational operation; and means associated with said signal reproducing means for magnetically recording on a selected one or ones of said ledger card record media a selected portion or portions of said computational results while effectively erasing the information previously recorded magnetically thereon.

5. An electronic digital computing machine comprising: a storage device of a type providing a plurality of addressable storage locations with each storage location

being capable of having a word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any chosen storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and reproducing a word manifestation previously stored at the particular storage location dictated by said address selecting means; key-set recording means capable of selectively printing a plurality of characters of a plurality of selected kinds on an associated medium during a cycle of operation thereof, said recording means including means for driving said recording means through cycles of operation, a plural order series of differential actuators each being movable between a normal position and any of a series of differential positions each indicative of a character of a different kind, means driven by said drive means to advance said differential actuators to differential positions and again return said actuators to normal position in a cycle of operation of said recording means, a first differential stop means settable from the keyboard of said recording means for selectively immobilizing each of said differential actuators in a differential position indicative of the particular kind of corresponding order character indexed in said keyboard, and a second electrically energizable differential stop means for selectively immobilizing each of said differential actuators in a differential position indicative of a particular kind of character; additional means for determining when the mobile actuators are at any of the plurality of differential positions; means for providing information indicative of the positions at which said actuators were differentially set; means operatively connected to said information providing means and said read-write means for causing a word manifestation indicative of the differentially-set positions of said actuators to be stored in said storage device at the selected storaged location dictated by said address selecting means; means for receiving a ledger card record medium having information both magnetically recorded and printed thereon, said means being capable of positioning said medium in operative relationship with respect to said recording means such that a subsequent printing operation on said medium is effected by said recording means along a predetermined printing line of said medium; signal reproducing means for sensing the information magnetically recorded on said ledger card record medium; means operatively connected to said signal reproducing means and to said read-write means for causing word manifestations indicative of said magnetically recorded information to be stored in said storage device at the particular storage locations selected by said address selecting means; means setable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with said read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced word manifestations as indicated by said pattern of control; means for initiating a cycle of operation of said recording means; means operatively connected to said second differential stop means and said additional means for causing said differential actuators to be immobilized at differential positions collectively indicative of a selected portion or portions of the results of said computational operation such that said selected portion or portions of said computational results is subsequently printed on said record medium along said predetermined printing line; and means associated with said signal reproducing means for magnetically recording on said ledger card record medium a selected portion or portions of said computational results while effectively erasing the information previously recorded magnetically thereon.

6. A computing machine in accordance with claim 5, which further includes means for recording on said ledger card record medium a magnetic indication indicative of the location of the next printing line of said medium along which a subsequent printing operation is to be effected.

7. A computing machine in accordance with claim 5, which further includes means for recording on said ledger card record medium a magnetic indication whose position corresponds to the vertical location of the next printing line of said medium along which a subsequent printing operation is to be effected.

8. An electronic digital computing machine comprising: a storage device of a type providing a plurality of addressable storage locations with each storage location being capable of having word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any chosen storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and reproducing a word manifestation previously stored at the particular storage location dictated by said address selecting means; key-set recording means capable of selectively printing a plurality of characters of a plurality of selected kinds on an associated medium during a cycle of operation thereof, said recording means including means for driving said recording means through cycles of operation, a plural order series of differential actuators each being movable between a normal position and any of a series of differential positions each indicative of a character of a different kind, means driven by said drive means to advance said differential actuators to differential positions and again return said actuators to normal position in a cycle of operation of said recording means, a first differential stop means settable from the keyboard of said recording means for selectively immobilizing each of said differential actuators in a differential position indicative of the particular kind of corresponding order character indexed in said keyboard, and a second electrically energizable differential stop means for selectively immobilizing each of said differential actuators in a differential position indicative of a particular kind of character; additional means for determining when the mobile actuators are at any of the plurality of differential positions; means for providing information indicative of the positions at which said actuators were differentially set; means operatively connected to said information providing means and said read-write means for causing a word manifestation indicative of the differentially-set positions of said actuators to be stored in said storage device at the selected storage location dictated by said address selecting means; means for receiving a ledger card record medium having data and synchronizing information magnetically recorded thereon, said means being capable of positioning said medium in operative relationship with respect to said recording means so that a subsequent printing operation on said medium is effected by said recording means along a predetermined printing line of said medium; signal reproducing means for sensing the information magnetically recorded on said ledger card record medium; means operatively connected to said signal reproducing means and to said read-write means for causing word manifestations indicative of said magnetically recorded data information to be stored in said storage device at the particular storage locations selected by said address selecting means and at a speed determined by said magnetically recorded synchronizing information; means settable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with said read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced

word manifestations as indicated by said pattern of control, all at a speed substantially higher than the operational speed of said signal reproducing means; synchronizing means operatively connected to said signal reproducing and said control means for synchronizing the operational speeds thereof; means for initiating a cycle of operation of said recording means; means operatively connected to said second differential stop means and to said additional means for causing said differential actuators to be immobilized at differential positions collectively indicative of a selected portion or portions of the results of said computational operation such that said selected portion or portions of said computational results is subsequently printed on said record medium along said predetermined printing line; and means associated with said signal reproducing means for magnetically recording on said ledger card record medium a selected portion or portions of said computational results while effectively erasing the data information previously recorded magnetically thereon, all at substantially the same rate of speed as the operational speed of said signal reproducing means.

9. An electronic digital computing machine comprising: a storage device of a type providing a plurality of addressable storage locations with each storage location being capable of having a word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any chosen storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and reproducing a word manifestation previously stored at the particular storage location dictated by said address selecting means; key-set recording means capable of selectively printing a plurality of characters of a plurality of selected kinds on an associated medium during a cycle of operation thereof, said recording means including means for driving said recording means through cycles of operation, a plural order series of differential actuators each being movable between a normal position and any of a series of differential positions each indicative of a character of a different kind, means driven by said drive means to advance said differential actuators to differential positions and again return said actuators to normal position in a cycle of operation of said recording means, a first differential stop means settable from the keyboard of said recording means for selectively immobilizing said differential actuators in a differential position indicative of a particular kind of character; additional means for determining when the mobile actuators are at any of the plurality of differential positions; means for providing information indicative of the positions at which said actuators were differentially set; means operatively connected to said information providing means and said read-write means for causing a word manifestation indicative of the differentially-set positions of said actuators to be stored in said storage device at the selected storage location dictated by said address selecting means; means for receiving a ledger card record medium having data, synchronizing and linefinding information magnetically recorded thereon; signal-reproducing means for individually sensing the data, synchronizing and linefinding information recorded on said ledger card record medium; means associated with said receiving means for causing said ledger card record medium to be positioned in operative relationship with respect to said recording means and at a position dicated by the sensed linefinding information, such that a subsequent printing operation on said medium is effected by said recording means along the particular printing line of said medium indicated by said linefinding information; means operatively connected to said signal-reproducing means and to said read-write means for causing word manifestations indicative of said magnetically recorded data information to be stored in said storage device at the particular storage locations selected by said address selecting means and at a speed determined by the sensed synchronizing information; means settable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with said read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced word manifestations as indicated by said pattern of control, all at a speed substantially higher than the operational speed of said signal reproducing means; synchronizing means operatively connected to said signal reproducing and said control means for synchronizing the operational speeds thereof; means for initiating a cycle of operation of said recording means; means operatively connected to said second differential stop means and to said additional means for causing said differential actuators to be immobilized at differential positions collectively indicative of a selected portion or portions of the results of said computational operation such that said selected portion or portions of said computational results is subsequently printed on said record medium along the particular printing line dictated by the sensed linefinding information; means associated with said signal reproducing means for magnetically recording on said ledger card record medium a magnetic indication whose position correspondings to the next printing line along which a subsequent printing operation is to be effected on said ledger card record medium, while effectively erasing the linefinding information previously recorded magnetically on said medium; and means associated with said signal reproducing means for magnetically recording on said record medium both synchronizing information and a selected portion or portions of said computational results synchronized therewith while effectively erasing the synchronizing and data information previously recorded magnetically thereon, all at substantially the same rate of speed as the operational speed of said signal reproducing means.

10. An electronic digital computing machine comprising: a storage device of a type providing a plurality of separate storage locations, each storage location being capable of having a word manifestation stored therein; address selecting means associated with said storage device for obtaining access to any chosen storage location thereof; read-write means associated with said storage device for selectively storing a word manifestation and reproducing a word manifestation previously stored at the particular storage location selected by said address selecting means; means for receiving ledger card record media each having data and synchronizing information recorded thereon; signal reproducing means for sensing said data and synchronizing information and operatively connected to said read-write means whereby word manifestations indicative of said data information are stored in said storage device at the selected storage locations selected by said address selecting means and at a speed determined by said synchronizing information; means settable to a predetermined pattern of control indicative of a desired computational operation to be executed; control means responsive to said pattern of control and operatively associated with said read-write means and with said address selecting means for effecting the reproduction of predetermined word manifestations from said storage device, and including means for executing the particular computational operation on said reproduced word manifestations as indicated by said pattern of control, all at a speed substantially higher than the operational speed of said signal reproducing means; synchronizing means operatively connected to said signal reproducing and said control means for synchronizing the operational speeds thereof; and means associated with said signal reproducing means for recording on a selected one or ones of said ledger card record media a selected portion or portions of the results of said computational operation together with said synchronizing information,

all at substantially the same rate of speed and the operational speed of said signal reproducing means.

References Cited in the file of this patent

UNITED STATES PATENTS

2,658,681	Palmer et al.	Nov. 10, 1953
2,882,817	Wolf	Apr. 21, 1959
2,902,215	Lauschke	Sept. 1, 1959
2,909,993	Shafer et al.	Oct. 27, 1959
2,928,080	Auerbach	Mar. 8, 1960
2,933,720	Newhouse et al.	Apr. 19, 1960
2,939,120	Estrems et al.	May 21, 1960
2,950,467	Hoffman	Aug. 23, 1960
2,972,444	Phelan	Feb. 21, 1961

UNITED STATES PATENT OFFICE

CERTIFICATE OF CORRECTION

Patent No. 3,112,394 November 26, 1963

Patrick B. Close et al.

It is hereby certified that error appears in the above numbered patent requiring correction and that the said Letters Patent should read as corrected below.

Column 254, line 72, for "means," read -- means; --; column 256, line 38, after "positioned" strike out the comma; column 257, line 36, for "storaged" read -- storage --; line 52, for "setable" read -- settable --; column 258, line 16, before "word" insert -- a --; column 259, line 45, after "recording" insert -- means for selectively immobilizing said differential actuators in predetermined positions indicative of the particular kind of corresponding order character indexed in said keyboard, and a second electrically energizable differential stop --; same line 45, after "immobilizing" insert -- each of --; line 65, for "dicated" read -- dictated --; column 260, line 26, for "correspondings" read -- corresponds --.

Signed and sealed this 12th day of May 1964.

(SEAL)
Attest:

ERNEST W. SWIDER
Attesting Officer

EDWARD J. BRENNER
Commissioner of Patents