PROJECT
WHIRLWIND

DIGITAL COMPUTER LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

REPORT R-221

WHIRLWIND I OPERATION LOGIC

Margaret F. Mann

Robert R. Rathbone

John B. Bennett

Digital Computer Laboratory

Massachusetts Institute of Technology
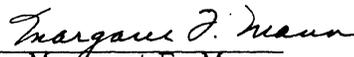
Cambridge 39, Massachusetts

1 May 1954

FOREWORD

This Report is intended to provide a general summary of Whirlwind I operation logic. The Report is somewhat more than the survey for the layman found in "Whirlwind I: A High-Speed Electronic Digital Computer;"[*] on the other hand, it brings up to date many of the descriptions started in "Whirlwind I Computer Block Diagrams."[**] The Appendix contains selected block diagrams, some simplified, of the computer.
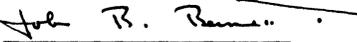
Sections 5.1-5.3 in Chapter 5, "Magnetic-Core Storage," have been adapted from "The MIT Magnetic-Core Memory," a paper given by William N. Papian at the Joint Eastern Computer Conference (Washington, D.C., December 1953).

Separate publications on Whirlwind I terminal equipment and electronic circuits will be issued at a later date. These topics are therefore not treated in detail here.

As far as possible, the computer terms used in this Report are those defined in "Standards on Electronic Computers: Definition of Terms, 1950" (50 IRE 8. S1); any exceptions are defined as they appear.

SIGNED _Margaret F. Mann_
Margaret F. Mann

Robert R. Rathbone
Robert R. Rathbone

John B. Bennett
John B. Bennett

APPROVED _Jay W. Forrester_
Jay W. Forrester

[*] Digital Computer Laboratory Report R-209, R.R. Rathbone, 15 August 1951

[**] Digital Computer Laboratory Report R-127-1, R.R. Everett and F.E. Swain, revised May 1952

ABSTRACT

Whirlwind I is an electronic digital computer in which numbers are represented by groups of short electrical impulses. High operating speed rather than simultaneous computations shortens the solution time of problems. Computations are carried out in the binary-number system, with a basic register length of sixteen binary digits.

Instructions are transferred from punched paper tape into an internal high-speed magnetic-core storage, supplemented by external magnetic-tape and magnetic-drum systems. A central control element regulates the operation of input and output gates between the bus system (providing parallel-digit transmission) and various units of the computer.

Instructions, initial data, and preliminary results are stored in arrays of magnetic cores, and every memory location is accessible in an equal length of time. Numbers and instructions may be stored interchangeably; the computer can therefore handle with equal ease problems with a small controlling program and a large amount of initial data and problems with a large controlling program and little initial data and partial results.

Punched-paper-tape and magnetic-tape devices, magnetic drum, typewriter, and oscilloscope are all used as computer terminal equipment.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

1. INTRODUCTION

1.1 The Whirlwind I Computer

Whirlwind I is a high-speed electronic computer in which numbers are represented by groups of short electrical impulses and computation is carried out in digital form. The basic principle of the machine is to perform one arithmetic operation at a time and to depend upon high operating speed rather than upon simultaneous computing operations to shorten the solution time of problems.

The binary-number system is used for Whirlwind computation. The basic register length is sixteen binary digits (equivalent to about 5 decimal digits). For general mathematical work, it is possible for the computer to handle numbers having a length which is a multiple of the standard 16-digit register.

Whirlwind I has an internal high-speed magnetic-core storage, sufficient in capacity for the solution of most problems. Initial data and instructions are transferred from relatively slow punched paper tape into this high-speed storage. External magnetic-tape and magnetic-drum systems supplement internal storage.

Whirlwind I uses parallel-digit transmission over a bus system which provides one channel for each binary digit of the 16-digit register. Each major unit of the computer can transmit digits to and receive digits from this bus. The operation of the various output and input gates between the bus and units of the computer is controlled in the proper sequence by a central control element.

The basic impulse rate (pulse-repetition frequency) for general operation of the computer is 1 mc; pulse duration is 0.1 μsec. Within the arithmetic element, the impulse rate for multiplication and shifting is 2 mc.

Instructions, initial data, and partial and intermediate results are stored in arrays of magnetic cores. Each digit of each storage register consists of a particular magnetic core; whether a '0' or a '1' is stored depends on the direction of magnetization of the core. Since readout of the cores is destructive, each storage access cycle includes a read and a rewrite or write, depending on whether the old information is to be left in the register or new information is to be inserted. Time is not a coordinate of the information; therefore, every memory location is accessible in an equal length of time.

Instructions and numbers may be stored interchangeably. Problems with a small controlling program and a large amount of initial data or problems with a large controlling program and small initial data and partial results may be handled with equal flexibility.

Terminal equipment consists of punched-paper-tape and magnetic-tape devices, magnetic drum, typewriter, and oscilloscope.

1.2 Summary of Whirlwind I Specifications

STORAGE (16-digit words)

Magnetic-Core Storage (MS):

2 banks, each having 1024 cores per digit plane; access time, 10 μsec.

Auxiliary Drum Storage:

12 groups each of 2048 registers on magnetic drum; single word or block transfer to and from MS; average access time to single work or block: 8.5 msec within a group, 16 msec to select new group; block transfer rate, 64 μsec/word.

Test Storage:

32 toggle-switch registers; 5 flip-flop registers (interchangeable with any 5 toggle-switch registers).

SPEED (in microseconds)

Addition:

To get one number from MS, add it to one already in AE 35

To get two numbers from MS, add them, and transfer answer to MS 100

Multiplication and Roundoff:

To get one number from MS, multiply it by one already in AE 50

To get two numbers from MS, multiply them, transfer product to MS 120

TERMINAL EQUIPMENT

Punched Paper Tape and Typewriters:

Flexowriter 7-hole tape (6 information, 1 index); 6-binary-digit code for letters and decimal numbers. Input to computer: mechanical tape reader (106 msec/line, 318 msec/word), photoelectric tape reader (7 msec/line, 21 msec/word); output: tape punch (93 msec/line, 279 msec/word), printers (about 135 msec/character, up to 900 msec for carriage return).

Magnetic Tape:

Parallel-serial storage of binary digits in 3 pairs of nonadjacent channels (2 information pairs, 1 index pair). Redundant recording in pairs minimizes tape-flaw errors. Maximum density, 100 lines (200 binary information digits) per inch; speed, 30 inches per sec. Coded tape recording of computations enables printer to operate independently of computer.

Oscilloscope Display:

Five modified Dumont 5-inch oscilloscopes and several 16-inch magnetically deflected CRT's are available for displays. X and Y axes each have 2048 discrete positions (about 350 μsec for point or vector set up and display, about 480 μsec for character set up and display). Fairchild camera, automatically controlled by computer, can be used with either type.

Buffer Drum:

Magnetic drum acts as temporary storage of input and output data arriving at computer in random and asynchronous manner from multiple sources and leaving for various output devices.

1.3 Arithmetic Considerations

1.31 The Binary-Number Base (see Section 3.1)

Whirlwind I uses the binary (or base 2) system of numbering, which facilitates the utilization of electronic storage elements requiring only two stable states*, such as conducting or cutoff for a tube and charged or discharged for a condenser; these states represent the two numbers, "0" and "1", of the binary system. Conversion from decimal to binary numbers is done in two steps: from decimal to coded binary by Flexowriter; from coded binary to binary by the computer (which must be instructed to do so).

1.32 Fixed Point

To meet the problem of requiring storage registers of finite range to hold numbers of theoretically infinite range, the fixed-point system is used in Whirlwind I. In this system, the range of numbers which can be handled is dependent on register length and is limited for any given problem; special provisions can be made in the instructions for those cases in which the normal range is inadequate. In normal operation, a scale factor assigned by the programmer is associated with each number in the machine and remains constant during the course of a problem. For extra-precision results and at great expense to speed, the problem may be programmed to use extra-precision and/or floating-point arithmetic.

1.33 Signs, Negatives, and the 1's Complement

An arithmetic computer must be capable of handling the signs of numbers. For WWI use, positive binary numbers are identified by a 0 in the sign-digit (first) position of the number, while negative numbers have a 1 in the sign-digit position. The sign digit is followed by the 15 digits expressing magnitude; by design, the binary point is understood to be placed immediately after the sign digit. In effect, this design convention causes all numbers to have positive or negative values less than 1, a fact for which compensation can be made if necessary in setting up problems for computer solution.

An arithmetic computer must also be able to subtract. To accomplish this end, Whirlwind I uses the 1's-complement system, which makes it unnecessary to build into the equipment the ability to subtract. In the 1's-complement method, subtraction becomes the addition of a complement, the actual subtraction having taken place when the complement was formed; such a process is advantageous in that the subtraction required to form a complement is simple and easily mechanized. In the binary system, a 1's complement is obtained with extreme speed and simplicity by interchanging 0's and 1's at each digit of the original number. (The use of the 1's complement in subtraction is discussed in Sec. 3.12.)

---

* The FLIP-FLOP (FF) is the most common type. This unit uses two tubes; according to which of the tubes is conducting, the flip-flop is said to be storing a 0 or a 1. A 0 is stored by pulsing the 0 input, a 1 by pulsing the 1 input of the FF, regardless of the previous contents. The 0 or 1 held by the FF may be changed to 1 or 0 by pulsing the complement input.

1.4 Switching Considerations

In the parallel-type system of Whirlwind I, a complete number is transmitted over a bus of 16 coaxial cables. The routing of a number becomes, then, the momentary connection to the bus of two registers, one sending, one receiving. More than one such compound bus requires the addition of considerable equipment; therefore Whirlwind I utilizes a single main bus system of parallel conductors to which are connected the several component elements (Fig. 1). Switching is therefore kept to a minimum -- connection of each element to the bus or disconnection from it -- and is done sequentially. Only one number is transmitted at any one time over the bus.

Figure 1. General Block Diagram, WWI System

1.5 Comparison with Manual System

Every computing system has certain basic elements. In a manual system these elements might be a messenger who brings in a problem, an operator, the notebook in which he enters incoming data and records answers, a desk calculator, and a messenger who takes the results. In the Whirlwind I system, Figure 1, the operator becomes CENTRAL CONTROL; the desk calculator, the ARITHMETIC ELEMENT; the notebook, STORAGE; and the messengers, the INPUT and OUTPUT units.

Just as in manual computation the operator controls the steps to be performed, so in Whirlwind I computation, CENTRAL CONTROL takes each instruction in sequence from STORAGE, examines it and sends pulses to the various parts of the computer to perform the necessary processes.

The ARITHMETIC ELEMENT of Whirlwind I has registers which correspond to the keyboard, accumulator, and counter of the desk calculator; storage of an instruction or a number in Whirlwind I corresponds to writing a line of information in the notebook.

1.6 The Operation Code

1.61 Characteristics

Whirlwind I employs binary instructions filed in storage registers allocated by the programmer. (Such binary instructions and the binary numbers for computer use are identical in form and are known indiscriminately as "words." Control determines whether a given word is to be treated as an instruction or as a number.) Where necessary, an instruction may be sent to the arithmetic element for arithmetic modification; this permits carrying out the same sequence of similar operations on several quantities with only a single sequence of instructions continually modified (or indexed) by the addition or subtraction of suitable constants. This ability of the computer is very useful for generalized instruction sequences in which a standard set of instructions is to be used in many similar problems.

1.62 The Standard Instruction

Each instruction describes the storage register to or from which a number is to be transferred, the element which is the source or destination of the number, and any necessary further operations, particularly those in the arithmetic element. These standard instructions have two parts:

1. A 5-digit code number designating one of a group of 32 computer operations. This operation section of the instruction is decoded and carried out by control; and

2. An 11-digit code number designating (a) the address of the storage register to be connected to the bus (selected by the storage switch); (b) the number of digits by which a word is to be shifted; or (c) the terminal equipment to be used.

The 32 computer operations can be represented by 5 binary digits if the first is described as 00000. Eleven binary digits are likewise sufficient to describe the 2048 storage registers in Whirlwind I. (Ten digits describe 1024 registers; the eleventh digit selects either Bank A or Bank B of MS, each having 1024 registers.)

The 16 digit positions of all words (both numbers and instructions) are arbitrarily numbered from 0 to 15, reading from left to right. If a number is involved, the binary point is considered as being between position 0 and position 1 (see Section 1.32). If an instruction is involved, the operation section and the address section are interpreted as whole numbers.

1.63 Included Operations

Built into Whirlwind I is the ability to add, subtract (see Section 1.33, above), multiply, divide, and to shift a number in a register to the right or to the left. Whirlwind I also is provided with certain control operations which allow great savings in the number of instructions

required for a problem, allow the machine to exercise a certain amount of program generation by permitting changes in computing procedure according to computed results, and allow the computer to perform such processes as sorting and extraction of data from tables. These control operations are:

1. Subprogramming (or transfer of control). Control is instructed to stop the sequence of instructions it is executing and to change to another sequence stored in the machine.

2. Conditional programming. Here the change in the instruction sequence depends on some previously computed result (most easily, the sign of a number). Any decision the machine may be called on to make is reduced to the question of whether or not a number in the accumulator is positive or negative.

3. Digital transfers. This operation allows computed data to be inserted in actual instructions -- for example, the insertion of a computed address if an entry is to be removed from a table. The address of the desired entry is computed and inserted in the removal instruction.

1.64 WWI Operations

The following table describes Whirlwind control operations. These operations are represented by lower-case letters, abbreviations of terms descriptive of the process. Certain representative operations are treated in more detail in Section 3.2; for a complete description of all operations see Digital Computer Laboratory report M-1624-1, "Short Guide to Coding and Whirlwind I Operation Code," P.R. Bagley (28 November 1952).

Fill accumulator:

ca	clear and add
cs	clear and subtract
cm	clear and add magnitude

Arithmetic:

ad	add
su	subtract
dm	difference of magnitudes
sa	special add
mr	multiply and round off
dv	divide
slr	shift left and round off
slh	shift left and hold
srr	shift right and round off
srh	shift right and hold

Store:

ts	transfer to storage
td	transfer digits
ta	transfer address

Transfer control:

sp	transfer control (subprogram)
cp	conditional transfer control (conditional program)

Convenience:

ex	exchange
ao	add one
ab	add B register

Special:

sf	scale factor
ck	check
clc	cycle left and clear
clh	cycle left and hold
md	multiply digits
rs	reset

In-out:

si	select in-out unit or stop the computer
rd	read
rc	record
bi	block transfer in
bo	block transfer out

1.7 Block-Diagram Symbols

The following symbols are peculiar to Digital Computer Laboratory block diagrams.

TYPES OF DATA

STANDARD PULSE

LONG PULSE

SIGNAL LEVEL

ALL ARROWS INDICATE DIRECTION OF FLOW OF INFORMATION. ARROWHEADS APPEAR AT FORWARD (DESTINATION) END OF LINE.

DATA CHANNELS

SINGLE-CHANNEL LINE

MULTIPLE-CHANNEL CABLE. INDIVIDUAL LINES ARE IDENTICAL AND PARALLEL IN TIME.

MULTIPLE-CHANNEL CABLE. INDIVIDUAL LINES ARE SIMILAR IN PURPOSE BUT NOT IDENTICAL NOR PARALLEL IN TIME.

CONNECTION OF LINES

NO CONNECTION

CONNECTION

EXAMPLES OF BRANCH CONNECTIONS. EXTRA ARROWHEADS PREVENT BACK-FLOW OF DATA.

OMISSION OF ARROWHEADS ALLOWS DATA FLOW IN BOTH DIRECTIONS BETWEEN A AND B.

CRYSTAL CONNECTION IN MATRIX

DIRECT CONNECTION IN MATRIX

COMPONENTS

FLIP-FLOP (FF)

OUTPUT "0"   OUTPUT "1"

FF

CLEAR OR SET TO "0"   SET TO "1"

COMPLEMENT (CHANGE SETTING)

GATE TUBE (GT)

PULSE INPUT   GT   PULSE OUTPUT   GATING INPUT

PULSE INPUT   GT   PULSE OUTPUT   GATING INPUT

GATE GENERATOR (GG)

PULSE INPUT   GG   FIXED-GATE OUTPUT

DELAY ELEMENT (DE)

PULSE INPUT   DE 0.5µs   PULSE OUTPUT   DELAYS DATA BY FIXED AMOUNT

PULSE INPUT   DE   PULSE OUTPUT   SLIGHT DELAY, REQUIRED LOGICALLY

2. A GENERAL OUTLINE OF THE SYSTEM

2.1 System Operation

System operation of the Whirlwind computer can best be understood by describing the use of each of the several basic elements shown in Figure 1.

First, consider that every step of a computation must be called for by the control element. It is necessary, therefore, to have a supply of instructions within the machine. These instructions are kept in consecutively numbered registers of the storage element. Thus, each instruction is put at a specific address and can be obtained when needed by having storage read out the contents of this register or extract the information contained therein.

2.11 Normal Operation

Instructions are selected -- ordinarily in sequence -- according to the setting of the program counter (PC), a component of the central control element; the program counter is indexed one as the instruction is performed. It is possible to change the sequence when desired; the procedure will be considered in Sections 2.12 and 2.13.

Figure 2 shows how an instruction is selected. The program counter contains the address of the register which holds the instruction. This address is read out onto the bus through a set of gate tubes (GT)* and travels via the bus through another set of gate tubes to set the storage switch. This switch, attached to storage, is thus set to select the storage register holding the instruction desired.

Figure 2. Select Instruction

---

* In Whirlwind I, a two-grid tube which will conduct only if positive voltages are supplied coincidentally to both grids.

Figure 3 shows the next step, extracting the instruction from storage. The gate tubes attached to storage are opened and the instruction is read out to central control's parity register (PAR)*, which holds the instruction until the storage switch can be cleared of the storage address to which it is set.

Figure 3. Read Out Instruction

Figure 4 shows how an instruction is set up. While the PAR holds the instruction, the storage switch is cleared of the address of this instruction. The entire instruction is then read from PAR onto the bus; from the bus it is distributed between the control switch and the storage

Figure 4. Set Up Instruction

---

* If the instruction comes from test storage (TS), it goes directly via the bus to PAR; if it comes from magnetic-core storage (MS), it goes through PAR to the bus and then back to PAR. (See Section 2.2 for details of storage. The apparently useless trip from PAR to the bus and back to PAR simplifies the problem of commanding MS and TS interchangeably.) No parity check is performed if the instruction comes from test storage.

switch. The control switch decodes the 5 digits of the operation code and sets the operation control for the desired operation. The storage switch decodes the 11 digits of the address, selecting the storage register which, for arithmetic operations, holds the number to be operated upon.

A computation such as an addition requires three instructions, two to send the numbers to be added to the arithmetic element, and one to store the result. In general, the first instruction transmits one number to the arithmetic element (Figure 5). The storage gate tubes are opened and the desired number read out onto the bus (through PAR if the number was in MS). The input gate tubes of the arithmetic element are also opened, allowing the number to go into the arithmetic element. (Some additional processes such as clearing may be needed within the arithmetic element. Once fed the operation code, operation control will supply all the necessary commands.) The next instruction will put the second number into the arithmetic element and command whatever operation on the numbers is designated by the operation code.

Figure 5. Perform Operation

In summary, the sequence below is followed for each instruction:
1. The quantity one is added to the program counter.
2. The counter contents (the address of the next instruction) is read into the storage switch via the bus.
3. The instruction is read out from storage and held temporarily in the parity register.

4. Going from PAR, the instruction is divided between the storage switch (to locate the number to be operated upon) and the control switch (to set up operation control for the desired operation).

5. The number is read from storage to the arithmetic element, where the desired operation on it is carried out under the direction of operation control.

Figure 6 shows the storage of an arithmetic result. The output gate tubes of the arithmetic element are opened and the result transmitted to the bus and into MS via the PAR and MS input GT or into TS directly via the test-storage input gate tubes.

Figure 6. Store Result

To produce an external record of a result, an appropriate instruction calls upon in-out control to activate the desired output device for recording the result. Similarly, to insert data into the computer an instruction calls upon in-out control to activate an input device. A further discussion of in-out control will be found in Section 2.24.

2.12 Subprogram Operation

The subprogram (sp) operation is shown in Figure 7. As in normal operation, the sp instruction is extracted from storage and stored in PAR. From PAR the instruction is distributed between the control switch and the storage switch. The storage switch is set up to the address section of the instruction. However, the subprogram operation code of this instruction calls for no readout from the storage register selected by the switch, but for a transfer of the address section of the instruction from the PAR to the program counter. After completion of the sp instruction, the next instruction to be taken by the program counter is the one whose address in storage has just been transferred to it, that is, the one to be found in the register designated by the subprogram instruction.

Figure 7. Subprogram

2.13 Conditional-Program Operation

The conditional-program operation is also shown in Figure 7. Operation control, activated by the operation code, cp, sends a pulse to the arithmetic element. If the sign of the number in the accumulator is positive as evidenced by a 0 in the sign-digit position, the pulse is stopped. The contents of the program counter are not changed; the next instruction in sequence as indicated by the program counter is taken, and the computation proceeds. If the number is the accumulator is negative, on the other hand, the pulse is returned to the control switch, where it changes the switch setting from conditional program to subprogram. Since operation control is now set up for the subprogram operation, the address section of the instruction in

PAR is transfered to the program counter, and the instruction sequence is changed to start at this new address, as described above.

2.2 Characteristics of Basic Elements

2.21 Storage

WWI presently has 2048 magnetic-core storage (MS) registers of 16 binary digits each. Each core digit plane stores a 32 x 32 array of digits (total 1024); sixteen planes in parallel store 1024 16-digit words; two banks, each of sixteen planes, store 2048 words.

WWI also has so-called test storage, consisting of 32 toggle-switch (manually changeable) storage registers and 5 flip-flop (electronically changeable) storage registers. The 5 flip-flop registers may be substituted for any 5 toggle-switch registers. The test-storage registers govern the flow of initial instructions into magnetic-core storage. Since 2048 addresses is the capacity of the program counter, the 32 test-storage registers always deny the use of an equal number of magnetic-core storage registers.

The test-storage registers are connected directly to the main bus; since they operate at computer speed they are controlled directly by central control. Magnetic-core storage, however, is not connected directly to the bus and has its own separate MS control system. Information is always transmitted to and from MS through parity register, never directly via the bus.

Timing of MS internal operations is directed by MS control. When an MS operation is called for, central control transfers computer control to MS control until an MS cycle has been completed; computer direction is then returned to central control from MS control.

2.211 The Storage Switch

As used in this report, the block labeled "Storage Switch" refers either to the electronic switch used with test storage or to the diode-matrix decoders used with magnetic-core storage. For test storage, the electronic switch selects a storage register from the addresses 0 through 31; for magnetic-core storage, the decoders select a storage register from the addresses 32 through 2047.

One digit of the 11-digit address selects one of the 2 banks of 16 planes. Of the remaining 10 digits, 5 establish the x-coordinate in the 32 x 32 array of the digit plane; the remaining 5 establish the y-coordinate. (See Section 5.4 , below.)

2.212 Parity-Register Characteristics

The parity register is a 16-digit flip-flop storage register provided with sending and receiving connections to the bus and to magnetic-core storage. Its purpose is threefold: to serve as a temporary storage for instructions while the storage switch is being cleared; to serve as the temporary storage location for a word being transferred to and from magnetic-core storage; and to determine the parity count of the word being transferred to or from magnetic-core storage.

2.22 Central Control

Whirlwind I, being a digital computer, operates on a step-by-step basis rather than on a continuous one. This concept holds from the simplest part of an operation through to the computation of complete problems.

2.221 Master Clock

Each operation is made up of a series of processes, each process consisting, for example, of a single transfer, shift, or check. A number of these processes may be performed by different pieces of equipment at the same instant of time for a given command, but only one such process at a time by a particular piece.

A command is a pulse which, in essence, orders the computer to perform a single process. The computer will do so and then await another command (pulse) to perform the next process. The master clock provides pulses which, when gated by operation control, call for these processes at the highest rate possible compatible with the operating speed of the equipment. (A lower rate would not affect computer functioning or results but only the time required for a computation.)

The master clock, then, is the source of all pulses concerned with timing of computer and in-out operations; it is not a unit which emits pulses on only one line and at only one frequency as might be expected of a clock. Emanating from the master clock are pulses at the various frequencies used by the computer: 2 mc for multiplication and shifting, 1 mc for most of the rest of computer operations, single pulses for push-button operation, and certain occasional pulses required for control of minor cycles of elements like magnetic-core storage and input-output. Moreover, pulses from push-button sources or from input-output equipment are asynchronous; therefore, another main function of the master clock is synchronization of these timing pulses with those at one of the main frequencies.

Figure 52 shows the components of the master clock. The time pulse distributor is discussed in Section 2.223.

2.222 The Program Counter

The program counter has the following characteristics:

1. Capacity. For WWI, capacity is 2048 addresses. Eleven binary digits are needed to describe 2048 registers; an 11-section scale-of-two (binary) counter is used. (See Section 2.21, above.)

2. Counting Rate. The counting rate is relatively unimportant since the counter is added to only once for each operation completed. The time per operation, using magnetic storage, is of the order of 35 μsec; using test storage, of the order of 17 μsec. There is no lower limit to the counting rate.

3. Reading and Changing. The counter must be able to read onto the bus. Provision must also be made for clearing the counter and reading in a new address from the bus, as described in Sections 2.12 and 2.13.

2.223 Time-Pulse Distributor and Control Switch

Central control commands the opening and closing of gates and the transferring of numbers and information in the machine by supplying pulses to the equipment at the right places and at the right times. For each operation to be performed a different sequence of pulses must be available. The control switch selects the equipment to be operated during the course of an operation; the time-pulse distributor furnishes the proper timing to the equipment.

A basic device needed for computer control is an electronic switch (Figure 8). This switch, simplified, consists of a number of flip-flop circuits connected through a crystal-diode matrix. The switch shown contains only two flip-flops. Since each of these flip-flops has two stable states, the switch has four possible positions (outputs). By adding more flip-flops, more positions can be obtained. The number of positions = 2^n, where n = number of flip-flops. The function of the electronic switch is to turn on, i.e., raise the potential of, only one output line at a time; this line is determined by the settings of all the associated flip-flops. All the remaining lines are at some lower potential.

Figure 8. Electronic Switch

Figure 9 shows how such an electronic-matrix switch can be used to make a time-pulse distributor (TPD) which will produce consecutive time pulses on consecutive output lines. The distributor consists of an eight-position diode-matrix switch; the eight outputs are fed to two-grid gate tubes, while the flip-flops driving the switch are connected together in a counter circuit. Each gate tube in turn is opened so that time pulses will come out in consecutive order, starting with the top line and finishing with the bottom line. After the last pulse appears, the switch will reset itself, and the next pulse will appear at the top line.

The distributor, then, receives a continuous string of 1-mc supply pulses and divides them up so that they appear in consecutive order on the output lines as time pulses; a particular piece of equipment supplied with one of these time pulses can be operated only at the time that pulse occurs in the rotating cycle of time pulses. A complete operation is considered to be started when the first time pulse appears out of the time-pulse distributor.

Figure 9. Time-Pulse Distributor

2.224 Operation Control

Figure 10 shows how the time-pulse distributor can be used in conjunction with a 32-position switch (called the control switch) to command the different computer processes. The time-pulse distributor feeds time pulses in consecutive order and in rotation onto lines 1 through 8. The 32-position switch receives the operation-code section of the instruction from the bus and selects the operation line designated by the operation code. Activating this line in effect sets into "standby" condition all the equipment needed for carrying out the operation. (The selected operation line supplies a gating voltage to one grid of the gate tube associated with each of these pieces of equipment.) Because an operation is performed step by step, all this equipment must operate in a previously established sequence. This sequence is effected by sending a particular time pulse from the TPD to the second input of the gate tube(s) associated with the piece(s) of equipment called for at that moment in the sequence. Each time pulse, then, activates the proper equipment at the proper time. (In WWI the lines from the distributor and the switch form what is called the "Control Matrix." An abstract of this matrix is given in Figure 56.)

Figure 10. Operation Control and Switch

For example, assume that an operation requires equipment a, g, m, q, r, and z, and that it be performed in the sequence: delay, a and r simultaneously, q, m, delay, g and z simultaneously, delay, a. The connections shown in Figure 10 would produce the sequencing of equipment as follows:

TIME PULSE	EQUIPMENT
1	(delay)
2	a, r
3	q
4	m
5	(delay)
6	g, z
7	(delay)
8	a

If the control switch had not been changed when time pulse 1 came around again, the same equipment would be cycled on again in the same order, as time progressed from 1 to 8.

Whirlwind I now has 120 gate tubes (called "Control Pulse Output Units") which serve as "keys" to the operation of computer equipment.

2.23 Arithmetic Element

The arithmetic element or AE (shown schematically in Figure 64) is capable of the following operations:

1. Addition;
2. Subtraction;
3. Multiplication;
4. Division;
5. Shifting and cycling;
6. Transmitting to and receiving from the bus;
7. Detecting the signs of numbers in the accumulator and/or A-register;
8. Complementing the contents of the accumulator and/or A-register.

In operation, the arithmetic element obtains one number at a time from the bus and performs such operations on it as are requested. Where two numbers are required, as in addition, the first number is in the accumulator of the arithmetic element; the second is brought in by the next operation. The first number in the accumulator may be the result of a previous arithmetic operation or may have been brought in by the previous transfer operation.

The result of an arithmetic operation remains in the accumulator unless special steps are taken to remove it. It may be placed in storage by an additional instruction, but such a transfer does not destroy the contents of the accumulator. A number in the accumulator remains there until cleared by a new instruction which requires an empty accumulator for its execution.

AE consists of three major components:

1. The A-register or AR;

2. The accumulator or AC;

3. The B-register or BR.

2.231 A-Register

AR is a simple flip-flop register. Its functions are:

1. To receive a number coming into AE from storage. AR must be connected to the bus by a set of read-in gate tubes.

2. To transmit numbers to AC for addition or subtraction. Gate tubes must be provided for reading either the number or its complement into AC.

3. To hold the multiplicand and divisor. No additional equipment is required for these services. The contents of AR are simply added or subtracted into AC in a normal fashion, but under the control of the process in question.

4. To sense the sign of the number in AR for use in determining the signs of products and quotients.

5. To change the sign of the number in AR by forming the complement of that number.

6. To hold for transfer to storage the location of the next instruction to which the computer is to return after a change in the program counter (sequence of instructions -- see Sections 2.12 and 2.13, above).

2.232 Accumulator

AC is the adding element used for all arithmetic operations. It forms the sum in addition, the difference in subtraction, the product in multiplication, and holds the remainder in division. It also performs the following computing procedures:

1. Read out to the bus. All information extracted from AE for storage or display must come from AC (except as indicated in Section 2.231-6, above).

2. Read into BR. The multiplier is originally in AC and must be transferred to BR before a multiplication can start. Since transfer by shifting is too slow, a set of transfer gate tubes is provided.

2.233 B-Register

BR holds the multiplier during multiplication and the quotient during division. It must be capable of:

1. Receiving numbers transferred into it from AC.

2. Transmitting numbers directly into AC to add the contents of BR to those of AC. On all other occasions, BR reads out to AC by shifting left.

3. Shifting its contents to the left or right, thereby inserting digits into or removing them from AC as the case may be. At the command "divide shift left," however, no digits pass between BR and AC since AC is not involved.

2.24 Input and Output System

The basic function of the WWI in-out system is to permit communication between the computer and people and equipment external to the computer. Communication with people is obviously necessary in order for instructions to reach the computer and for certain types of results to be interpreted. Communication with external equipment is needed if the computer is being used as a control element in a physical system or if some external storage is used to expand the facilities of the computer.

The computer requires its input and output data to be in very special form, namely, 0.1-μsec pulses in parallel lines representing some type of binary code. Input data may originate in many different forms, e.g., signals from a punched-tape or punched-card reader, signals from magnetic tape or magnetic drum, or voltages from analog measuring devices. Output data also may be required in different forms, such as pulses to operate display oscilloscopes and cameras, or relay voltages to operate typewriters and tape punches. The in-out system therefore must convert the form of the data and must match other characteristics of the terminal equipment to those of the computer.

Coordination between the computer and the terminal equipment is provided by the in-out (control) element (IOE) and includes the following specific functions:

1. Selecting external units and their modes of operation;

2. Providing buffer storage for data being transferred between the computer and external units;

3. Stopping the computer if it attempts to get ahead of an external unit;

4. Giving an alarm if an external unit gets ahead of the computer program;

5. Synchronizing external pulse signals with computer pulses;

6. Allowing certain external units to operate through several in-out processes or cycles in response to a single computer order;

7. Counting any delays needed between steps in the in-out procedure.

The above description gives only a birdseye view of the Whirlwind in-out system. Nothing has been said about the magnetic-drum installations as details of this system have not been completed. A separate volume covering the in-out system in detail is contemplated; specifications of various in-out devices are given in Section 1.2 of this Report, in Digital Computer Laboratory Engineering Note E-466, "Operation of the In-Out Element," by E.S. Rich, 15 July 1952, and in Digital Computer Laboratory Engineering Note E-520, "The WWI Auxiliary Magnetic Drum System," by J.W. Forgie, 9 Jan. 1953.

3. BASIC OPERATIONS

As the basis of its computations, WWI can perform upon command the fundamental operations of addition, subtraction, multiplication, and division. To adapt these operations to computer requirements, other operations essential to logical functioning have also been included. Among these latter are the shift (transposition of accumulator and B-register contents to right or left for a designated number of digits), the transfer (duplication of all or a part of accumulator contents in a designated storage location), the subprogram (see Section 2.12, above), and the conditional program (see Section 2.13, above).

3.1 Binary Arithmetic

The decimal system takes its name from the fact that it is based on ten digits (0, 1, ... 9); all numbers in the decimal system are composed of those 10 digits. The binary system, analogously, takes its name from the fact that it is based on 2 digits (0, 1); all numbers in the binary system are made up of those 2 digits. The decimal system has a base of 10; the binary system has a base of 2.

Decimal System	Equivalence	Binary System	Equivalence
1	$1 \times 10^{\circ}$	1	$1 \times 2^{\circ} = 1$
2	$2 \times 10^{\circ}$	10	$1 \times 2^{1} + 0 \times 2^{\circ} = 2$
3	$3 \times 10^{\circ}$	11	$1 \times 2^{1} + 1 \times 2^{\circ} = 3$
4	$4 \times 10^{\circ}$	100	$1 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{\circ} = 4$
5	$5 \times 10^{\circ}$	101	$1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{\circ} = 5$
6	$6 \times 10^{\circ}$	110	$1 \times 2^{2} + 1 \times 2^{1} + 0 \times 2^{\circ} = 6$
7	$7 \times 10^{\circ}$	111	$1 \times 2^{2} + 1 \times 2^{1} + 1 \times 2^{\circ} = 7$
8	$8 \times 10^{\circ}$	1000	$1 \times 2^{3} + 0 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{\circ} = 8$
9	$9 \times 10^{\circ}$	1001	$1 \times 2^{3} + 0 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{\circ} = 9$
10	$1 \times 10^{1} + 0 \times 10^{\circ}$	1010	$1 \times 2^{3} + 0 \times 2^{2} + 1 \times 2^{1} + 0 \times 2^{\circ} = 10$
11	$1 \times 10^{1} + 1 \times 10^{\circ}$	1011	$1 \times 2^{3} + 0 \times 2^{2} + 1 \times 2^{1} + 1 \times 2^{\circ} = 11$
12	$1 \times 10^{1} + 2 \times 10^{\circ}$	1100	$1 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{\circ} = 12$
13	$1 \times 10^{1} + 3 \times 10^{\circ}$	1101	$1 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{\circ} = 13$
14	$1 \times 10^{1} + 4 \times 10^{\circ}$	1110	$1 \times 2^{3} + 1 \times 2^{2} + 1 \times 2^{1} + 0 \times 2^{\circ} = 14$
15	$1 \times 10^{1} + 5 \times 10^{\circ}$	1111	$1 \times 2^{3} + 1 \times 2^{2} + 1 \times 2^{1} + 1 \times 2^{\circ} = 15$
16	$1 \times 10^{1} + 6 \times 10^{\circ}$	10000	$1 \times 2^{4} + 0 \times 2^{3} + 0 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{\circ} = 16$
17	$1 \times 10^{1} + 7 \times 10^{\circ}$	10001	$1 \times 2^{4} + 0 \times 2^{3} + 0 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{\circ} = 17$
20	$2 \times 10^{1} + 0 \times 10^{\circ}$	10100	$1 \times 2^{4} + 0 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 0 \times 2^{\circ} = 20$

Decimal numbers, since they have a base of 10, may be broken up into powers of 10:

e.g. $305.798 = 3 \times 10^{2} + 0 \times 10^{1} + 5 \times 10^{\circ} + 7 \times 10^{-1} + 9 \times 10^{-2} + 8 \times 10^{-3}$

In the same way, binary numbers, since they have a base of 2, may be broken up into powers of 2:

e.g. $101.011 = 1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{\circ} + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$

It can be seen from this arrangement of the powers of the bases that the decimal places (units, tens, hundreds, tenths, hundredths, thousandths, etc.) have a definite relation to the powers of the base 10 in the decimal system. They are numbered off consecutively from left to right, from $+\infty$ to $-\infty$, these numbers corresponding exactly with the powers of the base 10; the decimal point is placed between the units (0) place and the tenths (-1) place.

The binary places (units, twos, fours, eights, sixteens, halves, fourths, eighths, etc.) are also numbered exactly according to the powers of the base 2; the binary point is placed between the units (0) place and the halves (-1) place. Therefore, the place and point arrangement is the same in both decimal and binary systems.

3.11 Addition

It is necessary to consider the methods of binary arithmetic to understand the processes which WWI uses for computing. A detailed discussion of binary arithmetic is contained in Digital Computer Laboratory Report R-90-1, "The Binary System of Numbers," Margaret F. Mann, February 29, 1952.

In decimal addition (Figure 11A), when a sum in any column is greater than 9, the extra digit (or carry) must be added into the left-hand adjacent column. (Ordinarily, this carry operation is accomplished without writing it out specifically.)

(A)
Decimal

(B)
Binary

Figure 11. Addition

Figure 11B shows a binary addition. In a single column, the sum of 0 and 0 is 0. The sum of 1 and 0 is 1. The sum of 1 and 1, however, is 10, the 1 being a carry which must be added into the left-hand adjacent column with the possibility that it may produce another carry to be added in.

3.12 Subtraction; End-Around Carry

A computer must be capable of handling negative numbers; two methods are shown in Figure 12. For a discussion of binary sign notation for computer use, see Section 1.33, above.

Method	Decimal	Binary
ABSOLUTE MAGNITUDE, WITH SIGN	- 111	- 1101111
9's COMPLEMENT	9 888	
1's COMPLEMENT		1 0010000

Figure 12. Negative Numbers

The customary notation is to handle the number as its absolute magnitude with an associated sign. From the point of view of the computer, however, such a notation is not completely satisfactory since it requires a special subtracting unit and the ability to discriminate signs as such. Computerwise, a more convenient method of handling negative numbers is the N's-complement system, in which the magnitude of the negative number is subtracted from a power of the base N, less 1. The convenience may be noted from the illustration: the 9's complement of the decimal number is obtained by subtracting each digit of the positive number from 9. In binary notation, to form a negative-number 1's complement, each digit of a positive number is subtracted from 1, equivalent to interchanging all 0's and 1's.

By using the 1's complement system, an adder may be made to subtract. The first example of binary subtraction (Figure 13) shows a process in which the result is negative. The subtrahend has been complemented and added to the minuend. The result is the difference in its complemented form. The second example shows the subtraction of two numbers whose difference is positive. Once again, the subtrahend has been complemented and added to the minuend. In this particular case, however, the difference is not in the desired form. It consists of the desired difference, $N_1 - N_2$, plus the unwanted terms $2^n - 1*$. Fortunately, the correction for these terms is easy. The 2^n represents a digit off the left-hand end of the register. If, therefore, this digit is carried around and added into the rightmost place, both the 2^n and the -1 will be compensated for. This process, known as the "end-around carry," is easily executed physically by connecting the carry section of the leftmost digit of the accumulator to the adding section of the rightmost digit.

---

* $2^n = (1 \times 2^n)$ as shown in Part B, below. Part A, the familiar decimal breakdown, is given for reference.

A	B
Decimal	Binary
936 =	1...101 =
$(9 \times 10^2) + (3 \times 10^1) + (6 \times 10^0)$	$(1 \times 2^n) \ldots + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

BINARY

Minuend	+. 10110110	= N_1	
Subtrahend	-. 11010010	= $-N_2$	
Minuend	0. 10110110	= N_1	
Subtrahend	1. 00101101	= $2^n - N_2 - 1$	1's complement of N_2
Remainder	1. 11100011	= $2^n - (N_2-N_1) - 1$	1's complement of $N_1 - N_2$
Remainder	-. 00011100	= $N_1 - N_2$	

Minuend	0. 10110110	= N_1	
Subtrahend	-. 10010010	= N_2	
Minuend	0. 10110110	= N_1	
Subtrahend	1. 01101101	= $2^n - N_2 - 1$	1's complement of N_2
	10. 00100011	= $N_1 - N_2 + 2^n - 1$	1's complement of $-(N_1-N_2)$
	+1	= end-around carry	
Remainder	+. 00100100	= $N_1 - N_2$	

Figure 13. Subtraction Using 1's Complements

3. 13 Multiplication and Roundoff

Multiplication is the process of successive additions, the multiplicand being added to the partial product in a manner determined by the digits in the multiplier. Figure 14A shows an example of decimal multiplication. The rightmost digit of the multiplier is multiplied by the digits of the multiplicand in turn. It will be noted that the results of these small multiplications may be two-digit numbers, requiring, therefore, an addition process and carries to obtain the partial product. Following the use of the rightmost digit of the multiplier, the next digit to the left is used with the next partial product shifted one place left. The partial products formed are added together. Formation and addition of partial products is continued until all digits of the multiplier have been used.

In multiplication, binary notation presents many advantages, because the binary multiplication table is reduced to 0 x 0 = 0, 1 x 0 = 0, and 1 x 1 = 1. The largest product of any two binary digits is the single binary digit 1, and there is no carry in forming a partial product.

<pre>
        (A)   (B)
      Decimal   Binary

  22 Multiplicand   10110 Multiplicand
  19 Multiplier   10011 Multiplier
  18   10110 Partial product
  18   10110 Shift left and add (1 in multiplier)
 198 Partial product   1000010 Partial product
   2   00000 Shift left (0 in multiplier)
   2   1000010 Partial product
 418 Product   00000 Shift left (0 in multiplier)
                              01000010 Partial product
                                10110 Shift left and add (1 in multiplier)
                             110100010 Product
</pre>

Figure 14. Multiplication

Figure 14B shows an example of binary multiplication. It is carried out in the same fashion as the decimal example previously discussed except that each step is now simply a choice of whether or not to add in the shifted multiplicand. This choice is governed by whether or not the multiplier digit is a 1.

A modification of binary multiplication, helpful for mechanization of the process, is shown in Figure 15. The multiplication begins (Step 1) as before with the addition of the multiplicand (because the rightmost digit of the multiplier is a 1). In Step 2, however, both the multiplier and the partial product are shifted one digit to the right instead of the multiplicand only being shifted one digit to the left as in Figure 14B. (The rightmost digit of the original multiplier is no longer needed and is dropped.) The rightmost digit of the remaining part of the multiplier is examined and, in this case, is found to be a 1; therefore, the multiplier is again added in directly. This process of shifting multiplier and partial product to the right and adding in the multiplicand (without shifting it) if the rightmost digit of the multiplier is a 1 continues until the entire multiplier has been used (Steps 3, 4, 5).

<pre>
         10110   Multiplicand   1000010   Partial product
Step 1   10011   Multiplier
         10110   Partial product   10110   Multiplicand
                                    Step 4   10   Shifted multiplier
         10110   Multiplicand   1000010   Shifted partial product
          1001   Shifted multiplier   00000
Step 2   10110   Shifted partial product 01000010   Partial product
         10110   Add
        1000010 Partial product   10110   Multiplicand
                                              1   Shifted multiplier
         10110   Multiplicand   Step 5 01000010 Shifted partial product
          100   Shifted multiplier   10110   Add
Step 3 1000010 Shifted partial product 110100010 Product
         00000
        1000010 Partial product
</pre>

Figure 15. Modified Binary Multiplication

Since the product of two n-digit numbers can result in a 2n-digit product, some method must be provided for handling the right n digits. Although provision can be made for retaining them, in general they are insignificant and can be discarded. This process requires the proper rounding off of the number. In rounding off, if the last n digits are merely dropped off, the remainder will always be less than the complete number. In the course of a problem requiring a very large number of multiplications, the bias due to this procedure will become excessive. It is desirable to increase by 1 the rightmost digit that is kept if the part of the number discarded is greater than a half of this digit. A convenient way of accomplishing this in the decimal system is to add 5 to the first digit to be discarded. If this digit is 5 or greater, a 1 is carried in to the rightmost digit of the part of the number saved, then the rest is discarded; if this digit is less than 5, no 1 is carried into the rightmost digit of the part of the number saved and as before the less significant digits are discarded. This process is illustrated in Figure 16.

Figure 16. Roundoff

In binary notation a similar process can be used: a 1 is added to the most significant digit to be discarded, resulting in a carry if this digit is a 1 and in no carry if it is 0. (Figure 16B.) No provision is made for handling properly the ambiguous case in which the discarded part of the number is exactly one half. However, in the multiplication of two 16-digit numbers 16 of the 32 digits in the product will be discarded; the chances of this 16-digit discard being exactly one half are very small. The bias due to this process is equally small and can be neglected in most problems.

3.14 Division

Binary division is carried out by a process of successive subtractions in the same manner
as decimal division (Figure 17) is carried out on a small desk calculator. The divisor is sub-
tracted from the dividend as many times as possible until an overcast (negative remainder)
appears. In decimal division there are ten possibilities for the number of times the divisor may
be subtracted from the remainder; in binary division there are only two possibilities (0 or 1).
The net number of successful subtractions is the appropriate digit of the quotient. After any
subtraction causing an overcast, the divisor is added in, clearing the overcast, and the remain-
der is shifted one digit to the left. The process of subtraction is then repeated. In binary
division, after an overcast occurs, instead of (1) adding in the divisor, (2) shifting left 1, and
(3) subtracting the divisor again, it can be shown that this procedure is mathematically equiva-
lent to (1) shifting left 1 and (2) adding in the divisor.

```
            142
     28 |¯3976
          -28                    Sub. 1
          ────
          1176                   Pos. rem.
          -28                    Sub. 2
          ────
         -1624                   Overcast
         +28                     Restore -1
          ────
          1176                   Shift, 1 net sub.
          -28                    Sub. 1
          ────
           896                   Pos. rem.
          -28                    Sub. 2
          ────
           616                   Pos. rem.
          -28                    Sub. 3
          ────
           336                   Pos. rem.
          -28                    Sub. 4
          ────
            56                   Pos. rem.
          -28                    Sub. 5
          ────
          -224                   Overcast
          +28                    Restore -1
          ────
            56                   Shift, 4 net sub.
          -28                    Sub. 1
          ────
            28                   Pos. rem.
          -28                    Sub. 2
          ────
             0                   Pos. rem.
          -28                    Subtract 3
          ────
           -28                   Overcast
          +28                    Restore -1
          ────
             0                   Rem. 0, 2 net sub.
```

Figure 17. Decimal Division

Procedure for Binary Division (Figure 18)

1. Subtract the divisor from the dividend -- that is, add the 1's complement of the divisor.

2. Perform a carry operation (called "divide carry"). This may involve an end-around carry.

3. Examine the sign digit (leftmost digit) of the remainder. If sign digit is a 0 ("divide 0"), indicating that the remainder is positive, put a 1 in the quotient; if a 1 ("divide 1"), indicating that the remainder is negative, put a 0 in the quotient.

4. Shift the remainder one digit to the left (called "divide shift left"). The digit shifted off the leftmost digit position is used to fill in the vacated rightmost digit position. This digit is a 1 if the sign of the remainder (leftmost digit) was negative before the shift; a 0, if the sign of the remainder (leftmost digit) was positive. (A negative number is assumed to have 1's extending beyond the last digit, a positive number, 0's.)

5. If the remainder was positive (a 0 in the sign-digit position before the remainder was shifted), subtract the divisor from the shifted remainder; if the remainder was negative, add the divisor. (To simplify mechanization of binary division, the remainder is shifted left and the divisor is added in directly, instead of holding the remainder fixed and shifting the divisor right as is done in manual long division.)

6. Repeat the above operations, beginning at 2, until the quotient has the same number of digits as the dividend.

Figure 18 is an example of binary division. A divisor and dividend of the same number of digits are used because in WWI all numbers have 16 binary digits. Also, the divisor and dividend are both positive (0 sign digits) since the computer uses positive numbers during divide (if either or both were negative before the operation, they are made positive and sign corrections are made afterward). Finally, in WWI the registers cannot hold a number equal to or greater than 1; therefore the divisor must always exceed the dividend for a successful division.

```
              0. 10110
            ┌─────────
0. 10100    │0. 01110
              1. 01011      (a)  Subtract (add 1's complement of divisor)
            ─────────
              1. 0̶0̶1̶0̶1
               1̶ 1̶          (b)  Divide carry
            ─────────
              1. 11001           Divide 1 (sign digit of remainder was a 1 (negative);
                                   0 in quotient)

              1. 10011      (c)  Divide shift left
              0. 10100      (a)  Add divisor (overcast has occurred)
            ─────────
              1. 0̶0111
               1̶            (b)  Divide carry
            ─────────
            10. 00111
              └──────→ 1         End-around carry

              0. 01000           Divide 0 (sign digit of remainder was a 0 (positive);
                                   1 in quotient)

              0. 10000      (c)  Divide shift left
              1. 01011      (a)  Subtract
            ─────────
              1. 11011
              0.            (b)  Divide carry
            ─────────
              1. 11011           Divide 1 (neg. remainder; 0 in quotient)

              1. 10111      (c)  Divide shift left
              0. 10100      (a)  Add divisor (overcast has occurred)
            ─────────
              1. 00011
              1. 1          (b)  Divide carry
            ─────────
            10. 01011
              └──────→ 1         End-around carry

              0. 01100           Divide 0 (pos. remainder; 1 in quotient)

              0. 11000      (c)  Divide shift left
              1. 01011      (a)  Subtract
            ─────────
              1. 10011
              1             (b)  Divide carry
            ─────────
            10. 00011
              └──────→ 1         End-around carry

              0. 00100           Divide 0 (pos. remainder; 1 in quotient)

              0. 01000      (c)  Divide shift left
              1. 01011      (a)  Subtract
            ─────────
              1. 00011
              1             (b)  Divide carry
            ─────────
              1. 10011           Divide 1 (neg. remainder; 0 in quotient)
```

Figure 18.   Binary Division

3.15 Sign Control

To avoid complexity of logic and equipment, the operations of multiplication, division, and shifting are actually carried out only with positive numbers in AE. If these operations are called for with negative numbers, there must be a control which:

1. Recognizes whether numbers to undergo these operations are positive or negative;
2. If numbers are negative, makes them positive;
3. If numbers have been made positive, remembers whether one or both have been adjusted, so that the sign of the answer is corrected accordingly.

At the commands "AR sign check" and/or "AC sign check," the sign-control FF (seen in Figures 26, 31, and 32) "remembers" the indications of sign changes in AR and AC and makes any necessary adjustments to the sign of the answer in AC.

3.16 Shifting and Cycling

The process of transferring a number to the right or left within a register (or pair of registers) is called shifting and is equivalent to moving the binary point. Thus, shifting right is the same as multiplying by negative powers of 2; shifting left is the same as multiplying by positive powers of 2. Since WWI can shift only positive numbers, negative numbers are complemented to make them positive before shifting occurs. If a number is shifted one digit or more to the right, each digit is transferred to the next digit space on the right. The rightmost digit is transferred off the register and lost. For each shift right, a single 0 is inserted in the vacated digit space to the right of the sign (leftmost) digit. (Note that the sign digit itself is not shifted in this operation.) If a number is shifted one digit or more to the left, the digit to the right of the sign digit is transferred off the register and lost. Each shift left vacates the rightmost digit space which is allowed to remain blank, tantamount to filling it with the positive digit 0. (Note again that the sign digit is not shifted.) Either shifting right or shifting left may be extended by any number of digits up to 31 (which with the sign digit equals the total register length of AC and BR, specially combined for shifting and cycling in order to handle the double-length word stored within the two registers).

There are two modes of shifting to be distinguished: the operations shift left and shift right (discussed above) are always performed with positive numbers (negative numbers are made positive and their signs later restored as in multiplication and division); however, the commands shift left and shift right, as called for with slight alterations in division (divide shift left) and multiplication (shift right and carry), will accommodate negative numbers as well as positive numbers because the shifted-off digits in these cases are always physically carried around and used to fill in the digit spaces vacated at the opposite end of the register. (It should be noted that in multiplication and division shifting cannot occur between AC and BR since the contents of these registers perform separate functions in the complete operation; all shifts take place separately within each register.)

Cycling, except that it is performed without regard to sign, is similar to shifting left. In the cycling operation, the contents of AC, including the sign digit, are transferred to BR digit by digit (from the left-hand end of AC (AC 0) to the right-hand end of BR (BR 15)). At each transfer, the leftmost digit of the word in BR (BR 0) moves into the rightmost position of the word in AC (AC 15).

3.2 Computer Procedure for Basic Operations

In the introduction to this Chapter, certain arithmetic and logical operations were identified as fundamental to WWI computation. The manner in which the computer carries out these operations is described below. (The complete arithmetic element, where all WWI computations are performed, is shown in Figure 67).

3.21 Addition

3.211 Simple Adder

Figure 19 shows a simple binary adder, corresponding to two digits of AC. The digits of one of the numbers (to be added to) are held in the partial-sum FF's (FF1, FF2). The digits of the second number (to be added in) are transmitted from AR on the "add" lines to these FF's. (The blocks DE represent the inherent delay of the FF in switching and are shown only in those cases where they are logically significant.) When a 1 on the "add" line is added to a 1 in the PS FF, a carry signal goes through GT .01 to the carry FF directly above the partial-sum FF in which this addition occurs and the partial-sum FF is complemented (to 0) after waiting out the delay DE. The carry later is added into the partial-sum FF in the digit column to the left when the carry line is pulsed. These carry additions may produce new carries requiring many more carry additions. However, a system called "high-speed carry" has been developed in which all row-wise carries are consolidated in one addition.
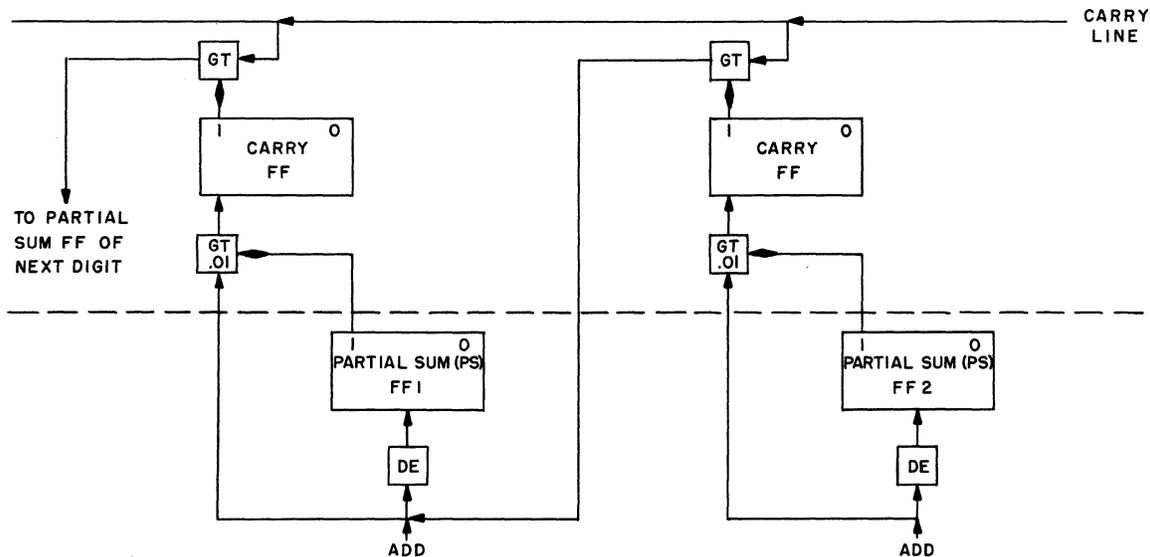

Figure 19. Simple Binary Adder

3.212 High-Speed Carry

It can be seen in Figure 20 that a new carry will be produced only if the carry being added to an adjacent partial-sum FF finds this FF already holding a 1. The resulting carry must be added to the next adjacent partial-sum FF. In WWI, the original carry digit is allowed to go

Figure 20. High-Speed Carry

past any partial-sum FF's holding 1's, complementing them as it passes, until it reaches and is added into a partial-sum FF holding a 0. The time-consuming side trip through each carry FF is thus avoided. Figure 20 shows the complete procedure.

3.213 Clear and Add -- ca (Figure 21)

The purpose of ca is (1) to clear the accumulator (AC) of any number it may hold from a previous problem; and (2) to add into AC a new number (the number contained in the storage register designated by the address section of the instruction). The sequence of significant commands is:

1. Clear AR (to receive new number from storage via the bus).
2. Read in from bus to AR.
3. Clear AC (which might be done at the same time as either 1 or 2).
4. Add (transferring contents of AR into AC). No further commands are necessary.

Figure 21. Clear and Add

3.214 Add -- <u>ad</u> (Figure 22)

The purpose of <u>ad</u> is to add the number contained in a specified storage register to a number already in AC. The sequence of significant commands is:

    1. Clear AR (to receive new number from storage via the bus).

    2. Read in from bus to AR.

    3. Add (contents of AR to AC).

    4. Carry (pulse the carry line once). The sum will then be in the AC partial-sum FF's. AC itself will take care of any end-around carries.

One problem, that of the arithmetic check, still remains. If the sum of the two numbers added exceeds unity, the register capacity of the computer will have been exceeded, and the computer will recognize the overflow with an overflow alarm.

    5. Arithmetic check for overflow.

Figure 22. Add, Arithmetic Check

This check for overflow is carried out in the following manner:

1. If the numbers added together are both positive, the sum will be positive. If there is an overflow, a 1 will appear in the sign-digit space after the carry (which will indicate an erroneously negative answer or a number greater than unity).

2. If one number is positive and the other negative or if positive numbers are subtracted, then the sum can never exceed in magnitude the larger of the two numbers. These numbers are each presumed to be less than unity; therefore, the sum can never be greater than unity, and an overflow cannot occur.

3. If both numbers are negative, the sum should be negative. The 1's in the sign digits of the negative numbers when summed result in a 0 in the sign digit of AC (the answer) plus a carry in this column (left-digit carry). This 0 sign digit should be replaced by a carry from the right to change it to a 1, indicating a correct negative answer. If no such carry appears, the sign digit will remain 0, representing an overflow.

3.22 Subtract -- su (Figure 23)

The su operation is the same as ad except that the complement of the number in AR is added to the number in AC. The arithmetic check is the same as for ad. The sequence of commands is:

1. Clear AR (to receive new number from storage via the bus).
2. Read into AR from bus.
3. Subtract (adding negative (complement) of AR contents to AC).
4. Carry (pulse the carry line once).
5. Arithmetic check for overflow.

Figure 23. Subtract; Arithmetic Check

3.23 Multiplication

3.231 Computer Procedure

The basic machine essentials for multiplication are shown in Figure 24. The multiplicand is stored in the A-register (AR), while the multiplier is stored in the B-register (BR). The product is built up in the adding unit, the Accumulator (AC). Gate tubes (GT) are provided for adding the contents of AR into AC. High-frequency clock pulses (2mc) are supplied at the bottom line. If the rightmost digit of the multiplier (BR 15) is a 0, GT 2 will be on and GT 1 will be off. The first clock pulse then will pass through GT 2 and cause the partial product in AC and the multiplier in BR to be shifted one digit to the right. A new multiplier digit will be put in the "sensing" position (indicated by crosshatching).

Figure 24. Multiplication

If the rightmost digit of the multiplier is sensed as a 1, GT 1 will be on and GT 2 will be off. The first clock pulse will pass through GT 1 and add the contents of AR into AC. This pulse after passing GT 1 will also return to the input of the rightmost digit of BR (BR 15), changing it from a 1 to a 0. The next clock pulse will find GT 2 on and will shift instead of add. At each shift of the process, the partial product in AC and the multiplier in BR are shifted. The digit which would ordinarily be shifted off AC (AC 15) is put into the now empty space in the left-hand end of BR (BR 0). If the multiplier digit is a 1, the multiplicand is added to the partial product prior to the shift. If the multiplier digit is 0, the shift only is performed.

Central control is used for directly commanding some of the simpler operations such as addition and subtraction. More complicated operations such as multiplication may be more simply and efficiently commanded by using a separate control as shown in Figure 25. A flip-flop (FF) is used to control a GT fed by 2-mc pulses on its other input. When a multiplication is desired, a multiply pulse (command) is sent into this FF, which sets it to a 1, turning on the GT and sending high-frequency clock pulses to the gate-tube combination (GT 1 and GT 2) previously mentioned in Figure 24 for performing the successive additions and shifts of the multiplication. A counter stops the additions and shifts when the multiplication is complete. At each shift, this counter is indexed one point. Although the number of additions varies according to the number of 1's and 0's in the multiplier, the number of shifts is constant for any multiplication.

Figure 25. Multiplication Control

When the counter is full, it will send an overflow (end-carry) pulse down to the FF, which will set it to 0, turning off the clock-pulse GT and stopping the multiplication. The counter has been set so that the multiplication operation will be stopped after all digits of the multiplier have been shifted to the rightmost position and sensed.

3.232 Multiply and Round Off -- mr (Figure 26)

In the mr operation, the number held in a specified storage register is to be multiplied by the number in AC. The product is to be rounded off to a single register length, the insignificant digits being discarded.

The multiplication operation is outlined in Section 3.231. The multiplicand will be in AR when received from the bus. The multiplier should be in BR and must be transferred there from AC.

Figure 26. Multiply and Round Off

The initial commands are:

1. Clear AR and BR (to prepare them for receiving multiplicand and multiplier).

2. Read in from bus to AR (putting multiplicand in AR).

Before transferring the multiplier from AC to BR, consider the problem of the multiplication of negative numbers. The multiplication process that has been described is only for positive numbers. The product of negative numbers represented in the 1's complement becomes very complicated. For this reason AE is designed to multiply positive numbers only. Negative numbers are changed to positive numbers prior to a multiplication, and the sign of the product is changed after the operation if necessary.

The further commands in preparation for a multiplication are:

3. AC sign check (to check sign of multiplier: make negative number positive; note the "left-digit 1" line going from AC to sign control and the AC complement line going from sign control to AC to make AC positive in case it was negative).

4. AC Read to BR (transferring positive multiplier from AC to BR).

5. Clear AC (ready for product).

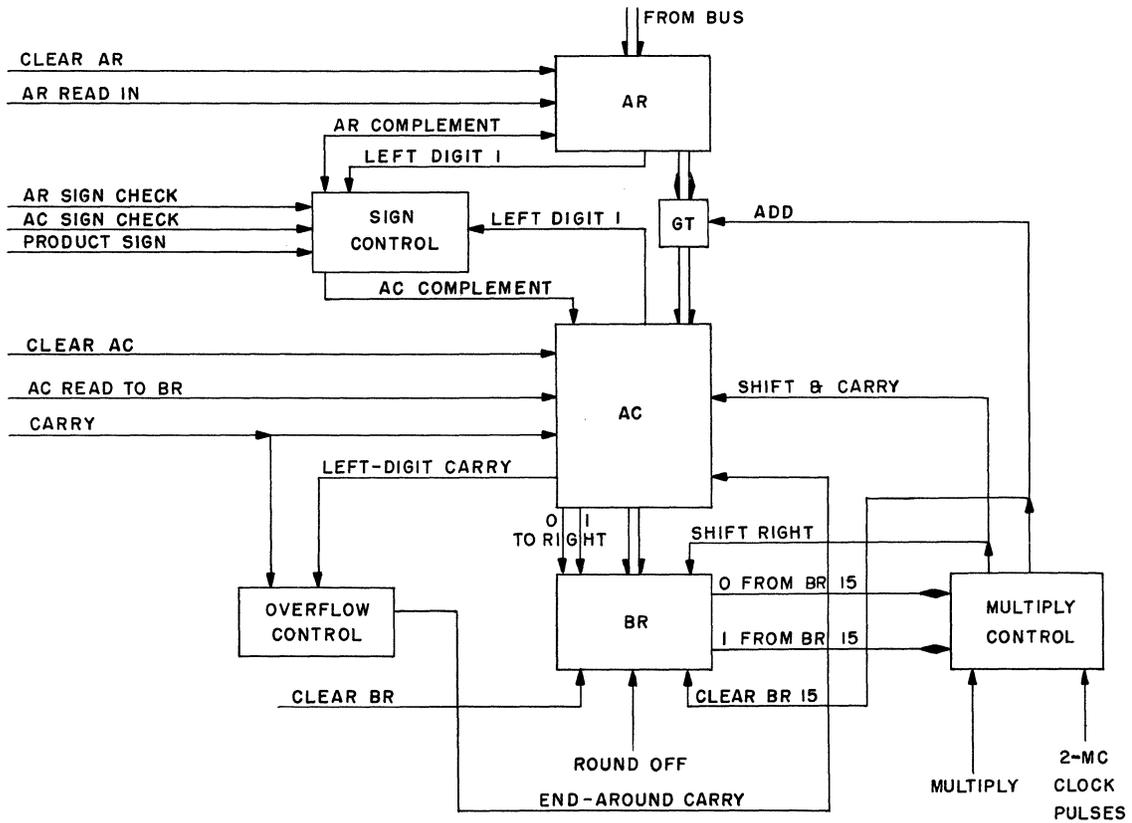6. AR sign check (to check sign of multiplicand; make negative number positive; note the "left-digit 1" line going from AR to sign control and the AR complement going from sign control to AR to make AR positive in case it was negative).

7. Multiply (and set step counter).

When the proper number of steps have been performed (15 for the 16-digit WWI), the step counter (associated with multiplication control) puts out an end-carry pulse which stops the operation.

The original multiply pulse (command) which was also fed to multiplication control stopped the flow of clock pulses to the time-pulse distributor in central control. The rest of the computer has been waiting during the high-frequency part of the operation. The end-carry pulse from the step counter not only stops the high-frequency clock pulses to multiplication control but also turns on the time-pulse distributor of central control, restarting the main part of the computer.

The contents of AC do not as yet constitute the final product. There may be some carries left in the carry flip-flops, since only single carries have been performed. The next commands are then:

8. Carry (high speed).

9. Round off.

10. Clear BR.

11. Product sign (and complement AC if necessary; note the "left-digit 1" line going from AC to sign check and the AC complement line going from sign control to AC to adjust sign of product in AC).

3.233 Shift (right) and Carry (Figure 27)

The accumulator must be able to shift right, not only because of the general value of this operation, but also because it is a basic part of the multiplication procedure. A special modification of shift right, very valuable in reducing multiplication time, consists of the single-carry and shift-right commands performed simultaneously with only one setting of the AC partial-sum and carry flip-flops.

With subcommand shift-and-carry, the new contents of the partial-sum and carry flip-flops of any digit column following a single carry are determined by the prior contents of the same partial-sum FF and the carry FF of the next digit to the right. This is so because the contents of the carry FF are added to the partial-sum FF, the sum of the two determining the contents of the digit following the carry.

Following both the shift and the carry, the contents of any digit column are completely determined by the original contents of the same carry FF and the partial-sum FF in the next digit column on the left.

The system for shift (right) and carry, shown in abbreviated form in Figure 27, is also satisfactory for simple shift right in which the carries are all 0's.

Note that if both partial sum and carry FF's hold a 1, a 0 goes to the right and the 1 is left in the carry FF.

Figure 27. Shift and Carry

In addition to its use in the operation s̲l̲, the command "shift left," Figure 28, is used in division ("divide shift left"). No circuit change is required for shifting left in division, as opposed to the case of shifting right in multiplication (in which the command "shift right and carry" necessitates additional elements for the carry process).
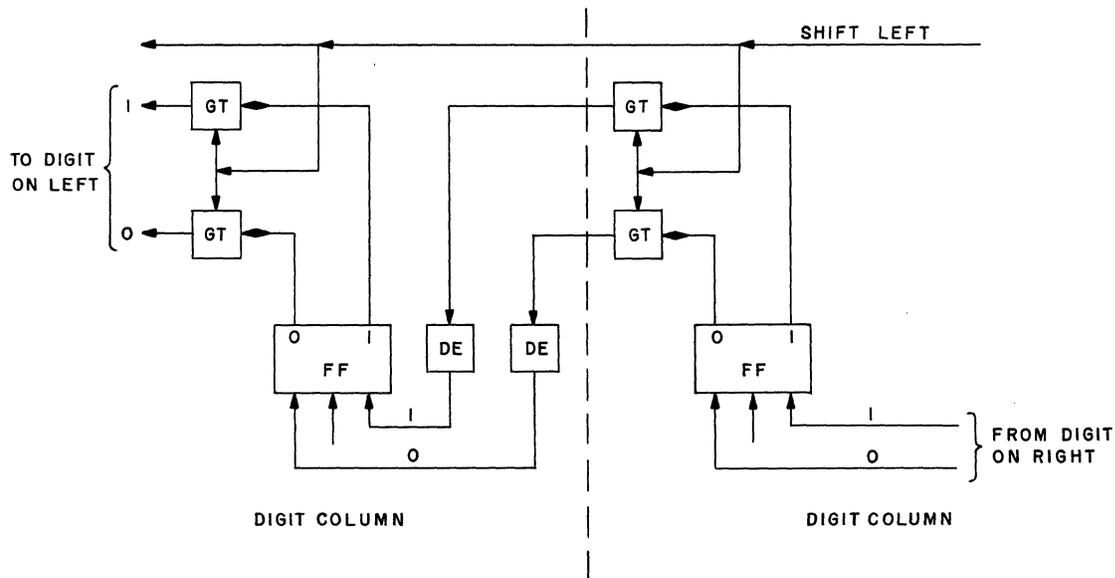

Figure 28. Shift Left

3.24 Division

Before the computer procedure for divide is explained, several points discussed briefly in Section 3.14 (the arithmetic of the divide operation) must be expanded.

3.241 Divide-Error Alarm

It was stated that the computer cannot hold numbers equal to or greater than 1; therefore, the divisor must be larger than the dividend. Although it is up to the programmer to see that such a situation is the actual case, it is quite conceivable that two numbers might be used, the results of intermediate operations, whose relative magnitudes could not be anticipated accurately every time. If the case arose where the divisor were smaller than the dividend, the division process would proceed along satisfactorily, but the answer would indicate an overflow had occurred. It is more desirable to anticipate this overflow than to allow the improper division to be completed, since an alarm due to an overflow in division would then be not immediately distinguishable from an alarm due to an overflow in addition or subtraction.

Accordingly, the method used in WWI is to examine the result of the first subtraction of the divisor from the dividend. If the remainder is positive, the dividend is greater than the divisor and an overflow will occur if the division is allowed to finish. Instead, the machine is

made to stop at the start of such a division with a divide-error alarm. The presence of a 0 in the sign-digit position of the remainder in AC after the first subtraction indicates to the divide-error alarm circuit that a positive remainder exists and that an improper division has been attempted.

3.242 Divide Control

An examination of Figure 18 will indicate that there are two procedures continually but alternately required in the arithmetic of division: divide carry and divide shift left. For any operation in the computer where repetitive procedures must be carried out, it is more efficient to command them from a control specifically for the purpose rather than from central control. Hence the arithmetic of the divide operation is commanded by a Divide Control, Figure 29. At the command, divide, from central control, the TPD in central control stops, and divide control takes over.
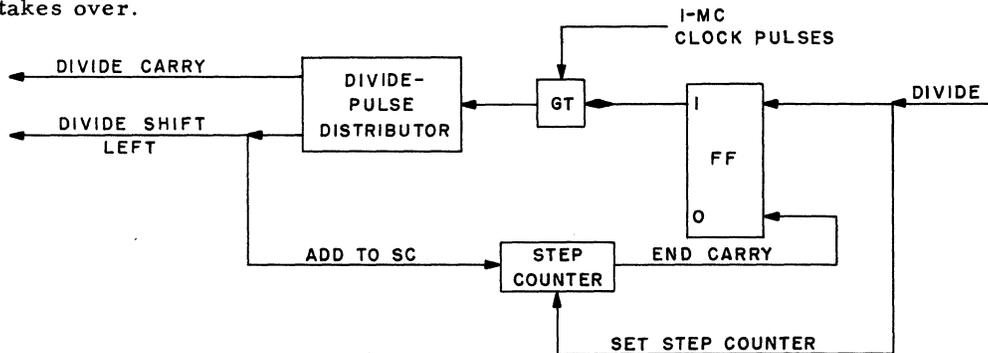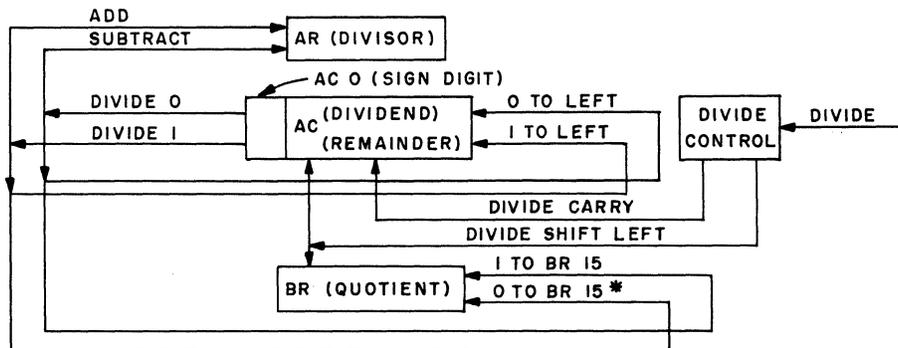

Figure 29. Divide Control

Figure 30 shows that a pulse distributor in divide control distributes alternately the subcommands, divide carry and divide shift left.

* Note that the line "0 to BR 15" is actually nonexistent because BR 15 is a 0 unless 1's are shifted into it by "Divide 0"; it is shown here to complete the theoretical logic.

Figure 30. Mechanized Arithmetic of Division

The arithmetic of division (Figure 18) also requires the subcommand add or subtract after the divide carry and divide shift left. These subcommands are directly dependent upon and actually emanate from the sign-digit position of the remainder in AC. Note that in Figure 31 a positive remainder (divide 0) will command a subtract and a negative remainder (divide 1) will command an add. The divide pulse distributor is timed so that after this add or subtract subcommand it then commands the divide carry and divide shift left. Of course, the sign of the remainder in AC also indicates whether the division was successful or not (whether a 1 or a 0 is put into the quotient in BR), and it is the sign digit which is shifted off the left end of AC and filled into the right end of AC at the subcommand divide shift left.

One more thing should be noted concerning the divide-shift-left subcommand: it does not cause a shift from BR0 to AC15; there is no connection whatever between AC and BR as there is in a conventional shift-left procedure.

The step counter (see Figure 29), indexed with each divide-shift-left pulse, keeps track of the progress of the arithmetic and will send an end carry to divide control to stop the division arithmetic, and to central control to restart the TPD, when the division arithmetic is complete.

3.243 Computer Procedure

The division operation begins with the dividend already in AC (as the result of some earlier operation). The divisor is brought into AR from a specified storage register via the bus. The necessary equipment for the operation is shown in Figure 31.
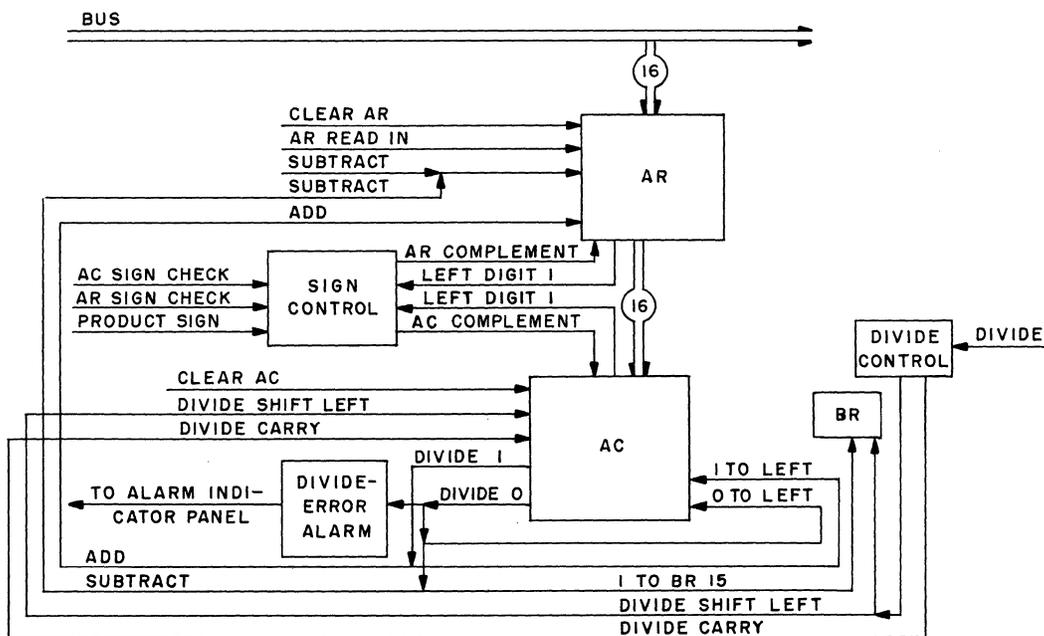

Figure 31. Divide Operation

The sequence of commands is:

1. Clear AR (prior to receiving the divisor from the bus).

2. Read into AR from the bus (putting the divisor in AR).

As in multiplication, it is more convenient to operate with positive numbers only. The divisor and dividend are made positive, and the desired sign of the quotient stored in the sign-control FF (see Section 3.232):

3. AC sign check (to check sign of dividend in AC and make negative AC positive).

4. AR sign check (to check sign of divisor in AR and make negative AR positive).

5. Subtract. (This first subtraction in the division procedure must be commanded by central control since divide control which is to take over on the next command, divide, presumes that a subtract (or an add) has already taken place when it puts out its first subcommand, divide carry. It is the result of this first subtraction that is sensed by the divide-error alarm circuits for an improper division.)

6. Divide. (On this command from central control, command is transferred to divide control which initiates the divide subcommands discussed in Section 3.242.)

When the arithmetic of division is complete, command is given back to central control:

7. Clear AC. (After the division arithmetic is completed, AC will hold a remainder including possible carries. The contents of AC are worthless and may be cleared. It is necessary to clear AC to assure at least the sign digit's being 0 prior to the command "Product Sign.")

8. Product Sign. (The product-sign pulse now gives the proper sign belonging to the quotient in BR to AC since BR of itself does not have provision for maintenance of a sign indication.)

The division process is now complete. The quotient (with one extra digit for roundoff purposes) is in BR and the sign of the quotient is in AC. A quotient in BR is, however, inaccessible to any further operations. Therefore at some time the quotient must be shifted from BR into AC where it will be accessible. At that time it will take its sign automatically from AC, which has been adjusted properly as a result of the product-sign command to AC (Section 3.25). This shifting of the quotient from BR into AC is accomplished by a separate instruction (shift left 15 and round off), which must always follow a divide instruction in the program.

3.25 Shift Right -- sr (Figure 32)

In the sr operation, the contents present in AC and BR as the result of some previous operation are shifted to the right. The digits shifted off AC are put into the left end of BR, hence the 1-to-right, 0-to-right lines from AC to BR in Figure 32. The digits shifted off BR are simply lost; zeros are inserted in the vacated left end of AC. (To simplify the corrections necessary for the shifting of negative numbers, the numbers are made positive prior to shifting and their sign corrected after shifting.) Following the shift the number is rounded off and BR is cleared.
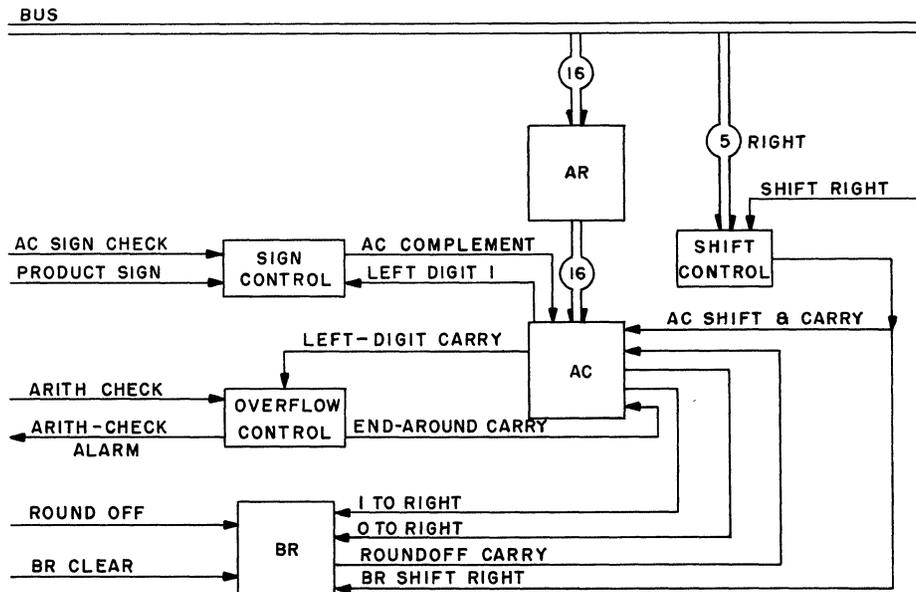
Figure 32. Shift Right

Information as to the number of places to be shifted is given in the address section of the sr instruction. The "address" is always sent to both the step counter and the storage switch. The step counter is thus properly set to count the number of shifts. The storage switch is also set to the "address" (equal to the number of shifts desired), but no harm is done since the storage register selected by the storage switch is not read. When the step counter is to be used with some operation other than shifting, it is reset to the desired quantity. When the step counter is not used, its contents are irrelevant.

Assuming that the step counter has been previously set from the bus by the address section of the instruction to the number of digit shifts desired, the sequence of commands is:

1. AC sign check (changing sign of number in AC to positive if necessary). It is not necessary to change the sign of BR, since its number is always obtained as the result of some operation involving only numbers whose signs have been made positive. In correcting for sign following these shifting operations, the contents of AC only are changed, not those of BR. The sign of the number in AC is assumed to belong to the quantity in BR. In BR, therefore, a negative number is represented by its absolute magnitude plus sign -- that of AC. The contents of BR will be adjusted for sign to that of AC only when and if BR is shifted into AC for further use.

2. Shift right.

The shift control, Figure 33, now takes over, the TPD in central control having been stopped, and produces the subcommands necessary for proper functioning of the shift operation:

AC shift (right) and carry (the carries are all 0's), BR shift right.

The step counter will count the number of shifts. When the desired number have been counted, the step counter will produce an end carry, which will shut off the supply of high-frequency clock pulses to the shift control. The end carry will also restart the TPD in central control as after the multiplication and division operations.

3. Round off. (The round-off pulse will add 1 to the high-speed-carry input of AC 15 if BR 0 is a 1. The high-speed-carry system will take care of any carries resulting from this addition.)

4. Arithmetic Check (for overflow due to round off).

5. Product sign. (Corrects sign of AC if necessary.)

6. Clear BR. (The contents of BR are worthless following the round off.)

Figure 33. Simplified Shift Control

3. 26 Transfer to Storage -- ts (Figure 34)

The ts operation sends the contents of AC out onto the bus, from which they are sent to a storage register designated by the address section of the instruction.

AC is not cleared, because its contents may be of immediate as well as future use. A number sent into magnetic storage (MS) must go through PAR. Therefore, an order to read in from the bus to PAR must be given, followed by the order to write the information in PAR into MS.

The sequence of commands is:

1. Pulse the to-bus line on AC, sends number in AC onto bus.

2. Read in from bus to PAR (PAR in) if MS, or Read in from bus to TS (TS in). (Choice made by Storage Selection Control, Section 5. 4.)

3. MS write (according to PAR contents).

Figure 34. Transfer to Storage

4. CHECKING

4. 1 General Considerations

The need for checking any sort of calculations is immediately evident, as a body of results is worthless unless some guarantee of accuracy exists. This is particularly true for computing machines where the amount of calculation may be enormous and the slightest mistake anywhere may destroy the answer.

Furthermore, under optimum conditions the computer must check itself, since the body of computation is apt to be too large to be checked efficiently by any other means. Nor would the purpose of the machine be served if any appreciable amount of computation within its capabilities devolved upon the operator. In addition, the computer must check itself while in operation in order to discover errors as they occur and prevent previous correct results from being disturbed.

Checking methods applied by the machine fall into two classes:
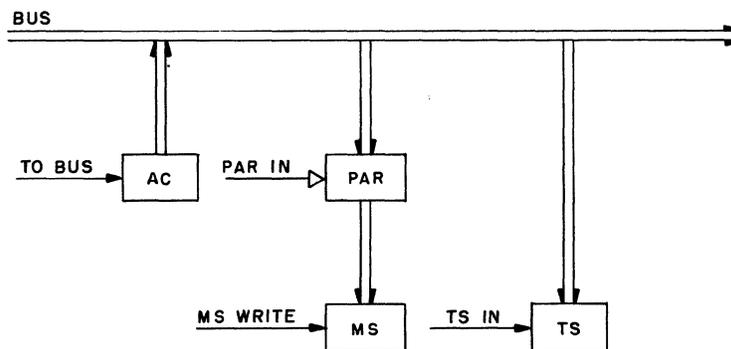
1. Built-in checks: arithmetic check, transfer check, and parity check

2. Programmed checks: spot check, mathematical check, check instruction, and marginal checking.

4. 2 Arithmetic Checks: Overflow and Divide-Error

It is possible for the operator as well as the machine to make mistakes. As discussed in Section 1.32, the scale factor must be set by the operator so as to be consistent and to keep every number for computer consideration within the finite register length of the machine. Since it may be difficult to place a reasonable upper bound on some of the partial results of a complicated problem, the operator may make the error of allowing some number to exceed or overflow register capacity. Under the circumstances, the machine must recognize the overflow and stop the computation to allow a correction to be made; otherwise the value of the computation will be destroyed. A divide operation must be stopped if the divisor is less than the dividend; in such a case, the quotient would be greater than 1 and overflow the register.

4. 3 Transfer Check

The single bus system transmits all words (numbers and instructions) between elements of the computer. These elements are connected to the bus by sets of gate tubes, any of which may be "on" or "off" as required. Ever-present possible sources of error are the failure of a gate tube to transmit a digit, the loss of a digit by reason of intermittent or permanent failure of the bus, or the generation of a spurious pulse.

For these reasons and as a practical expedient, those bus transfers common to all operations are checked by reading the words from the receiving element back into the bus to a special register reserved for checking purposes. This register has already received the original word via an entirely different bus and gate-tube system. Failure of the two words to coincide will halt further operations. This transfer-check system also checks flip-flop performance in the receiving element.

4.4 Parity Check

The parity of a binary word is the oddness or evenness of the sum of its digits. Since it can be safely assumed that only single errors are probable within any one word, the parity check or MS check indicates whether the parity of a word as stored for each MS register is the same as the parity of the word as read for that register.

4.5 Spot Check

The spot check is of value only in finding steady-state failures. This check consists largely of previously set up check sequences and complete check problems to be performed during maintenance time or to be programmed by the operator for inclusion in any given problem. Spot checking is usually done only in conjunction with marginal checking (q.v.).

4.6 Mathematical Check

This is a smoothness check on results, check by repetition of problems using different mathematical methods, or a special check designed for the particular problem being considered. The computer carries out the calculations necessary to the check in accordance with a program prepared by the operator.

4.7 Check Instruction

Spot checks and mathematical checks can be more easily performed if a means is provided for stopping the computer when a calculated result is not identical with a known result (in the case of a spot check) or with a previously calculated result (in the case of a mathematical check). The check instruction performs such a check, when programmed, by a process of comparison, as in a transfer check.

4.8 Marginal Checking

Whirlwind I has been designed primarily for applications which involve real time, in which an error can cause considerable damage in the real world (for example, civilian air-traffic control). Whirlwind I uses a marginal-checking scheme (with spot-check techniques) for detecting deteriorating components before they can cause errors. But since a spot check finds only steady failures, a voltage-variation system raises or lowers supply voltages in any one of several-hundred isolated sections of the machine to the point where steady failures occur. Failures are indicated by errors in a special test program performed repeatedly by the computer at the same time as the voltage is varied. The difference in voltage between normal value and failure value indicates the operating margin. If the margin for a particular section remains constant from day to day, it is assumed that no components in that section of the computer are deteriorating toward a point which would cause failure. Special switching equipment allows all sections to be tested automatically.

5. MAGNETIC-CORE STORAGE*

One recent development which is significantly raising the reliability of today's high-speed automatic digital computer is the multicoordinate magnetic-core memory, or storage. Two banks of 32 by 32 by 17 magnetic-core storage have been in full-time operation in the Whirlwind I Computer for some months. A description of the units and of the tests and operational data available on them will be preceded by a short review of the operating principles of this type of storage.

5.1 Operating Principles[1]**

Each binary digit is stored in the magnetic field of a small, ring-shaped, ferromagnetic core. Two aspects of the core's rectangular flux-current characteristic are utilized:

a. The flux remanence of the core is utilized for the storage operation;[2]

b. The extreme nonlinearity of the flux-current characteristic is utilized to advantage in the selection operation.[3,4]

Fig. 35 shows the flux-current loop for a ferrite core. The remanent flux points are arbitrarily designated as ZERO and ONE. Note that the loop is sufficiently nonlinear so that the application of $I_m/2$ cannot switch the core, whereas the full I_m can. Fig. 36 illustrates how this nonlinearity may be used to select one core out of many by the coincidence of two half-currents in a 2-coordinate scheme. The extension to three coordinates may be accomplished by stacking planes like those of Fig. 36 behind each other and connecting respective x and y coordinate lines in common as shown for x_2 and y_1 in Fig. 37.

The application of a half current to the coordinate x_2 results in the half excitation of a "selection plane" through the "volume." The same is true for the coordinate y_1, and the result is full-current excitation of the line of cores at the intersection of these two selection planes. The internal storage for a parallel-type machine might well resemble Fig. 37, and the selected line of cores might well represent the selected storage register, or word. A read-out or sensing winding threaded through every core in each z plane, or digit plane, would bring out the signal representing the stored digit. This part of the read operation is destructive, and the word must be rewritten.

For the rewrite part of the cycle the selection technique remains the same, except that the half currents on the selection planes are now in the write polarity, which would result in the writing of ONE's into all the cores of the selected register; this writing is controllable for each z, or digit, plane by the use of a digit-plane winding on which may be applied a half current of an effective polarity opposite to the write currents. The presence of this "inhibit" current in any digit during the write operation leaves a ZERO; absence of the inhibit current

---

* Sections 5.1-5.3 have been reproduced from a talk given by W.N. Papian at the Joint Eastern Computer Conference in Washington, D.C., December 8, 1953.

** Superscripts refer to similarly numbered entries in the Bibliography at the end of this section.

Figure 35.   Flux-Current Charac-
teristic of Ferrite Toroid

Figure 36.   Two-Coordinate Array

Figure 37.   Three-Coordinate Write

leaves a ONE.

R.R. Everett of MIT has shown that these techniques may be extended into any number of coordinates but that the 2-coordinate read and 3-coordinate write system just described is one of the most desirable for the Whirlwind type of machine.

5.2 Description of Whirlwind I Storage

The capacity of each storage bank is 1024 registers, with 16 digits (plus 1 parity digit) per register. The basic operating mode, or cycle, consists of setting the storage-address register to the new address and applying the read-current pulses, followed by the write currents for rewriting the information just removed. The information is stored in a storage-buffer register. The speed of the machine may be judged from the timing diagram (Fig. 38). Note that the read-rewrite time, or cycle time, is approximately 9 microseconds (recently reduced to 8 microseconds) and that there are no restraints on how frequently this cycle may be applied to the storage. It is capable, therefore, of a basic repetition rate of over 100 kilocycles per second. Note also that the information can be available to the machine approximately 2 microseconds from the beginning of the cycle.

Figure 38. Storage Cycle

5.21 Block Schematic

Fig. 39 is a simplified block schematic of one bank of core storage. Each half of the binary address in the address register is translated to a 1-out-of-32 selection by a crystal-diode matrix and sets up a pair of "AND" gates for x and a pair for y. The read flip-flop forms a 1.5-microsecond pulse and sends it to the two selected read drivers which supply the 0.45-ampere currents to two selection planes. The output signal voltages from each digit plane are amplified in the sense amplifiers and applied to "AND" gates which are strobed at the optimum time by a short (0.1-microsecond) pulse. Pulses representing ONE's then go off to set the buffer register to the just-extracted number. At the end of the read currents the rewrite part of the cycle starts in the same manner, except that the write currents have to be safely overlapped by the inhibit currents at those digit planes where ZERO's are to be written. This is accomplished by having the "on" time of the inhibit flip-flop overlap slightly that of the write flip-flop. Short (1-microsecond) currents may be applied to all digit planes after the rewrite; they are called post-write disturb (PWD) currents and are used to improve the ONE-to-ZERO signal ratios under certain conditions. The PWD flip-flop forms this pulse and applies it to all 17 of the digit-plane drivers through "OR" inputs.

Figure 39. Magnetic-Core Storage System (Simplified)

5.22 The Cores

The cores are made of General Ceramics material MF-1326B. The first bank contains their core size F-291 which has an outside diameter of 90 mils. A smaller core was used in the second bank; this is F-394, 80 mils in outside diameter. Single-turn switching currents are approximately 950 and 850 milliamperes respectively, and single-turn output voltages (at optimum strobe time) are about 0.1 volt. Switching time, under these conditions, is approximately 1.2 microseconds.

One of the largest problems in the building of a memory of this type is the procurement of large numbers of uniform cores. Core selection was made on the basis of a series of pulse tests, approximately four per core, and resulted in a yield for the first bank of approximately 30 per cent of those shipped to us by the producer. (Yields have been improving materially since this first run.) The selection criterion was fundamentally that of an upper and lower limit on the voltage output from each core when the core was excited by a sequence of current pulses devised to resemble computer operation. Fig. 40 shows typical output-voltage pulse shapes, the nominal limits within which cores were considered acceptable, and the strobe time at which these amplitudes were taken. The horizontal limit lines are at 90 and 120 millivolts, total pulse length is about 1.2 microseconds, and the vertical line showing the strobe time is about 0.5 microsecond from the start of the pulse.

Figure 40.   Test Core Outputs

5.23 Basic-Circuit Types

The read and write currents for the selection planes of the storage are supplied directly from vacuum-tube plates. A single type-6080 vacuum tube, with its sections paralleled, is used to drive a given selection plane in the read direction. Another such tube drives the same plane in the write direction. The control grids of the 6080's are driven through 6BL7 amplifiers from the crystal-matrix output lines. The cathodes of all of the 6080 tubes in the x-read group are connected together, then through a large resistor to a negative-voltage supply. The cathodes of the three other groups of 6080's (x-write, y-read, y-write) are all connected in a similar manner. Each group of cathodes is normally held at a relatively high potential by a power amplifier and is allowed to drop at the proper time. Thus, each 6080 acts not only as a cathode follower but as the logical "AND" gate shown separately in Fig. 39. The large amount

of degeneration caused by the high common-cathode resistor compensates for nonuniformity and aging changes in the characteristics of the tubes. As a result, selection-plane currents remain within very close limits (plus or minus 2 or 3 per cent).

The digit-plane driver consists of a 6080 dual triode driven from two amplifier stages and incorporating sufficient negative feedback from the output to the input to keep the current amplitude within plus or minus 3 per cent over expected tube, component, and power-supply variations.

The output signal from the sensing, or read-out, winding is linearly amplifier from the 100-millivolt level up to approximately a 30-volt level in a single-sided, a-c coupled, wide-band feedback amplifier. The signal is then rectified and applied to the suppressor grid of a 7AK7 gate tube on a bias level of about 30 volts. The control grid of the gate tube is pulsed with a 0.1-microsecond pulse at the optimum moment so that a "standard" Whirlwind pulse issues from the gate to indicate when a ONE is being read.

5.24 Layout and Packaging

A finished storage plane is shown in Fig. 41. The frame's outside dimensions are approximately 9.5 by 9.5 inches. All the windings consist of 32-gauge magnet wire with quadruple-Formex insulation. Fig. 42 shows the cores and wires in some detail. The x and y pairs run vertically and horizontally, the sense winding runs along the diagonals, and the digit-plane

Figure 41.   32-by-32 Plane

Figure 42.   Closeup of 32-by-32 Plane

winding runs horizontally (but is obscured in the shadow around the y pairs). Wiring time for one plane was about one man-week, including an intermediate test and final inspection. The intermediate test was performed when all the cores and x and y wires were in place but before the digit-plane and sense windings were installed; core replacement is relatively easy at this point. The test consisted of applying a sequence of current pulses to a given x line and observing the response of each of the 32 cores on that line by manually stepping the observing-scope probe from one y line to the next. This test was repeated for subsequent x lines, until the 1024 cores were completed. Cores which displayed abnormally high or low outputs were marked for replacement. About 1 core per plane was replaced.

The 17 finished planes were mounted in a stack or array as shown in Fig. 43. Plane-to-plane connections are made by means of the vertical busses soldered into the slotted lugs. Digit-plane and sense-winding connections were made from the same corner of each plane to a mounting board of connectors for coaxial connection to another rack. Selection-plane-driving connections fan out horizontally at the top and bottom of the array. It takes 3 to 4 hours to replace either the entire array or any single plane.

Figure 43. Array of Assembled Planes Mounted in Stall

Fig. 43 includes a view of part of the four-posted stall, or rack, in which the array is mounted. The Whirlwind I magnetic-core storage is shown in Fig. 44. Selection-plane-driver panels are mounted on the four faces of each stall with tubes pointing outward. Visible in each stall above and below the selection-plane-driver panels are the two crystal-matrix switches. The general arrangement is such that temperature-sensitive components, such as cores and crystal diodes, are inside the stall, and large heat-dissipating components, such as tubes, are on the outside.

The sense amplifiers and digit-plane drivers are in plug-in chassis stacked in vertical racks next to the stalls.

5.3 Tests and Performance

Ultimate judgment on the reliability of this particular core storage must rest on its performance over the next year or two. Tentative evaluation may be made, however, from observations of performance during the 4 months that one bank operated in the Digital Computer Laboratory's Memory Test Computer and the 3 months of 2-bank operation in Whirlwind I. In addition, much may be determined from the results of tests made on the core storage to ascertain its tolerance to variations in the parameters significant to its operation.

5.31 Parameter Variations

Many conditions, or parameters, affect the operation of a core memory; driving currents (x, y, read, write, inhibit, and disturb), sense-amplifier gains, strobe time, ambient temperature, memory-information pattern, and repetition rate are good examples. These parameters are not all equally significant or equally easy to manipulate, and so some of them have, as yet, been examined in only a cursory manner. Because sense-amplifier gains have a simple, nearly linear, effect on operation they were adjusted and held at one setting during the tests. Ambient temperature is expected to be held within close tolerances in any operating machine, and a fair amount of information is available on the subject from the core-testing work; temperature was not controlled during the tests but recorded readings were kept. Memory-information pattern and repetition rate were controllable to some degree by the program being run; a program which seemed to give the most adverse pattern and rate was designed and used during most of the testing.

The tests were made on the Memory Test Computer, a high-speed, 16-digit, parallel machine of the Whirlwind type. The machine has a parity checking system which computes whether each 16-digit word to be stored contains an odd or an even number of ONE's, stores the result of this "parity count" in the 17th digit, recomputes the count when the word is read out, and rings an alarm if the result does not check with the contents of the 17th digit. Although major reliance was placed on parity checking for detecting storage malfunction, there was also some programmed identity checking used.

The bias bounds of the sense gates' suppressor grids were chosen as a very convenient measure of the quality of the storage output. The upper bound (least bias) is the point at which

Figure 44.   Core Storage in WWI

errors occur because the gate is mistaking the largest ZERO output for a ONE; at the lower bound (most bias) errors occur because the gate mistakes the smallest ONE output for a ZERO. The bias difference, in volts, is a direct measure of the voltage difference at strobe time between the smallest ONE and the largest ZERO.

Fig. 45 shows the bias bounds for all 17 sense gates as a function of the selection-plane driving-current amplitudes (x, y, read, and write). The program used was the so-called "inchworm" in which 16 words of instructions "bootstrap" themselves around the 1024 registers of the memory. The ambient temperature was recorded at approximately 88 degrees Fahrenheit, about 15 degrees higher than what is now believed to be optimum. Two curves are shown, one for digit-plane currents set at 400 milliamperes and the other at 450 milliamperes. The enclosed areas indicate how much the safe operating point of the storage bank may wander; recent circuit and adjustment improvements have enlarged these enclosed areas somewhat.

Figure 45. Bias Bounds versus Drive Currents

Fig. 46 shows the bias bounds as a function of the timing of the strobe pulse. Time is measured, on this graph, from the instant the read flip-flop is pulsed by the start read pulse. The three curves are for the three values of selection-plane driving current, two extremes and one near optimum. A wide operating region is again indicated.

Figure 46. Bias Bounds versus Strobe Time

5.32 Computer Operation

The first bank of core storage has been in use in Whirlwind since mid-August 1953, the second since 5 September. There has been a steady improvement in their operation as the installation, which was an extremely hurried one, has been gradually cleaned up and made permanent and, also, as the process of debugging these relatively new equipments proceeded. The two banks have not quite been brought to an equal degree of reliability; this may be due, in part, to the fact that the cores in the first bank were not selected as carefully as those in the second so that output ONE/ZERO ratios are not as large. The demands on the Whirlwind computer are heavy, and only a few hours a month are available for further development work on its storage.

Parity alarms occurred, at first, about 3 or 4 times per week; at this writing (November 27, 1953) there has not been a parity alarm for four weeks. This comes to about 460 hours of useful operation or, assuming a 30-microsecond average order time and 2 accesses per average order, it comes to slightly over 100 billion word accesses with each access parity checked and no error detected.

The exact nature of the errors which do occur is, as yet, not known. It is hoped that further work on the system will shed more light on the problem as well as reduce the error rate yet further.

5.4 Storage Selection (MS or TS)

When the 11-digit address section of an instruction contains a number greater than 31 (11111), MS must be used. A 1 in any digit position before the last five (i.e., in any digit position 5-10) will cause the Storage Selection Mixer (SSM) and the Storage Selection Control (SSC) automatically to select MS rather than TS. The SSM (Fig. 47) mixes the outputs of the read-in gate tubes of the Memory Address Register in positions 5-10. If any of these gates is on (that is, if a 1 is in any of those positions) then an output will get through the SSM as a "Select MS" pulse.

Figure 47. Storage Selection Mixer

This "Select MS" pulse is then fed to Storage Selection Control (Fig. 48), a flip-flop, which operates as follows:

COMMANDS FROM CPO UNITS (CALLED FOR BY THE PARTICULAR OPERATIONS INVOLVED SUCH AS CA, SU, ETC.

MS READ — MS READ
GT
MS WRITE — MS STORE
TO MS CONTROL

TS   MS   MS START
0   1
FF
TO CP CONTROL

SELECT MS
(FROM SSM)

Figure 48. Storage Selection Control

Any operation involving Storage must require Operation Control to send out all the commands necessary to operate both TS and MS because the storage-determining address is unknown to Operation Control. Then if MS has been selected because the address is larger than 31, or TS has been selected because the address is equal to or less than 31, one or the other set of commands is canceled by Storage Selection Control (SSC) which allows only the proper set of commands to pass. When MS has been selected, a "Select MS" pulse will be generated from SSM; this pulse will then set the Storage Selection Control (SSC) FF to a 1, and allow the commands associated with MS to pass through the GT: MSC Start (to CPC), and MSC Read or MSC Store (to MSC).

5.5 Transfer of Control

Because reading or writing in MSC takes 9 microseconds, the time pulses through the TPD controlling other computer operations must be stopped during MS operations. Once MS has been selected, computer control is transferred from Central Control to MS Control. The following diagram (Fig. 49) is abstracted from Fig. 60, Clock Pulse Control (CPC).

I MC
GT
.03
LFTP (TO TPD) (GATED OFF)
MS ON   MS OFF
1   0
FF

MS START
(FROM SSC)
MS END CARRY
(MS CYCLE COMPLETED)

NOTE THAT ANY

GATES OFF GT

Figure 49. MS Section of Clock Pulse Control

Control is transferred to MSC in the following manner: The "MSC Start" pulse, passed by SSC as explained above, sets FF.01 to 1 which turns off GT.03 so that it no longer passes 1-mc pulses to the TPD, effectively stopping the clock in Central Control.

1. Papian, William N., "A Coincident-Current Magnetic Memory Cell for the Storage of Digital Information," Proc. I.R.E., vol. 40, pp. 475-478, April 1952. (Also "A Coincident-Current Magnetic Memory Unit," Master's thesis, E.E. Dept., MIT, August 1950.)

2. Harvard University Computation Laboratory, "Investigations for Design of Digital Calculating Machinery," Progress Reports, 2-6 (particularly No. 2), August 1948 - November 1949.

3. Forrester, Jay W., "Digital Information Storage in Three Dimensions Using Magnetic Cores," Jour. Appl. Phys., vol. 22, pp. 44-48, January 1951.

4. Rajchman, Jan, "Static Magnetic Matrix Memory and Switching Circuits," RCA Review, vol. XIII, pp. 183-201, June 1952.

6. PROGRAM TIMING AND OPERATION TIMING

From the continuous string of time pulses fed into it, the time-pulse distributor (Section 2. 223) supplies consecutive time pulses on eight consecutive output lines. Each pulse initiates certain commands which control the functioning of particular pieces of equipment. The timing diagrams included in this chapter indicate the commands required by the representative operations discussed in Chapter 3. References to commands which occur at a fraction of a time pulse (TP 2-1/2, 6-1/2, etc.) mean that these commands are initiated by the basic time pulse, delayed by the fraction of a microsecond indicated. Each command is emitted through a control-pulse-output (CPO) unit in the operation-control matrix.

6. 1 Definition of Terms

Program timing (PT) is the set of commands which takes an instruction from storage and sets up the control switch and storage switch (and step counter) so that the computer, upon receiving further commands, can carry out this instruction. The commands of program timing are the same for every instruction; they always begin at time pulse 1 and extend through time pulse 7. For this reason, program timing has been given only once in the timing diagrams of representative operations.

Operation timing (OT) is the set of commands used to execute the instruction which has been set up by program timing. Operation timing begins at time pulse 6 of the same TPD cycle in which setup of that instruction occurs. The commands used in operation timing vary with the nature of each operation.

6. 2 Overlap of Program Timing and Operation Timing

Because the same pieces of equipment can be used for program timing and operation timing if the commands to this equipment come at different time pulses (or fraction thereof), it is possible to overlap program timing and operation timing -- thus saving considerable computation time. Figure 63 shows this overlap as the instructions in the program are carried out.

In Figure 63 operation timing for operation 1 overlaps program timing for instruction 1 during time pulses 6 and 7. (This is also the case for operation 2 and instruction 2, etc.) Overlap is possible because the PT commands at these time pulses are for transfer checks and adding 1 to the program counter, and because these latter facilities are not needed by most operations timing at those times.

While operation 1 is being performed, operation 2 is being set up from instruction 2 of the program. This overlap from time pulse 1 through time pulse 4 means that the two sets of commands are either operating separate pieces of equipment or are operating the same pieces on a time-sharing basis.

Time Pulse

Commands

1   — Commands to set up Operation 1
2   from Instruction 1
3
4   PT
5
6
7   — Commands to carry out
8   Operation 1
1   OT
2
3
4   PT
5   — Commands to set up Operation 2
6   from Instruction 2
7   OT
8   — Commands to carry out
       Operation 2
1   PT
2   — Commands to set up Operation 3
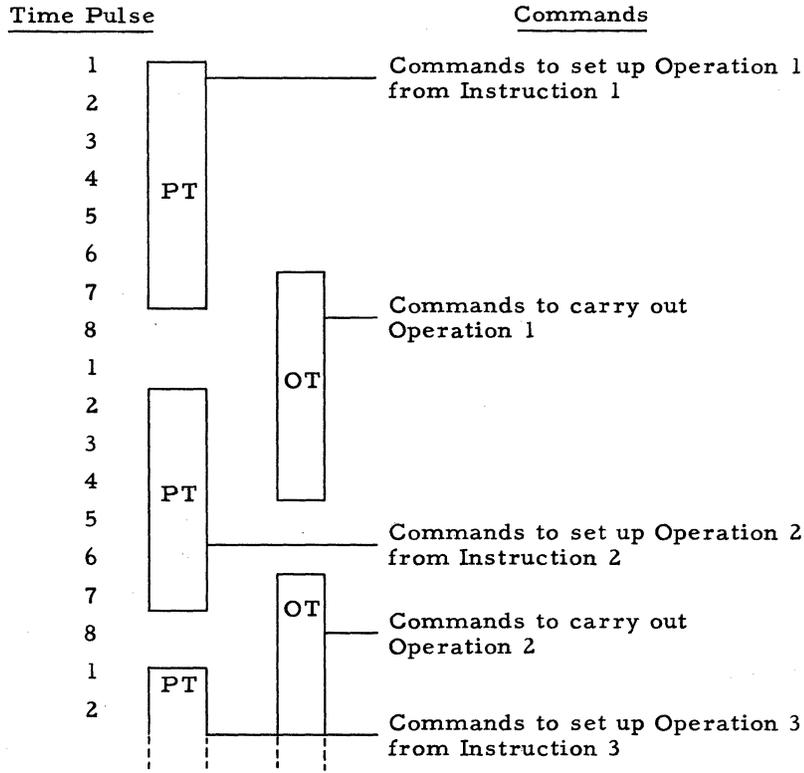   from Instruction 3

Figure 50. Overlap of Program Timing and Operation Timing

6.3 Timing Diagram for Representative Operations (Figure 51)

The attached diagram gives the necessary commands for program timing and for opera-
tion timing of the representative operations covered in Chapter 3. Timing of other operations
may be obtained from the latest tables compiled by the Block Diagrams Group.

PROGRAM TIMING

TP	COMMAND	CPO
1.	* SS Clear (TSS 0.5 μsec later)	19 or 82
2.	PC Rd Out	12
	SS Rd In	17
	*' CR Clear ('PC Rd to CB 0.5 μsec later)	09
3.	' SS Rd Out, CR Rd In	18
	MS Rd, PAR Clear	81
4.	Trfr Check	56
	SC Preset (SRC 0.25 μsec later)	03
	Stor Rd Out) (PAR also)	76
	PAR Rd In)	
	Stor Rd to CB	74
	CS Clear	68
	SS Clear	20
5.	SS Rd In	17
	PAR Rd Out	28
	CS Rd In	66
	SC & SRC Rd In	02
6.	' Control Matrix to Bus	116 - 120
	' SS Rd Out, CR Rd In	49
7.	' Trfr Check	56
	Add to PC	14

TIMING OF REPRESENTATIVE OPERATIONS

Figure 51

Note: the ad operation is identical with ca except that "AC Carry Clear, CPO 55" replaces "AC & SRC Clear" and "BR Clear" at time pulse 6; "Special Carry" is not required at time pulse 7. The su operation is identical with ad except that "Subtract, CPO 80" replaces "Add" at time pulse 2.

Legend: * quasi Program Timing commands
' Transfer-checking commands

OPERATION TIMING

TP	Clear and Add		Multiply and Round Off		Divide		Shift Right		Transfer to Storage	
	COMMAND	CPO	COMMAND	CPO	COMMAND	CPO	COMMAND	CPO	COMMAND	CPO
6.	AR Clear	70	AR Clear	70	AR Clear	70	AC Carry Clear	55		
	AC & SRC Clear	41	AC Carry Clear	55	AC Carry Clear	55	AC Sign Check	23		
	BR Clear	53	BR Clear	53	MS Rd, PAR Clear	84				
	MS Rd, PAR Clear	84	MS Rd, PR Clear	84	AC Sign Check	23				
			AC Sign Check	23						
7.	Special Carry	26	Stor & PAR Rd Out	64	Stor & PAR Rd Out	64	Add to SC	04		
	Stor & PAR Rd Out	64	'' Stor & PAR Rd to CB	65	' Stor & PAR Rd to CB	65				
	' AR Rd In	79	AR Rd In	79	AR Rd In	79				
	' Stor & PAR Rd to CB	65	AC Rd to BR	59						
8.	' AR Rd Out	69	' AR Rd Out	69	' AR Rd Out	69			Stor Clear	61
	' CR Rd In	58	' CR Rd In	58	' CR Rd In	58			PAR Clear	88
1.	' Trfr Check	47	' Trfr Check	47	' Trfr Check	47	* SS Clear (TSS 0.5 μsec later)	19	AC Rd Out	43
	* SS Clear (TSS 0.5 μsec later)	19	* SS Clear (TSS 0.5 μsec later)	19	* SS Clear (TSS 0.5 μsec later)	19			Stor Rd In)	
			AC & SRC Clear	42	AR Sign Check	22			PAR Rd In)	71
			AR Sign Check	22					MS Write, SS Clear	82
2.	Add	78	*' CR Clear ('PC Rd to CB 0.5 μsec later)	09	*' CR Clear ('PC Rd to CB 0.5 μsec later)	09	Stop Clock	48	*' CR Clear ('PC Rd to CB 0.5 μsec later)	09
	*' CR Clear ('PC Rd to CB 0.5 μsec later)	09	Stop Clock	48	Stop Clock	48	Add to SC	04		
			Multiply	35	Subtract	80	Shift Right	31		
					Divide	36	*' CR Clear ('PC Rd to CB 0.5 μsec later)	09		
3.	Carry	46	Carry	46	AC Clear	51	Round Off (via SRC)	52		
			Round Off (via SRC)	52						
4.	Arith Check	32	BR Clear (via SRC)	44	AC Carry Clear	15	Arith Check	32		
			Product Sign 0.5 μsec later	24	Product Sign 0.5 μsec later	24	PR Clear (via SRC)	44		
							Product Sign 0.5 μsec later	24		
5.										

APPENDIX

WWI BLOCK DIAGRAMS

This chapter contains block diagrams of central control, test storage, the arithmetic element, the check register, and the MS system. In many instances these diagrams have been somewhat simplified for the better understanding of certain elements.

For greater ease and exactness in reference, the various elements of WWI have been arbitrarily assigned system numbers. The system number of the arithmetic element, for example, is 300; all subordinate elements of AE are assigned numbers in the 300 block (e. g., the accumulator is 302). In the same manner, each flip-flop and gate tube associated with such a subordinate element is identified by the number of that element and a number of its own expressed as a decimal fraction   Thus "FF 302.02" identifies a flip-flop in the accumulator. (The number by itself, without the "FF" prefix, may designate either a flip-flop or a gate tube, as they may carry the same numbers, although otherwise unrelated. It should also be noted that FF's performing identical functions within the same subordinate element will bear identical numbers; the same is true of GT's.) A list of system numbers is given below.

100 CENTRAL CONTROL (CC)

- 101 Pulse Generator (PG)
- 102 Program Counter (PC)
- 104 Control Switch (CS)
- 105 Control Matrix (CM)
- 106 Time-Pulse Distributor (TPD)
- 109 Clock Pulse Control (CPC)
- 110 Frequency Divider (FDV)
- 111 Synchronizer (SYN)

200 TEST STORAGE

- 201 Test Storage Switch (TSS)
- 202 Toggle-Switch Storage (TG)
- 203 Flip-Flop Storage (FFS)

300 ARITHMETIC ELEMENT (AE)

- 301 A-Register (AR)
- 302 Accumulator (AC)
- 303 B-Register (BR)
- 304 Sign Control
- 305 Step Counter (SC)
- 306 Multiply Control
- 307 Shift Control
- 308 Divide Control
- 309 Overflow Control

400 INPUT-OUTPUT ELEMENT (IOE)

- 403 In-Out Register (IOR)
- 404 In-Out Delay Counters
- 410 In-Out Control (IOC)
- 420 In-Out Switch (IOS)

500 GENERAL UNITS

- 510 Display Scopes & Control
- 513 Vertical Decoder
- 514 Horizontal Decoder
- 520 Magnetic Tape
- 530 Paper Tape
- 540 Camera

600 CHECKING CIRCUITS

- 601 Check Register (CR)
- 602 Alarm Indicator
- 603 Alarm Control (for ck)

700 CONSOLE

800 MAGNETIC-CORE STORAGE

- 842 Parity Register

The WWI system has been simplified in order to show the major units and their pertinent interconnections. The multiple command cables, not named on the drawing, are itemized according to the system numbers of the units they connect. (For identification of units by system number, see the list at the beginning of this Section.)

CABLES BETWEEN MAJOR UNITS

100 to 200 (5 lines)

   105 to 201 - TSS Read In
                   TSS Read Out
   105 to 202/3 - TS Read Out
                   TS Read to Check Bus
   105 to 203 - TS Read In

100 to 300 (28 + 2 lines)

   105 to 301 - AR Clear
               AR Read In
               AR Read Out
               Add
               Subtract
   105 to 302 - AC Clear
               AC Carry Clear
               AC Read Out
               BR Clear via SRC
               AC Read to BR
               End-Around Carry
               Compare
   105 to 302/9 - Carry
   105 to 303 - BR Clear
               Round Off
   105 to 304 - Check Magnitude
               AR Sign Check
               Product Sign
   105 to 304/8 - AC Sign Check
   105 to 305 - SC Preset
               SC Read In
               SC Read Out
               Add to SC
   105 to 305/6 - Multiply

105 to 307 - Shift Left

               Shift Right
   105 to 305/8 - Divide
   105 to 309 - Arithmetic Check
   109 to 308 - Low-Frequency Clock Pulses
   109 to 306/7/10 - High-Frequency Clock
                     Pulses

100 to 400 (3 lines)*

Select External Unit and Mode of Operation
Set up for Reading
Set up for Recording

100 to 600 (2 lines)

   105/8 to 601 - CR Read In
                 Transfer Check

400 to External Unit (4 lines)*

Stop
Start
Read
Record

700 to 100 (8 lines)

   700 to 111 - Restart from Stop
               Clear All Registers
               Start Over from Stop
               Single Pulses
               Reset FF
               Change to PB (Stop)
               Instruction by Instruction
               Read In the Program

* These commands are simply a functional description of the many commands actually required.

SIMPLIFIED SYSTEM BLOCK DIAGRAM, WWI

Figure 53 shows the major WWI elements and the gate tubes through which they are connected to the main bus and the check bus.

The bus system is complicated by transfer checking: all elements which are required only to receive data from the main bus must also be able, for checking reasons, to transmit this information back to the main bus. For example, the switches (104, 201, 420) are forced to read information back to the bus although their function in the system demands only that they receive information.

Those elements whose function is both to receive information from and to transmit information to the main bus must also be able, for checking reasons, to transmit to the check bus. An example is the Program Counter, which has three sets of gate tubes -- one for receiving information from the main bus, one for transmitting to the main bus, and one for transmitting to the check bus.

The principle of transfer checking is described briefly in Chapter 4.

BUS CONNECTIONS, WWI

CENTRAL CONTROL, WWI (CONDENSED) -- FIGURE 54

Central Control, corresponding to the operator of a manual system, calls for each instruction in sequence (directed by the Program Counter), extracts it from Storage, holds it temporarily in the Program Register, and then distributes the operation-code section to the Control Switch and the address section to the Storage Switch. The operation-code section sets the Control Switch to select the appropriate operation line to the Control Matrix; the Control Matrix then provides the proper commands for the operation. Timing of these commands is directed by the Time-Pulse Distributor of the Master Clock.

The Storage Selection Control chooses the correct set of commands for the storage (MS or TS) being used.

FROM CONTROL MATRIX
- PC READ OUT
- PC READ TO CHECK BUS
- PC READ IN
- ADD TO PC
- PC CLEAR

102 PROGRAM COUNTER

FROM CONTROL MATRIX
- PAR READ OUT
- PAR READ IN
- PAR CLEAR
- PAR READ TO CHECK BUS

842 PARITY REGISTER

"O" TO MS INPUT

"1" FROM MS OUTPUT

MAIN BUS

CHECK BUS

MAIN BUS

CHECK BUS

FROM CONTROL MATRIX
- CHANGE FROM cp TO sp (FROM AC)
- CS READ IN
- CS CLEAR

104 CONTROL SWITCH

CS OUTPUT LINES

32

(OPERATIONS)

105 CONTROL MATRIX

112

CONTROL PULSES (COMMANDS)

- ALARM
- IOC END CARRY
- STOP CLOCK (FROM CONTROL MATRIX,IOC)
- STEP-COUNTER END CARRY
- MSC END CARRY
- MSC START (FROM SSC,108)

101,106 109,110 MASTER CLOCK

TIME PULSES

8

2

108 STORAGE SELECTION CONTROL

SELECT MS FROM MAR

PUSHBUTTON SIGNALS FROM OPERATOR'S CONSOLE
- RESTART
- SINGLE PULSES
- CHANGE TO PUSHBUTTON (STOP)
- CLEAR
- RESET
- START OVER
- READ IN
- ORDER BY ORDER
- EXAMINE
- COMPLEMENT

HFCP
LFCP
LFSP

TO AE

TO IN-OUT CONTROL

CLEAR
RESET

TO ALL FF'S
TO FF STORAGE, SC.CS,
(TPD).TSS,PC,CR

CENTRAL CONTROL, WWI (CONDENSED)

The basic source of all pulses in the computer is the Pulse Generator, an element which consists of a 2-mc oscillator for producing pulses at a 2-mc frequency, and a 2:1 frequency divider for producing pulses at a 1-mc frequency. These 2-mc and 1-mc pulses are then fed into Clock Pulse Control, an element which, according to the control pulses also fed into it, determines the distribution of the original 2-mc and 1-mc pulses to various parts of the computer. An important element fed from Clock Pulse Control is the Time-Pulse Distributor which directs the timing of the various commands used in performing an operation.

The Synchronizer takes asynchronous commands or control pulses from various sections of the computer and synchronizes them with a major frequency before they are permitted to go to CPC.

2-MC
OSCILLATOR

2:1
FREQUENCY
DIVIDER

101
PULSE GENERATOR

2MC

1MC

109
CLOCK
PULSE
CONTROL

SYNC.

STARTOVER
RESTART
ETC.

CONTROL
PULSES

HIGH-
FREQUENCY
CLOCK
PULSES

LOW-
FREQUENCY
CLOCK
PULSES

TO
ARITHMETIC
ELEMENT

LOW-
FREQUENCY
TIME
PULSES

TO TIME-PULSE
DISTRIBUTOR

GENERALIZED BLOCK DIAGRAM
MASTER CLOCK, WWI

THE PROGRAM COUNTER -- FIGURE 56

Three sections of the counter are shown. The counting is done by the FF's .01. The add pulse comes in from the right. If the first FF is a 0, it will be switched to a 1, increasing the binary number contained in PC by 1. If the first FF is a 1, GT .05 will be on and the add pulse can pass through to add into the second FF. The first FF will be reset to 0, after a delay, by the same add pulse. The system is very similar to that used for the Accumulator high-speed carry.

The GT's .01 are used for reading in a new number from the bus in a subprogram operation. Prior to the reading in, the counter must be cleared or reset to 0 by a pulse on the clear line.

It is also possible to change the contents of PC by pulsing the reset line. Toggle switches are provided so that the counter may be set to the address of any desired "first" instruction prior to starting a calculation.

The Program Counter must also be able to read out to the bus, via GT .02, and into the check bus via GT .03. When PC is full, it will put out an end-carry pulse which may be used for resetting the counter.

PROGRAM COUNTER

C.B.

THE PARITY REGISTER (SIMPLIFIED) -- FIGURE 57

Two sections of the register are shown. Digits are read in from the bus via GT's .01 and stored in FF's .01. GT's .02 and GT's .03 are provided for reading out to the main bus and check bus respectively. The clear line is pulsed prior to each read-in, in order to clear the register. A "0" output from PAR will cause a "0" to be written in magnetic-core storage. The parity check circuits are considered part of the parity register but have been omitted in this simplified drawing.

PARITY REGISTER (SIMPLIFIED)

CONTROL SWITCH AND TEST-STORAGE SWITCH -- FIGURE 58

This same switch, with minor modifications as noted, is used in both Operation Control and Test Storage. Five FF's .01 are used to store the five-digit binary number describing the desired output line. GT's .01 are used for reading in the operation-code section of the instruction from the bus. GT's .02 are used for sending this information back to the bus for check of the transfer if this switch is the Test Storage Switch; the transfer of information to the Control Switch is checked for proper line selection through read-out GT's .116-.120 of the Control Matrix.

The FF's drive the crystal matrix. A positive signal will appear on one of the output lines if all the crystals in that line are on.

The purpose of the change-from-cp-to-sp line has been described in Section 2.13. The binary codes for the different operations have been established so that the cp operation has the same code as sp, except that the fifth digit of cp is a 0 instead of a 1. The change-to-sp pulse changes FF .01 of CS-4 to a 1. The control switch which had been set to cp (01110) is then reset to sp (01111).

OUTPUT LINES
(OPERATIONS)

0	00000
1	00001
2	00010
3	00011
4	00100
5	00101

TO
CONTROL
MATRIX

26	11010
27	110,11
28	11100
29	11101
30	11110
31	11111

READ OUT

(TEST - STORAGE
SWITCH ONLY)

GT .02   GT .02   GT .02   GT .02   GT .02

FF .01   FF .01   FF .01   FF .01   FF .01

CHANGE
CONTROL (CONTROL SWITCH ONLY)
FROM CP
TO SP

RESET

CLEAR

GT .01   GT .01   GT .01   GT .01   GT .01

READ IN

FROM TO   FROM TO   FROM TO   FROM TO   FROM TO
BUS BUS   BUS BUS   BUS BUS   BUS BUS   BUS BUS

NOTE:
1. OUTPUT LINE IS SELECTED
   IF ALL ITS CRYSTALS ARE
   ON.
2. TEST STORAGE SWITCH IS
   LOGICALLY IDENTICAL, BUT
   IT SELECTS A STORAGE
   REGISTER INSTEAD OF AN
   OPERATION LINE.

C.B.

CONTROL MATRIX -- FIGURE 59

The Control Matrix consists of 32 operation lines from the Control Switch and 8 time-pulse lines from the Time-Pulse Distributor. The outputs from the matrix (commands) originate at the various CPO gate tubes, which represent the intersections of selected operation lines with selected time-pulse lines. To simplify the drawing, only 3 operation lines and the associated CPO units are shown.

Going from one instruction to the next in a program, certain processes are continually repeated. For example: the program counter is indexed; the instruction is located in storage. The commands for all these processes are independent of the particular operation included in the instruction. Moreover, each successive instruction in the program must provide for the commands which are needed to set up the next instruction in the program. These instruction-setup commands are common to most instructions (and operations) and, therefore, can utilize a common operations line. This line is called the Program Timing Line in all Laboratory literature. Although previously this title had meaning, a better name might be "Instruction-Setup Line."

ABSTRACT OF CONTROL MATRIX

CPO'S
(CONTROL-PULSE-OUTPUT UNITS
ARE GATE TUBES THROUGH
WHICH ARE PUT OUT
COMPUTER CONTROL PULSES)

TIME PULSES
FROM TPD

CONTROL SWITCH

BUS
CLEAR
READ IN
CHANGE FROM
OP TO SP

SC & SRC RD IN — 02
SC PRESET — 03
PC RD TO CB — 09
PC RD OUT — 12
ADD TO PC — 14
SS RD IN — 17
SS RD OUT, CR RD IN — 18
SS CLEAR — 19
SS CLEAR — 20
SPECIAL CARRY — 26
PAR RD OUT — 28
ARITH CHECK — 32
PAR RD IN — 39
AC & SRC CLEAR — 41
CARRY — 46
TRFR CHECK — 47
SS RD OUT, CR RD IN — 49
BR CLEAR — 53
AC CARRY CLEAR — 55
TRFR CHECK — 56
CR RD IN — 58
STOR RD OUT — 64
STOR RD TO CB — 65
CS RD IN — 66
CS CLEAR — 68
AR RD OUT — 69
AR CLEAR — 70
STOR RD TO CB — 74
STOR RD OUT — 76
ADD — 78
AR RD IN — 79
SUBTRACT — 80
CLEAR — 81
SS CLEAR — 82
MS RD, PAR CLEAR — 84

CONTROL MATRIX TO BUS
(FOR CHECKING THE
TRANSFER OF AN
INSTRUCTION TO
THE CONTROL SWITCH)

116
TO
120

29 OTHER LINES

79 OTHER LINES

PROGRAM TIMING LINE
(SELLECTS CPO'S TO SEND
OUT COMMAND PULSES
NEEDED TO BRING THE
NEXT INSTRUCTION FROM
THE PROGRAM IN STORAGE)
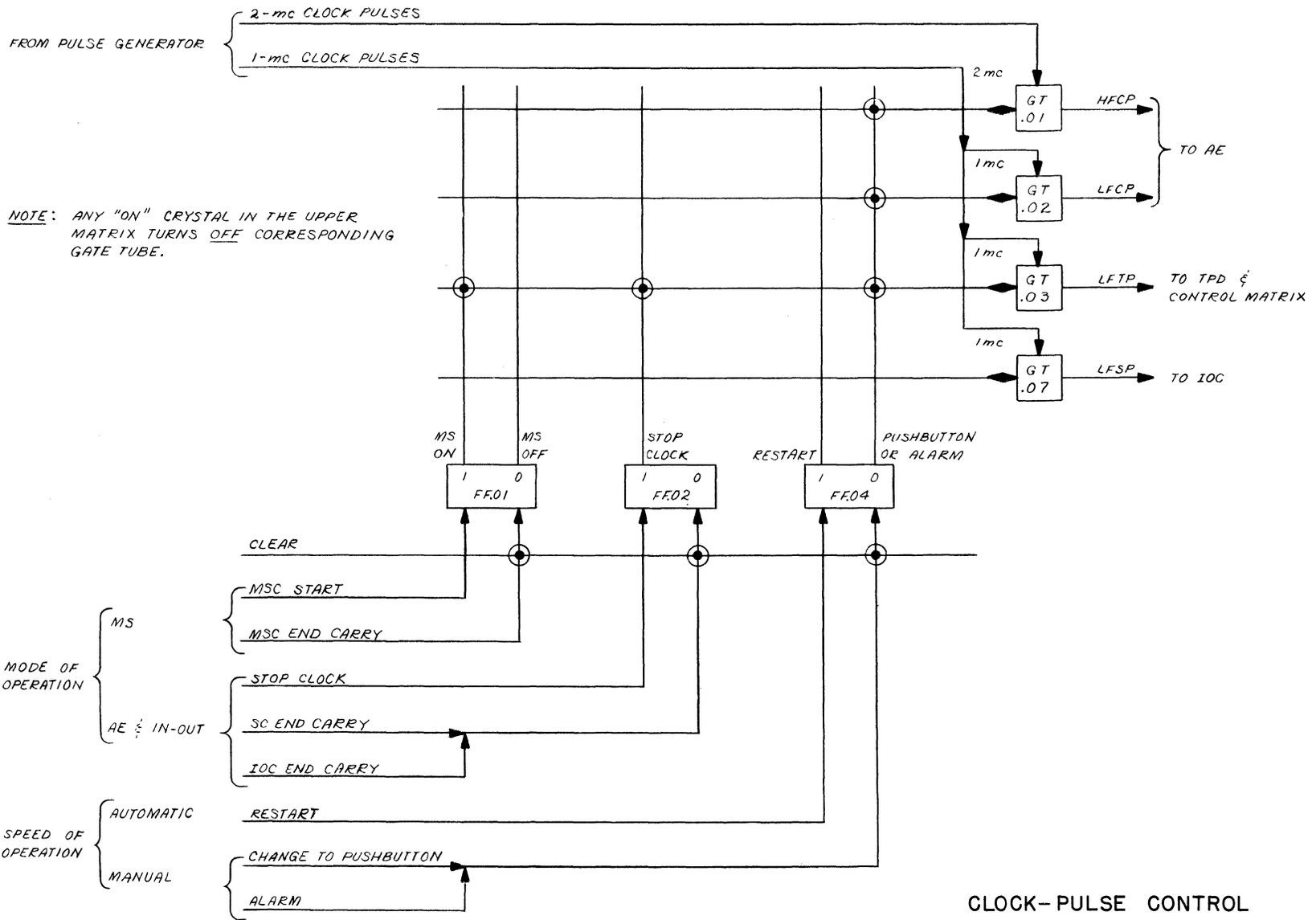
FF .02 in Clock Pulse Control is used for starting and stopping the Time-Pulse Distributor which supplies the timing pulses for the Control Matrix. When FF .02 is set to 0, GT .03 will be on and low-frequency clock pulses will be supplied to the TPD. When FF .02 is set to 1 by a stop-clock control pulse, GT .03 will be off and no further time pulses will be produced by the TPD.

A multiply, divide, or shift pulse sent to the Arithmetic Element (AE) is also sent in on the stop-clock line, setting FF .02 to a 1 and permitting GT's .01 and .02 to pass 1-mc and 2-mc pulses to AE. When the AE has finished its part of the operation, the Step Counter (SC) (which has been counting steps for the operations: multiply, divide, and shift) will produce an end-carry (SC End Carry) which resets FF .02 to 0, restarting the TPD, and shutting off the supply of clock pulses to the AE.

Clock Pulse Control will receive an alarm pulse every time the computer detects an error. The alarm pulse will set FF .04 to 0, stopping the flow of 1-mc pulses to the TPD and the 1-mc and 2-mc pulses to the AE, and keeping the situation as it was at the time of the error. A pulse on the restart line sets FF .04 to a 1, permitting the pulses to the TPD and AE to be restarted after a fault has been corrected or simply after a stop has been made.

FROM PULSE GENERATOR
{
2-mc CLOCK PULSES
1-mc CLOCK PULSES

2 mc

GT .01 → HFCP
GT .02 → LFCP
} TO AE

NOTE: ANY "ON" CRYSTAL IN THE UPPER MATRIX TURNS OFF CORRESPONDING GATE TUBE.

1 mc
1 mc

GT .03 → LFTP → TO TPD & CONTROL MATRIX

1 mc

GT .07 → LFSP → TO IOC

MS ON   MS OFF   STOP CLOCK   RESTART   PUSHBUTTON OR ALARM

| 1 | 0 |
FF.01

| 1 | 0 |
FF.02

| 1 | 0 |
FF.04

CLEAR

MODE OF OPERATION
{
MS
{
MSC START
MSC END CARRY
}
AE & IN-OUT
{
STOP CLOCK
SC END CARRY
IOC END CARRY
}
}

SPEED OF OPERATION
{
AUTOMATIC
RESTART
MANUAL
{
CHANGE TO PUSHBUTTON
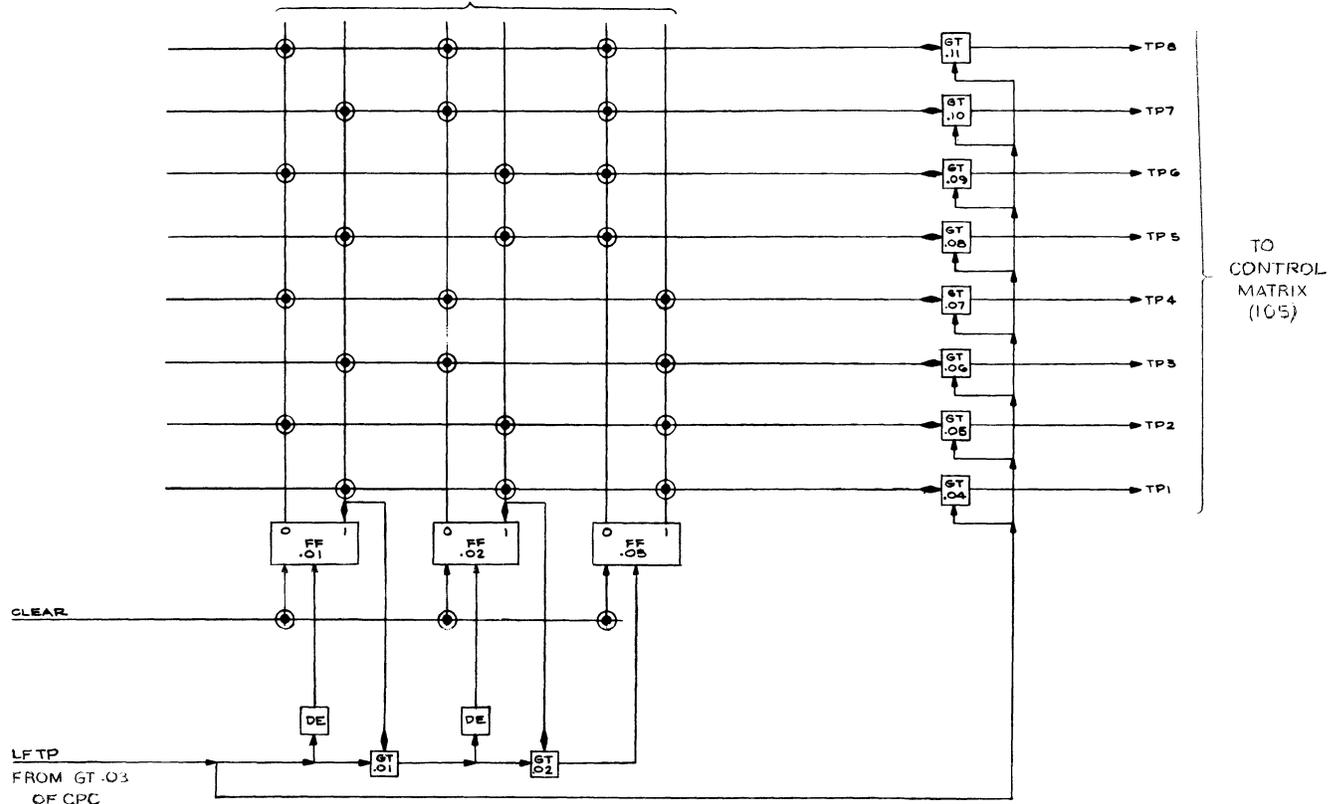ALARM
}
}

CLOCK-PULSE CONTROL

TIME-PULSE DISTRIBUTOR -- FIGURE 61

The Time-Pulse Distributor consists of an 8-way diode matrix switch with the three driving FF's connected in a counter circuit. Each time pulse, as it appears, will change the contents of the counter register to a number greater by 1. The switch output will thus change for each time pulse.

A gate tube is connected to each of the switch outputs. The time pulses coming in are supplied to the grids of all these tubes. The switch will supply a gating voltage to only one tube at a time; the time pulse will thus appear only on the selected output.

The TPD is off during the repetitive parts of operations such as multiplication, division, and shifting, but it can be restarted from Clock Pulse Control at any other time to produce pulses in the normal order.

NOTE: ANY "ON" CRYSTAL IN UPPER MATRIX
TURNS OFF CORRESPONDING GATE TUBE

GT .11	►TP8
GT .10	►TP7
GT .09	►TP6
GT .08	►TP5
GT .07	►TP4
GT .06	►TP3
GT .05	►TP2
GT .04	►TP1

TO
CONTROL
MATRIX
(105)

| 0 FF .01 1 |
| 0 FF .02 1 |
| 0 FF .03 1 |

CLEAR

DE

DE

LFTP
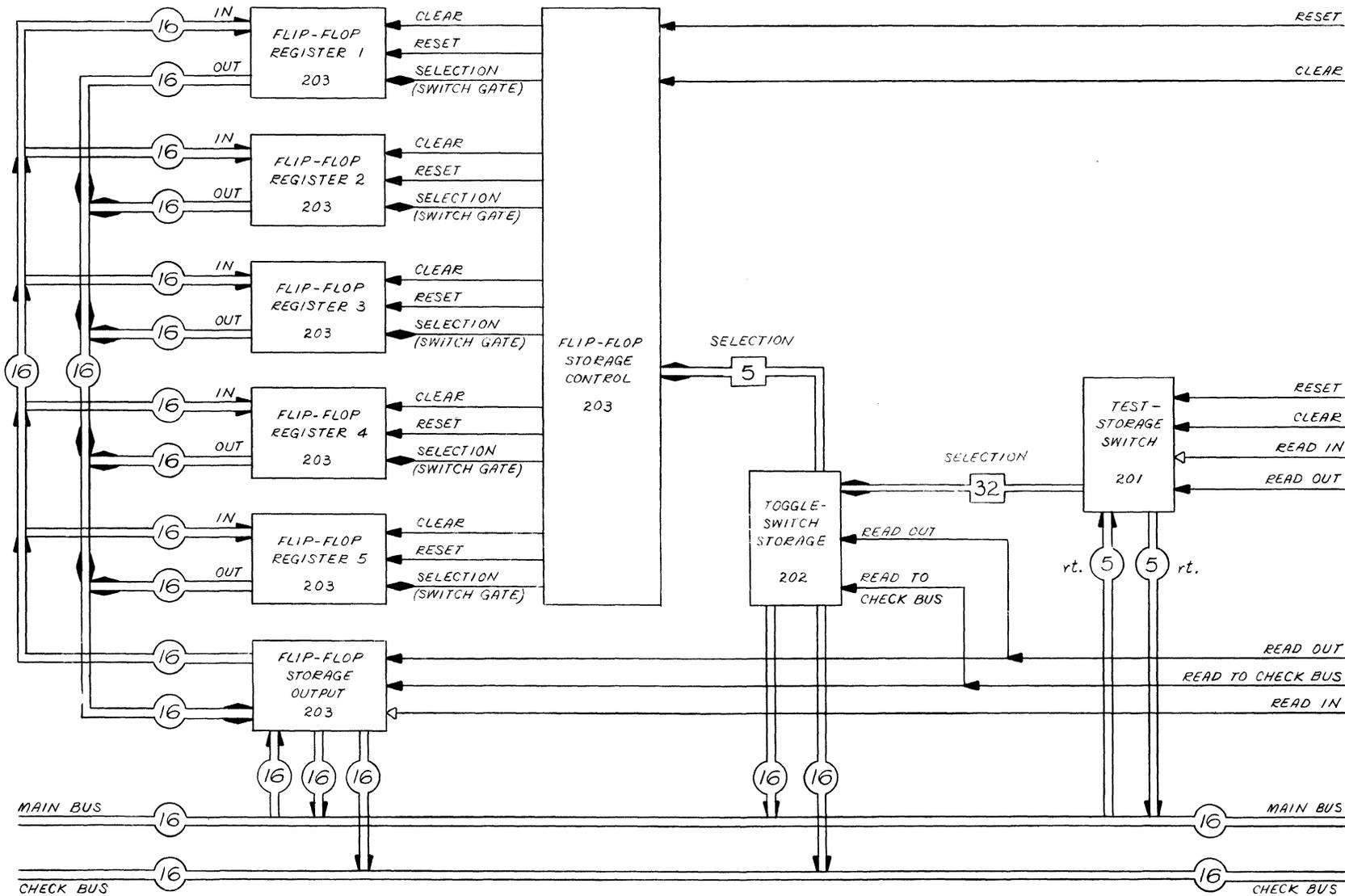FROM GT .03
OF CPC

GT .01

GT .02

TIME PULSE DISTRIBUTOR

C.B.

TEST STORAGE -- FIGURE 62

This drawing shows the arrangement of the elements of test storage. Five flip-flop registers and twenty-seven toggle-switch registers are provided. The desired one of the 32 registers (27 toggle-switch + 5 flip-flop) is selected by the 32-position test-storage switch.

Provision is also made for reading out to the check bus and for reading into flip-flop storage from the main bus. Toggle-switch storage is, of course, fixed manually and cannot be changed by information from the bus.

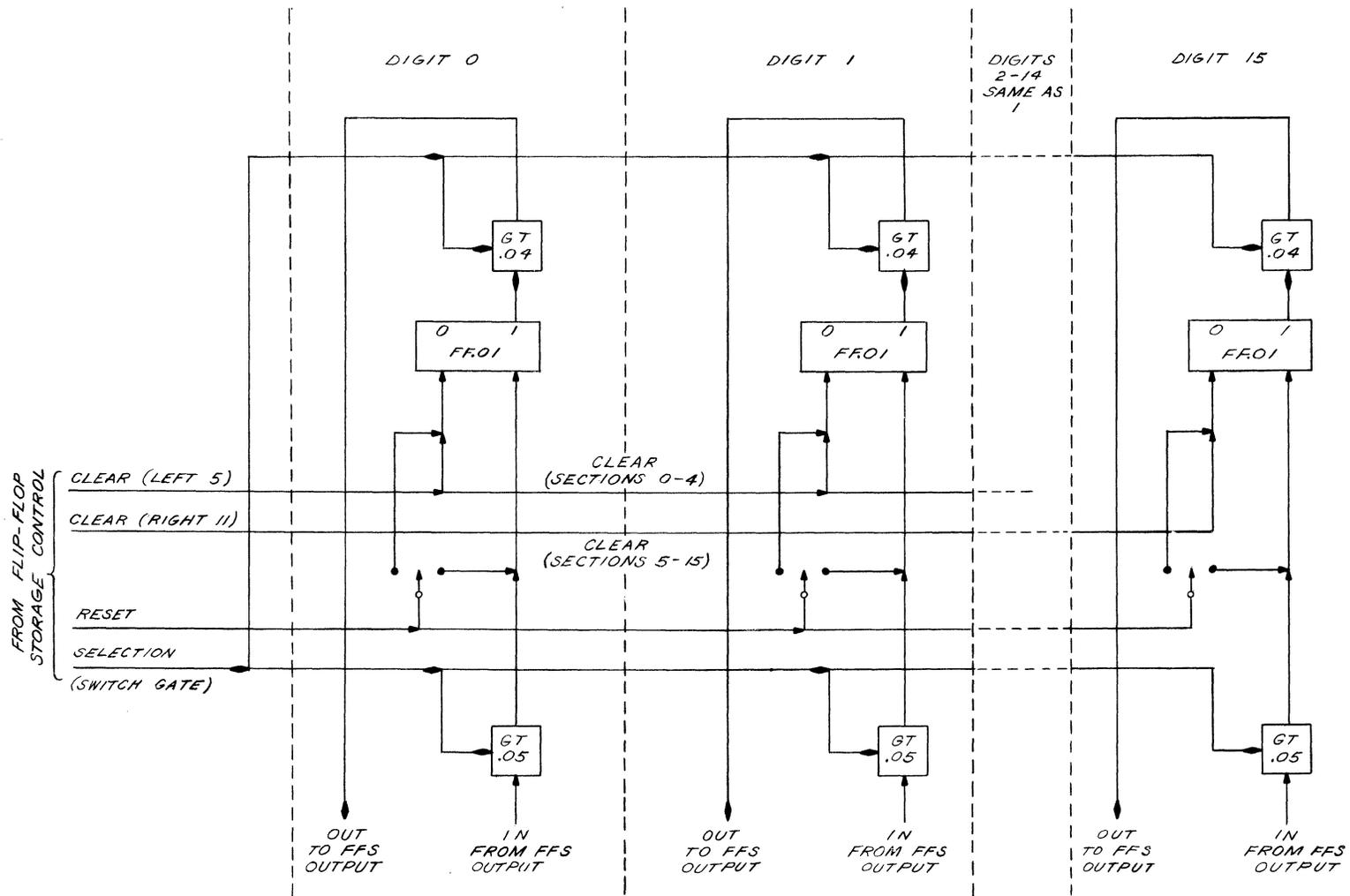TEST STORAGE

FLIP-FLOP STORAGE -- FIGURE 63

A number stored in the Flip-Flop Storage Register is held in the 16 FF's .01.

When a particular flip-flop storage register is selected by the Test Storage Switch a selection-switch gate (from the switch output) is supplied to both GT's .04 and GT's .05 of that register. If a FF .01 in the selected register is set to a 1, a gate will be supplied to the FF-Storage output panel. If a FF .01 is set to 0, no gate will be supplied. This enables the 5 FF registers to share a common output panel.

Before a number is stored, a clear pulse must be sent to the selected register. The selection-switch gate turns on GT's .05 as well as GT's .04. A number coming in the IN line will then be stored in the FF's .01.

The reset line allows insertion of initial values in the flip-flop registers prior to a test problem. The single-pole double-throw toggle-switches are set to the desired number for each register. When the reset line is pulsed, these numbers will be inserted into the registers. The reset line is shared by all 5 flip-flop registers. Thus, all FF registers are set simultaneously, independent of the Test Storage Switch setting.

The IN and OUT lines are common to all registers, the connections to the lines being determined by the selection-switch gate from the Test Storage Switch.

FLIP-FLOP STORAGE

FLIP-FLOP STORAGE OUTPUT -- FIGURE 64

When a number is to be stored in Flip-Flop Storage, a gate is supplied to GT's .01 on the storage-read-in line. Pulses proceeding along the bus are thus sent through GT .01 onto the IN line to Flip-Flop Storage.

For reading out, a gate is supplied from the selected register in flip-flop storage to GT's .02. When the storage-read-out line is pulsed, the contents of the selected register will be transmitted onto the bus. The gate is supplied to GT's .03, as well as to GT's .02, so that the number may be read out onto the check bus by pulsing the storage-read-to-check-bus line.

FLIP - FLOP - STORAGE OUTPUT

FLIP-FLOP STORAGE CONTROL -- FIGURE 65

Flip-Flop Storage Control relays the selection-switch gate from the Test Storage Switch to the selected register of Flip-Flop Storage. The selection-switch gate is also supplied to GT's .07 and .08 which send the clear pulse to the proper register and to the proper digits of that register before a number is stored there.

The storage-reset pulse is sent to all flip-flop registers through the reset lines of FF Storage Control.

TO FF REGISTER 1   TO FF REGISTER 2   TO FF REGISTERS 3 & 4   TO FF REGISTER 5

SAME AS 2

CLEAR (rt. II)   CLEAR (left 5)   RESET   SELECTION (SWITCH GATE)

CLEAR (rt. II)   CLEAR (left 5)   RESET   SELECTION (SWITCH GATE)

CLEAR (rt. II)   CLEAR (left 5)   RESET   SELECTION (SWITCH GATE)

FROM SYN. (III)   RESET

GT .08   GT .07   GT .08   GT .07   GT .08   GT .07

FROM CONTROL MATRIX   CLEAR   CLEAR (rt. II)

SELECTION (SWITCH GATE)   SELECTION (SWITCH GATE)   SELECTION (SWITCH GATE)

NOTE:
   SELECTION LINES COME FROM THE
   OUTPUT OF THE TEST-STORAGE SWITCH (201).

FLIP-FLOP-STORAGE CONTROL

C.B.

A selected output line from the Test Storage Switch will connect the proper toggle-switch register to GT's .02 to transmit information to the main bus and to GT's .03 to transmit to the check bus. A closed toggle switch, representing a 1, will then allow a 1 to pass to the main bus and the check bus. Crystals prevent sneak paths in this circuit.

TOGGLE-SWITCH STORAGE

THE ARITHMETIC ELEMENT -- FIGURE 67

The purposes and functions of the elements shown in this drawing have been discussed in general in Chapter 3.   The detailed drawing is presented for perusal by the brave.   Most details presented here were discussed in the text.
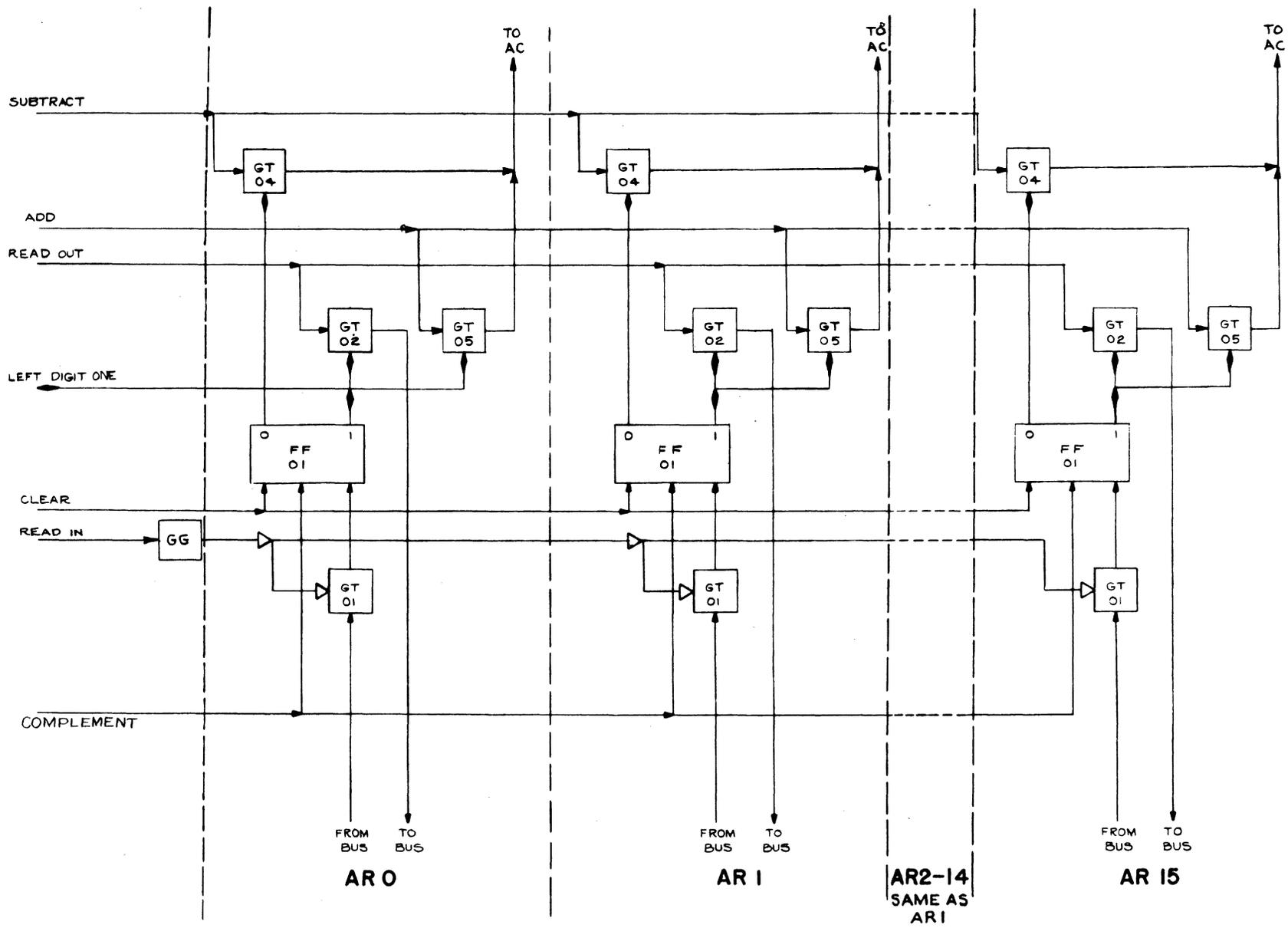
BUS

BUS

AR READ IN
AR READ OUT
AR CLEAR

A-REGISTER 301

16   16

LEFT DIGIT ONE
AR COMPLEMENT

SUBTRACT
ADD

SUBTRACT

ONE FROM BR 15

CHECK MAGNITUDE

SIGN CONTROL 304

GT .06

GT .07

AR SIGN CHECK

AC SIGN CHECK

GT .05

LEFT DIGIT ONE

PRODUCT SIGN

GT .04

O FF .01 1

DE 0.25

DE 0.25

AC COMPLEMENT

SIGN CONTROL CLEAR

COMPARE

CHANGE CONTROL

DIVIDE ERROR ALARM

GT .07

1 FF .03

GT .12

DIVIDE ERROR 308

DIVIDE ZERO

DIVIDE ERROR CLEAR

DIVIDE ONE

ARITH-CHECK ALARM

ARITHMETIC CHECK

OVERFLOW CONTROL 309

CYCLE LEFT ONE TO BR 15

LEFT DIGIT CARRY

TO AC0

END-AROUND CARRY

DIVIDE SHIFT LEFT

CARRY

END-AROUND CARRY

OVERFLOW CLEAR

END-AROUND CARRY
CARRY

END-AROUND CARRY

DIVIDE ONE
DIVIDE ZERO

DIVIDE CARRY

ACCUMULATOR 302

(INCLUDES SHIFT ROUNDOFF CONTROL)

AR TO AC

AC TO BUS

DIGIT 6 TO SRC

SC PRESET
SC READ IN GATE

FROM STEP COUNTER

FROM OPERATION MATRIX ON C1

CYCLE LEFT
TO AC0 & AC1

CP0 86

SHIFT

SHIFT & CARRY

DIVIDE SHIFT LEFT

AC TO BR
BR TO AC
ONE TO RIGHT

ZERO TO RIGHT

ROUNDOFF CARRY

ZERO TO LEFT

ONE TO LEFT

BR CLEAR

LEFT

SHIFT RIGHT

16
16

AC CLEAR
AC CARRY CLEAR
AC READ OUT
AC READ TO BR
BR CLEAR VIA SRC
END-AROUND CARRY

DE 0.25 MSC

AC READ OUT

B-REGISTER 303

ONE TO BR15

ONE FROM BR15

DIVIDE CARRY

SHIFT RIGHT
SHIFT LEFT

ZERO FROM BR15

BR15 DIGIT SENSE

BR CLEAR
ROUNDOFF

CYCLE LEFT ONE TO BR15

DIVIDE PULSE DISTR. 308

GT .05

O FF .02 1

DE

GT .06

CLEAR

NO UN

Q1   Q2

DIVIDE

STEP COUNTER 305

rt s   rt 6

TO SRC IN AC 15

SC PRESET
SC READ IN GATE

SC PRESET

SC READ IN

SC READ OUT

SPECIAL CLEAR

SC END CARRY

DIVIDE RESET

SC CLEAR

MULTIPLY RESET

TO GT'S 307.06 & 307.07

ADD TO SC

SC CLEAR

ADD TO SC

DIVIDE SHIFT LEFT

ADD TO SC

LFCP

MFCP

DIVIDE

ARITHMETIC CONTROL

GT 308 .04

GT 307 .05

GT 307 .04

GT 308 .04

1 FF 308 .01 0

1 FF 307 .02 0

1 FF 307 .01 0

1 FF 308 .01 0

DE .35

DE .35

GT 307 .07

GT 307 .06

SHIFT RIGHT

SHIFT LEFT

MULTIPLY

C.B.

Before a number can be stored in AR, the register must be cleared of previous information.   The number is then read in through GT's .01 to FF's .01 and may be read back onto the bus, for checking, via GT's .02.

GT's .05 permit a number to be read out of AR and to be added into AC; GT's .04 permit the number to be subtracted (its complement to be added).

A complement line is provided which will switch all the FF's, essentially changing the sign of the number in AR when required for sign handling.   The "left-digit-one" line from AR 0 goes to the Sign Control to record AR as negative if AR 0 is a 1.

A-REGISTER

Sections AC 1-AC 14: A single AC digit column includes one partial-sum FF (FF.01) and one carry FF (FF.02). For construction reasons, the carry FF's are associated physically with the PS FF's one section to the left, i.e., a carry from PS FF.01 in AC 3 will be held in carry FF.02 in AC 2.

For addition, the incoming pulse from AR is supplied to DE.01 and GT.06 (see AC 14). If FF.01 holds a 1, GT.06 will be on and the incoming pulse will pass through GT.06 along the carry-digit line to add 1 into the associated carry FF (.02) to the left. After a delay, the pulse will switch FF.01 from 1 to 0. DE.01 is just long enough to allow the carry pulse to pass through GT.06 before FF.01 is changed to 0. An included high-speed-carry system uses GT.12 and GT.05 (Sec. 3.212).

In shifting left, GT.14 will be on during a shift-left command if FF.01 holds a 0, while GT.13 will be on if FF.01 holds a 1. A shift-left pulse will pass through the "on" tube and set the adjacent FF on the left to agree with the setting of the original FF. DE.02 and DE.03 delay the shifting pulses until the shift-left pulse has gone through all GT's.13 and .14.

The contents of AC may be read out to the main bus through GT.02, to BR through GT.04, and to check bus through GT.03. A clear line is provided for clearing AC prior to inserting a number; a complement line permits correcting the sign of the number in AC (after a division, for example). Crystal diodes prevent feedback of control pulses or commands into other control lines.

The whiffletree (GT's.16, .17, .08-.11), driven from the PS and carry FF's, is used for the shift-right-and-carry operation (Sec. 3.233). This 4-way tree converts the original settings of the PS and carry-digit FF's into control pulses for properly setting the FF's during shift-and-carry. One of the GT's.08-.11 is selected by the tree through GT's.16 and .17, depending on whether FF.01 holds a 0 or a 1 and on whether FF.02 holds a 0 or a 1. The shift-and-carry command pulse proceeds through the selected GT to set the PS FF one digit to the right. Thus the whiffletree decides whether to shift 1's or 0's to the right according to the contents of the PS and carry FF's.

Sections AC 0, AC 15: AC 0 and AC 15 differ somewhat from AC 1-AC 14. AC 0 has a connection between the high-speed-carry line (carry-digit-from-right) and the left-digit-carry line as well as a special divide-shift-left line which shifts AC 0 and AC 1 left during division. These digits are not shifted during normal shift operations. AC 15 has no carry FF. (The carry FF associated with AC 15 is actually part of the AC 14 chassis.) Shift-and-carry is reduced to a simple shift-right to BR 0, involving only GT.15 and GT.16. FF.03 is used for Shift Roundoff Control. In shifting, the contents of BR can be retained or cleared and rounded off to AC 15.
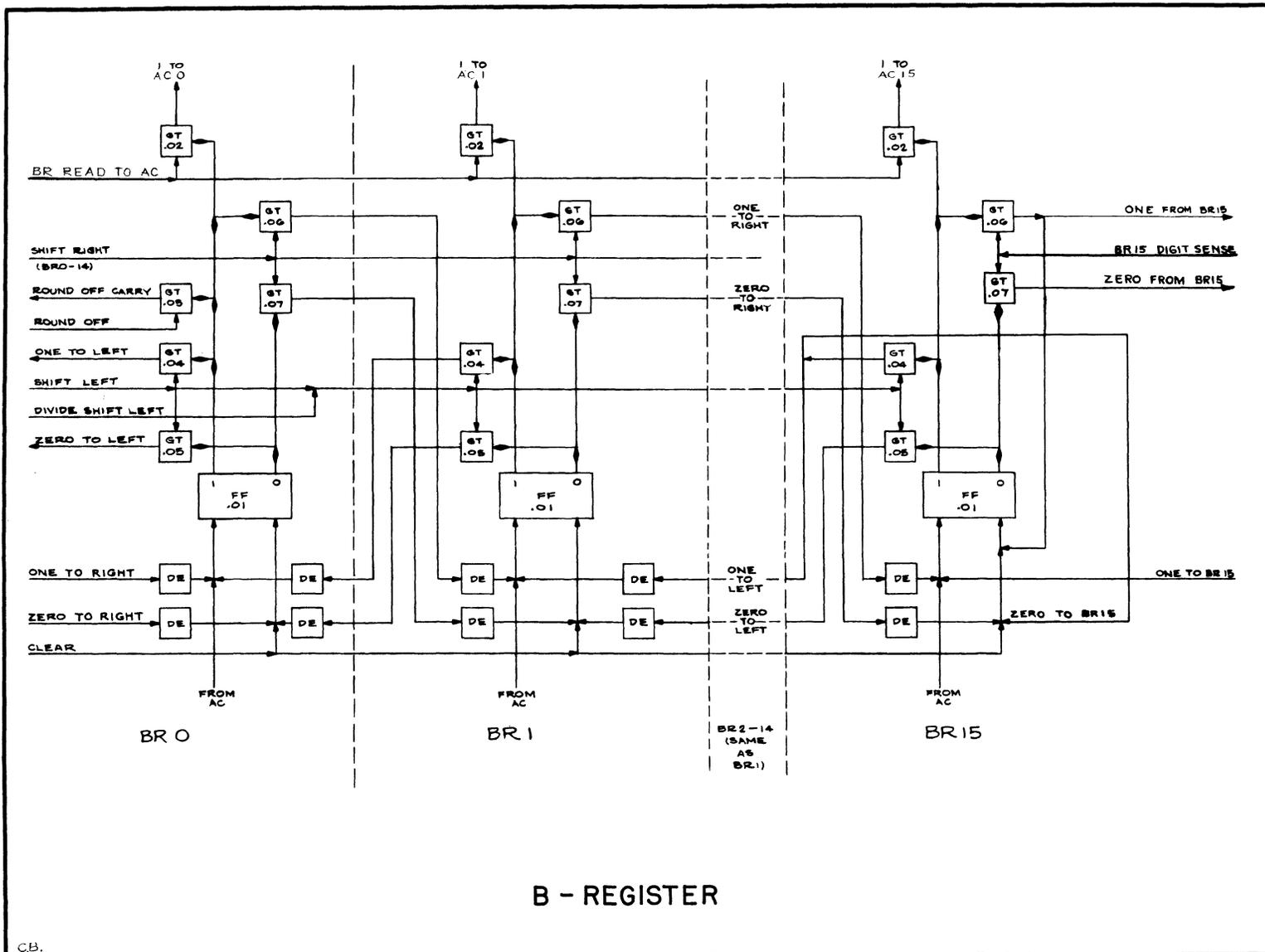
ACCUMULATOR

Sections BR 1 - BR 14

The B-Register stores a number in the 16 FF's. 01. A clear line is provided, and also a means for reading in the contents from AC. The BR ordinarily does not read out except by shifting left into AC. (Read-out GT's. 02 are used for the operation ab in which the contents of BR are supplied to the AC FF inputs.)

Two pairs of gate tubes with delays are used for shifting right and left from each digit to its neighbor.

Sections BR 0, BR 15

An extra gate tube GT. 03 is added to BR 0 for providing roundoff. The divide-shift-left line does not become effective until GT's. 04 and . 05 of BR 1, preventing BR 0 from being shifted on divide.

The only difference in BR 15 is the addition of the line connecting "one from BR 15" to the 0 side of BR 15. This line is used for resetting BR 15 during multiplication. (See Section 3. 23, Figure 25.) The regular shift-right gate tubes, GT. 07 and GT. 06, are used to get the "from-BR 15" signals necessary for multiplication.
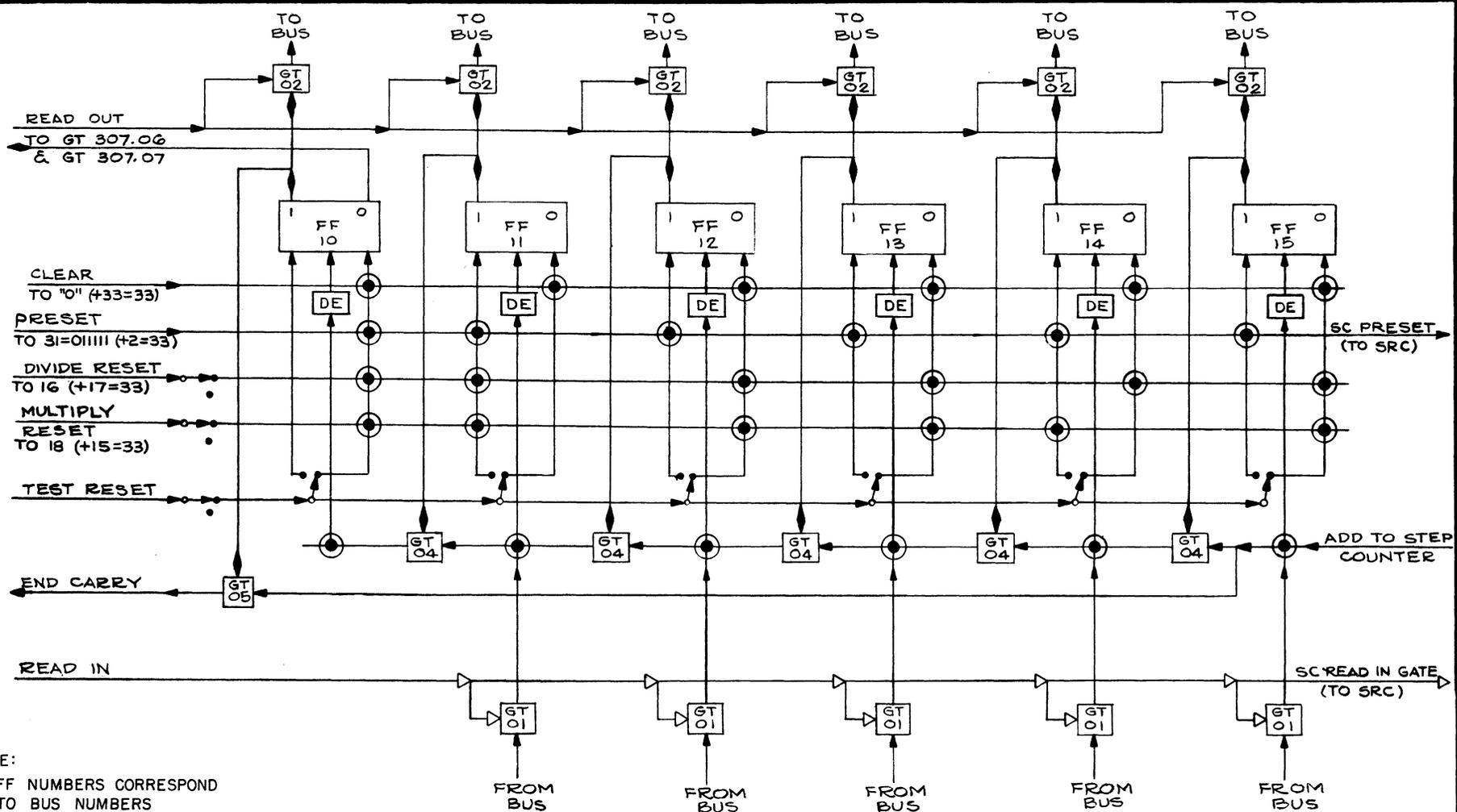
B - REGISTER

STEP COUNTER -- FIGURE 71

The Step Counter is used in the Arithmetic Element to count the number of steps performed in shifting, multiplying, dividing, cycling, and scale factoring. Six sections are needed to allow the counter to count to 33. The sixth section minimizes delay time by providing an end-carry line with only one gate tube.

In divide and in multiply, the counter is reset by a pulse on the designated line; crystal diodes are connected to give the initial settings of the counter. The toggle switches on the test-reset line allow convenient changing of counter settings for testing. After the proper number of "add-to-step-counter" pulses have been supplied, the counter flip-flops 11-15 will all have returned to 0 and FF.10 will contain a 1. The next add pulse will then pass through GT.05 as an end carry, indicating the completion of an operation; it will also set FF.15, giving $32 + 1 = 33$.

For the shift and cycle operations, the counter is set by the "address," which comes in from the bus and which indicates the number of digits to be shifted or cycled. (See Section 3.25, Figure 33.) The counter is first cleared. The address is then read into the complemented input of the counter from the bus through GT's.01. The counter will permit shifting and cycling to occur until an overflow puts out an end carry.

GT's.02 are used for reading the counter contents back onto the bus after an sf (scale factor) operation.

STEP COUNTER

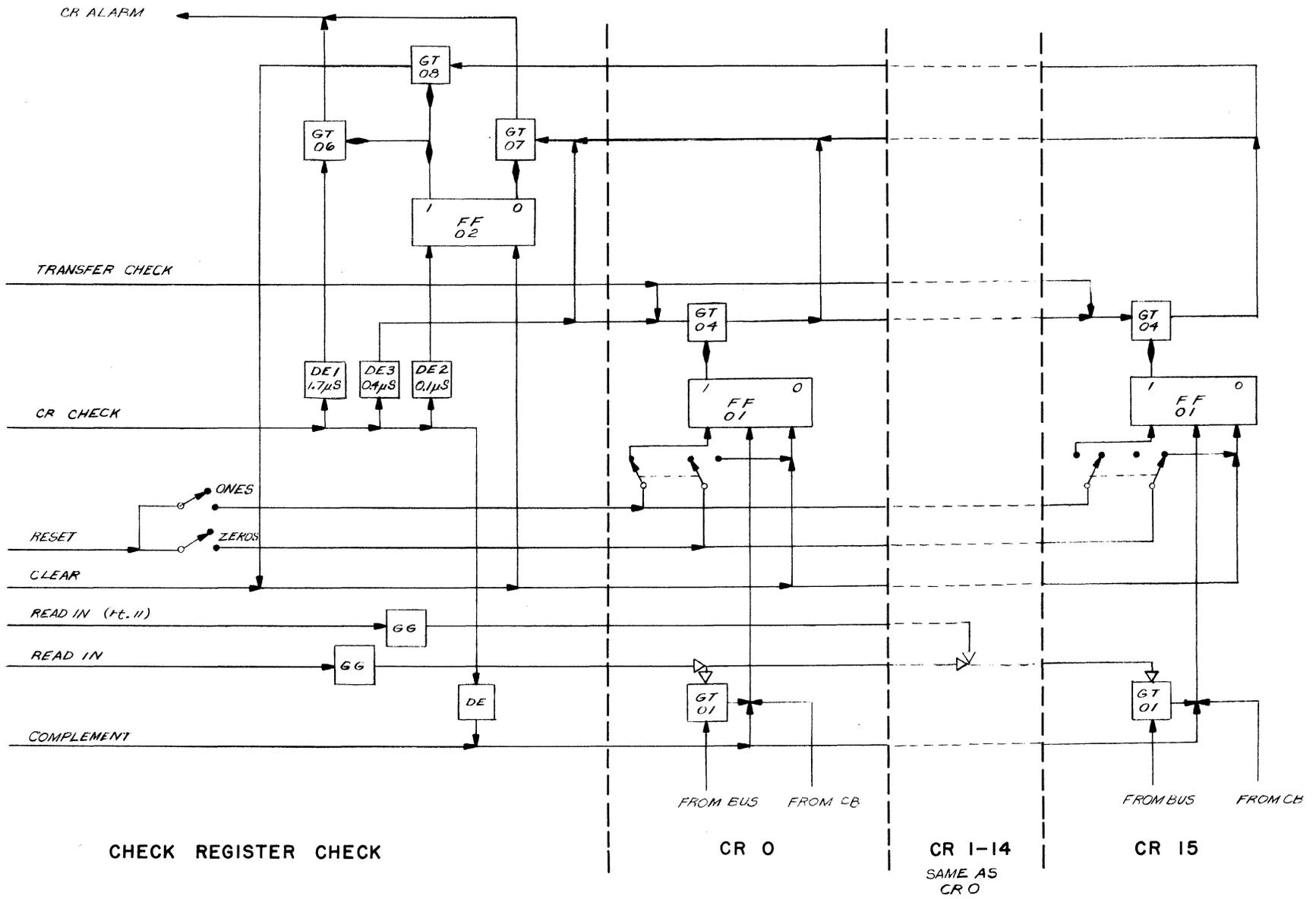NOTE:
FF NUMBERS CORRESPOND
TO BUS NUMBERS

C.B.

The check register contains 16 FF's .01 which can be read into from both main bus and check bus. The complement input is always used. A number on the main bus is read in through GT's .01, whereas a number on the check bus goes directly into CR.

When CR is clear, if the numbers coming in from the main bus and the check bus are identical, all FF's .01 will then hold 0 (complement and recomplement). If there is any discrepancy, one of the flip-flops will hold a 1, GT .04 will be on, and a "transfer check" pulse will produce an alarm.

The register is self-clearing in the case of no error. A manual-clear line is provided for clearing the register after correction of a fault.

The transfer-checking system depends on the check register's continual correct operation. To check that CR is operating correctly, it is itself continually checked. The command "CR check" first complements the cleared CR, so that all FF's .01 should then be set to 1. After a 0.1-μsec delay, the CR check pulse sets FF .02 to 1; after a 0.4-μsec delay, it attempts to go through all GT's .04 in series. (It will be able to do this if all FF's .01 are still holding 1 as they should be.) After passing GT .04 of CR 15, it will pass GT .08 (because FF .02 has been set to 1 earlier) and will come back in on the clear line, resetting FF .02 to 0. After a 1.7-μsec delay it will try to pass GT .06 but will be unsuccessful because FF .02 has been cleared. No CR alarm pulse, then, will be produced, proof that the check register is functioning properly.
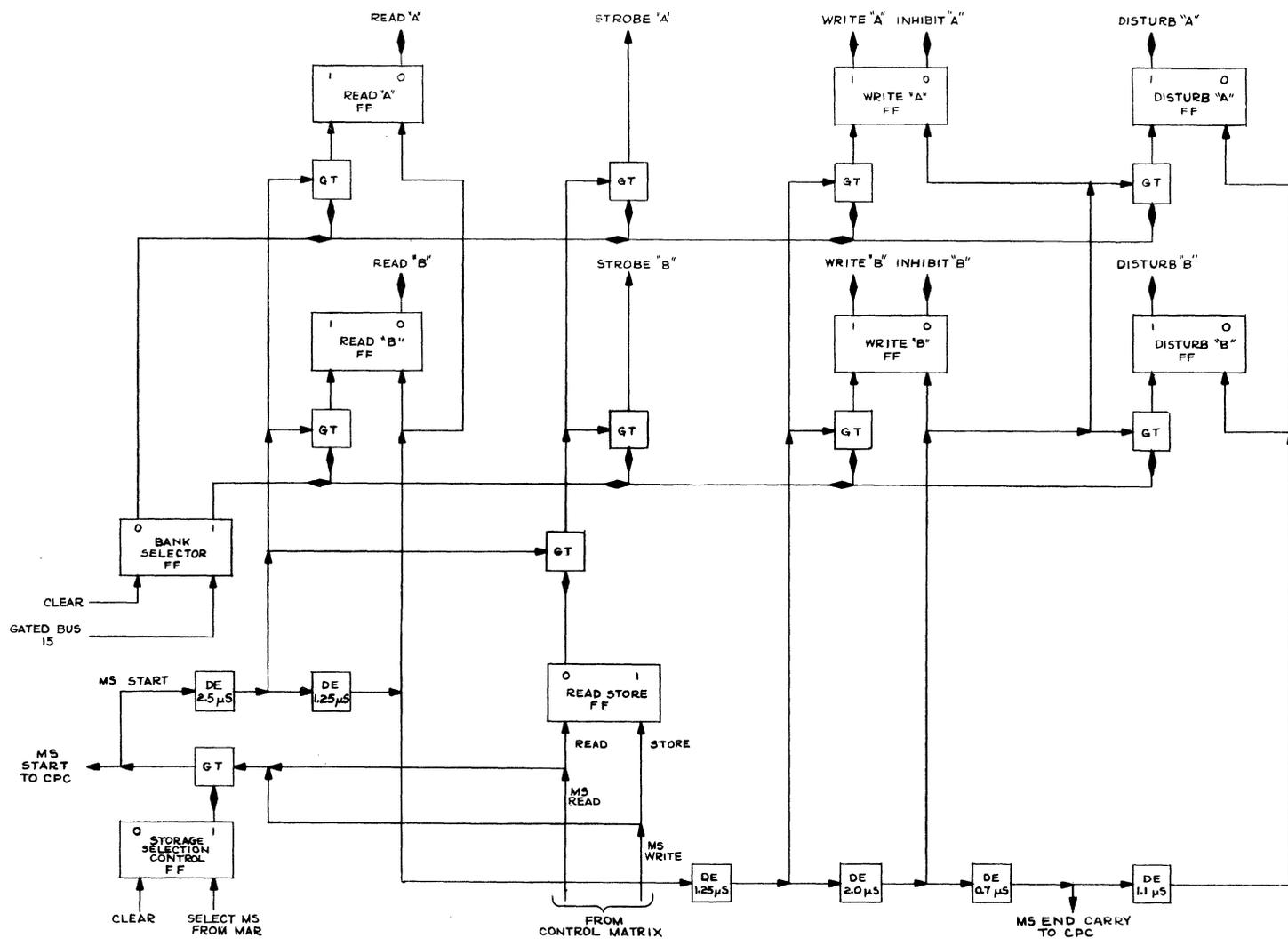
If the CR-check pulse, delayed 0.4 μsec, should fail to pass any of the GT's .04 (predicating the failure of any digit of the check register to hold a 1), there will be no pulse passing GT .04 of CR 15 through GT .08 to clear FF .02. Then when the CR-check pulse, delayed 1.7 μsec, comes along, it will pass GT .06 because FF .02 still holds a 1, and a CR alarm will be produced.

CHECK REGISTER (SIMPLIFIED)

For a detailed discussion of the operation of magnetic-core storage in Whirlwind I, see Section 5 of this report.

MAGNETIC — CORE— MEMORY CONTROL

(TWO-BANK OPERATION)

MIT