R.P.L

D.C.S.

DISK CONTROLLER

D.C.S. DISK-DKT-STREAMER-CONTROLLER

CONNECTED TO THE S.G.M.2 SYSTEM IS A SINGLE BOARD THAT PROVIDES THE TRADITIONAL AND ADVANCED FEATURES REQUIRED TO CONTROL WINCHESTER TYPE HARD DISK, FLOPPY DISK AND STREAMER TAPE

FEATURES

- WINCHESTER DISK INTERFACE ST506 (MFM protocol)
- FLOPPY DISK INTERFACE
- STREAMER TAPE INTERFACE QIC02
- SUPPORT MFM PROTOCOL
- SOFT SECTORED
- CYLINDER NUMBER UP TO 16 BIT
- PROGRAMMABLE DISK FORMAT
- DOUBLE SPEED ON FLOPPY 250-500 KHz (GSV, 1 Meg.)
- MULTISECTOR READ/WRITE
- ERROR CORRECTION UP TO 11 BIT
- 32 BIT ECC CODE
- DISK SERIAL DATA RATE 5 MEGABITS/SEC
- SELF DIAGNOSTIC
- AUTOMATIC BAD SECTOR HANDLING
- HIGH LEVEL COMMANDS
- OVERLAPPING FUNCTIONALITIES
- MASTER/SLAVE MODE
- ALL THE COMMAND ALLOWED ON STREAMER TAPE

BOARD CONFIGURATION

- UP TO 3 WINCHESTER TYPE HARD DISK
- 1 FLOPPY DRIVE SINGLE-DOUBLE SPEED
- 1 STREAMER TAPE QIC02 INTERFACE

ABOUT DISKS NO RESTRICTION
IN TERM OF CONNETTIBILITY
FROM A CAPACITY POINT OF VIEW

UP TO 2 DCS CAN BE CONNECTED
TO THE SGM2 SYSTEM

INTERNAL LOGIC BLOCK

INTERFACE AND LOGICAL HANDLING VME BUS

DATA TRANSFER BUS ADDRESS

THIS FUNCTIONAL BLOCK SUPPLIES IN OUTPUT THE MAIN MEMORY ADDRESS 32 BIT IN MASTER MODE.

IN SLAVE MODE ACCEPTS IN INPUT THE ADDRESS BIT RELATED TO THE MAPPED I/O TO GENERATE INTERNAL REGISTER SELECTION.

FIRST DCS IS MAPPED I/O STARTING FROM:

54000000 HEX ADDRESS

SECOND DCS IS MAPPED I/O STARTING FROM:

55000000 HEX ADDRESS

DATA TRANSFER BUS DATA

THIS FUNCTIONAL BLOCK SUPPLIES AND ACCEPTS 8/16 BIT OF DATA TO/FROM VME BUS

DATA TRANSFER BUS CONTROL LINES

THIS FUNCTIONAL BLOCK CONTROLS ALL THE SIGNALS INVOLVED DURING A READ/WRITE CYCLE IN MASTER-SLAVE MODE

ARBITER INTERFACE

THIS FUNCTIONAL BLOCK CONTROLS ALL THE SIGNALS INVOLVED DURING A REQUEST BUSCYCLE.
THIS BLOCK HANDLES ALSO THE DAISY CHAIN CONNECTION BETWEEN CONTROLLERS CONNECTED TO THE SAME BUS REQUEST.

D.C.S USES BUS REQUEST N° 3

INTERRUPTER

THIS FUNCTIONAL BLOCK IS CAPABLE TO GENERATE INTERRUPT REQUEST TO THE HOST AND RECOGNISE THE PROPER INTERRUPT ACKNOWLEDGE RESPONSE. THIS BLOCK HANDLES ALSO THE DAISY CHAIN CONNECTION BETWEEN CONTROLLERS CONNECTED TO THE SAME INTERRUPT REQUEST. DCS WILL SUPPLIES ALWAYS AN INTERRUPT VECTOR DURING AN INTERRUPT CYCLE COMPATIBLE WITH THE VME BUS PHILOSOPHY.

DCS USES INTERRUPT REQUEST N° 4

MAJOR COMPONENTS DESCRIPTION

DISK CONTROLLER CONTAINS THE FOLLOWING ITEMS:

- IMDC 68454 SIGNETICS
  INTELLIGENT MULTIPLE DISK CONTROLLER
- DATA SEPARATOR
  BASED ON DPLL 68459 SIGNETICS
- SPECIFIC HW FOR TAPE HANDLING
- INTERRUPTER MODULE
  BASED ON BIM 68153 MOTOROLA

IMDC 68454

THE SCN68454 INTELLIGENT MULTIPLE DISC CONTROLLER PROVIDES THE TRADITIONAL AND ADVANCED FEATURES REQUIRED TO CONTROL WINCHESTER RIGID DISK AND FLOPPY DISK DRIVES.

FEATURES

- MOS TECHNOLOGY 48 PINS
- 31 BIT ADDRESS COUNTER
- 8/16 BIT DATA CHANNEL
- VECTORED INTERRUPT
- AUTOMATIC RERUN ON BUS ERROR
- ON CHIP DMA CONTROLLER
- ON CHIP FIFO BUFFER 128 BYTES
- AUTOMATIC MASTER FUNCTIONALITIES
- SUPPORTS UP TO 4 CHANNELS IN ANY MIX
- SUPPORTS SA1000 AND ST506 INTERFACE
- DATA RATE UP TO 10 Mbits/SEC MFM PROTOCOL
- HIGH LEVEL COMMAND SET
- MULTIPLE SECTOR R/W WITH IMPLIED SEEK
- OVERLAPPING ON SEEK COMMANDS
- PROGRAMMABLE TRACK FORMAT
- AUTOMATIC BAD SECTOR HANDLING
- 32/40 bit ECC PROGRAMMABLE POLYNOMIALS

OPERATING MODES

THE IMDC PROVIDES FOUR POSSIBLE OPERATING MODES.

- REGISTER MODE

  REFER TO THE STATE WHEN THE IMDC IS CHIP SELECTED TO ALLOW INTERNAL REGISTER READ-WRITE

- DMA MODE

  DMA MODE REFERS TO THE STATE WHEN THE IMDC ASSUMES OWNERSHIP OF THE BUS

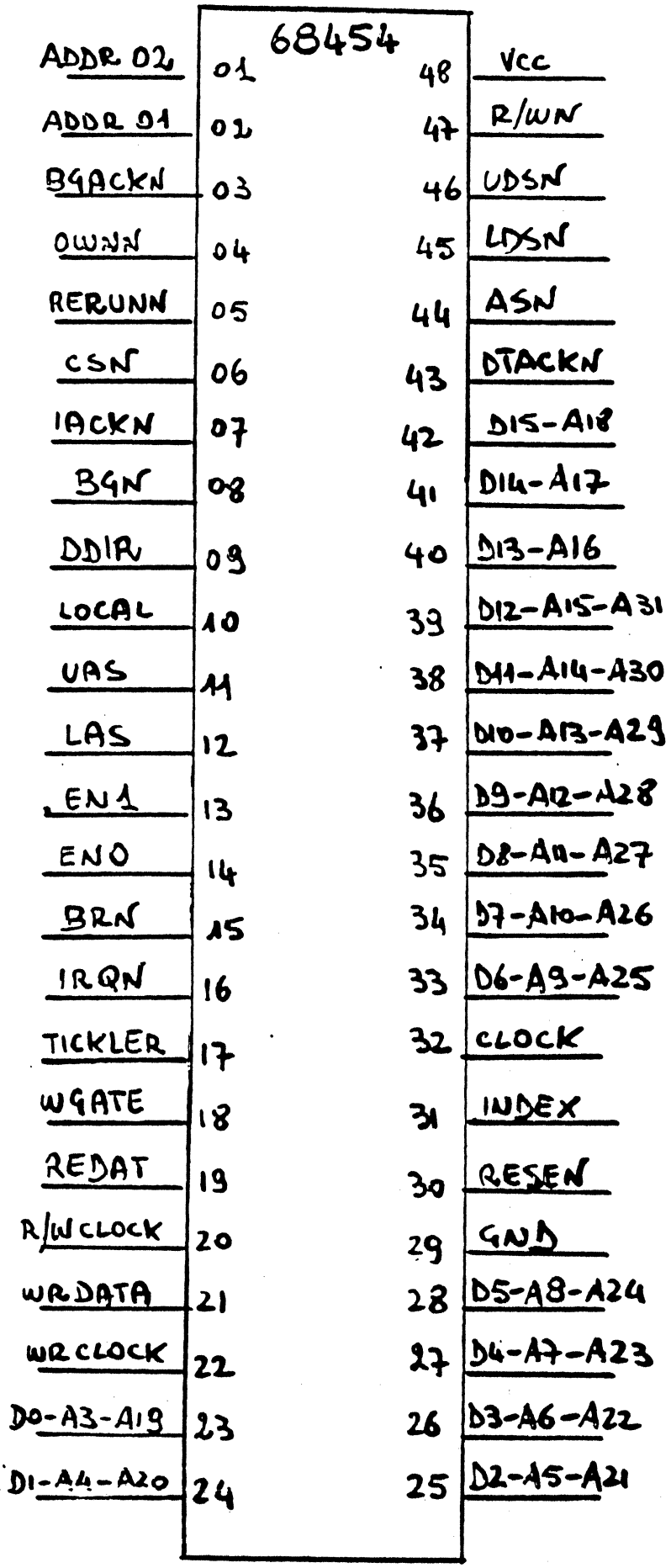
- LOCAL MODE

  REFERS TO THE STATE WHEN THE IMDC IS TRANSFERRING COMMANDS TO THE DISK INTERFACE OR RECEIVES CONTROLS INFORMATIONS FROM DISK INTERFACE

- IDLE MODE

  THIS MODE REFERS TO A WAIT STATE

FDC PIN-OUT


68454
ADDR 02
01
ADDR 01
02
BGACKN
03
OWNN
04
RERUNN
05
CSN
06
IACKN
07
BGN
08
DDIR
09
LOCAL
10
UAS
11
LAS
12
EN1
13
EN0
14
BRN
15
IRQN
16
TICKLER
17
WGATE
18
REDAT
19
R/W CLOCK
20
WR DATA
21
WR CLOCK
22
D0-A3-A19
23
D1-A4-A20
24
48
VCC
47
R/WN
46
UDSN
45
LDSN
44
ASN
43
DTACKN
42
D15-A18
41
D14-A17
40
D13-A16
39
D12-A15-A31
38
D11-A14-A30
37
D10-A13-A29
36
D9-A12-A28
35
D8-A11-A27
34
D7-A10-A26
33
D6-A9-A25
32
CLOCK
31
INDEX
30
RESEN
29
GND
28
D5-A8-A24
27
D4-A7-A23
26
D3-A6-A22
25
D2-A5-A21

INTERNAL REGISTERS MAP

SEVEN INTERNAL REGISTERS ARE FORESEEN ON IMDC

HEX ADDR. 54000000 / 55000000 EPH ECA POINTER HIGH

HEX ADDR. 54000001 / 55000001 EPMH ECA POINTER MIDDLE HIGH

HEX ADDR. 54000002 / 55000002 EPML ECA POINTER MIDDLE LOW

HEX ADDR. 54000003 / 55000003 EPL ECA POINTER LOW

THESE FOUR REGISTERS ARE USED TO DIRECT THE IMDC TO A TABLE OF POINTERS AT RESET TIME THESE REGISTERS WILL BE INITIALIZED TO ZERO

HEX ADDR. 54000004 / 55000004 IVR INTERRUPT VECTOR REGISTER

THIS REGISTED CONTAINS THE VALUE OF VECTOR THAT THE IMDC WILL PLACE ON THE DATA BUS UPON RECEIPT OF AN INTERRUPT ACKNOWLEDGE

HEX ADDR. 54000005 / 55000005 ISR INTERRUPT SOURCE REGISTE

FOR THIS REGISTER JUST FOUR BIT ARE MEANINGFULLY. BIT 7-6-5-4 SET TO 1 MEAN THE CHANNEL THAT REQUIRED A COMMAND COMPLETION INTERRUPT

BIT 4	REFLECT	CHANNEL	0
BIT 5	"	"	1
BIT 6	"	"	2
BIT 7	"	"	3

HEX ADDR. 54000006 / 55000006 DSCR DRIVE STATUS AND CONFIGURATION REGISTER

BIT 0 MEANS 8/16 bit DATA MODE

BIT 4-5-6-7 MEAN DRIVE BUSY AND ARE USED TO START THE I/O OPERATION

BIT 4	REFLECT	CHANNEL	0
BIT 5	"	"	1
BIT 6	"	"	2
BIT 7	"	"	3

DATA SEPARATOR

DATA SEPARATOR BLOCK USES THE DPLL 68459 BIPOLAR CHIPS.
DATA SEPARATOR PROVIDES THE CAPABILITY TO LOCK THE INPUT STREAM OF DATA AND SUPPLIES IN OUTPUT A STREAM OF DATA IN PHASE WITH THE PROPER CLOCK TO OBTAIN THE CORRECT DATA CODED.
TWO DPLL ARE FORESEEN ON DCS BOARD

- FIRST DPLL USED FOR HARD DISK AND RUNNING AT 5MHZ
- SECOND DPLL USED FOR FLOPPY DISK AND RUNNING AT TWO DIFFERENT SPEED
  - 250 KHz for Single Speed double density
  - 500 KHz for Double Speed double density

THE DPLL PROVIDES ALSO THE WRITE DATA IN MFM PROTOCOL STARTING FROM A WRITE DATA IN NRZ MODE AND ASSOCIATED CLOCK.

2) DPLL SCB68459

The SCB68459 Disk Phase Locked Loop (DPLL) is a bipolar chip that used together with the IMDC 68454 provides all the functions to control disk/dkt devices with SA800,ST506,ST1000 types interfaces.
The DPLL uses an external VCO for the variable clock rate which tracks the read data from the disk unit.
Disk/dkt controller uses two DPLL chips because two different frequences are needed.In fact disk device runs at 5Mbit per second while dkt device runs at 250/500Kbit per second.
The SCB68459 operates producing an oscillator frequency to match the frequency of an input signal.
In this locked condition any slight change in the input frequency will appear as a change in phase between the input frequency and the VCO frequency.This phase shift then acts as an error signal to change the frequency of the local DPLL VCO to match the input frequency.
The DPLL SCB68459 is able to accept NRZ data and write clock information in parallel from SNC68454.It combines data and clock to obtain a resulting signal that becomes the write data signal which goes to the disk drive.
The DPLL also built write precompensation algorithm in order to adjust special clock data patterns that require compensation.An external delay line is used to generate the precision compensation needed.
With a PLL circuit built into a read chain data can be more accurately recovered.Aided by an external voltage controlled oscillator the DPLL function derives a clock signal from the read data streams and tracks it through a range of variation.
The decoding hardware uses the clock signal to couple the read data stream.

REGISTER PROGRAMMING

HEX ADDR.	54005000 55005000	SDM	SINGLE SPEED MODE SELECTED
HEX ADDR.	54006000 55006000	DDM	DOUBLE SPEED MODE SELECTED

THESE TWO POINTERS SELECT THE CORRECT FREQUENCY TO LOCK WORKING AT DPLL AND PLO LEVEL.
AT RESET TIME THIS LOGIC SELECTS THE SINGLE SPEED MODE

HW for TAPE HANDLING

ALL THE HW FEATURES ARE FORESEEN ON THE DCS BOARD IN ORDER TO SUPPORT THE TAPE QIC02 INTERFACE CONNECTION.
A DEDICATED DATA PATH PROVIDES THE COMMAND AND DATA IN WRITE OPERATION AND RECEIVES STATUS AND DATA IN READ OPERATION.
A DEDICATED LOGIC IS FORESEEN TO SEMPLIFY THE HANDSHAKE WITH THE DRIVE DURING THE R/W OPERATION IN ORDER TO MINIMIZE THE ACCESS TO THE VME BUS.
READY AND EXCEPTION SIGNALS CAN BE USED BOTH IN POLLING OR INTERRUPT MODE.

ADDRESSING REGISTERS

HEX ADDR.	54003000 55003000	CRW	COMMAND REGISTER WRITE
HEX ADDR.	54003002 55003002	CRR	COMMAND REGISTER READ
HEX ADDR.	54003004 55003004	DRW	DATA REGISTER WRITE
HEX ADDR.	54003006 55003006	DRR	DATA REGISTER READ
HEX ADDR.	54003008 55003008	RFFR	READY RESET
HEX ADDR.	5400300A 5500300A	RFFE	EXCEPTION RESET

TAPE

INTERRUPTER MODULE

SPECIFIC HW USING THE 68153 MOTOROLA BUS INTERRUPTER MODULE THAT HANDLES THE INTERRUPT PROTOCOL AT VME BUS SIDE AND SUPPLIES CONTROL SIGNALS TO THE INTERNAL COMPONENTS.
SPECIFICALLY IT SERVES THE INTERRUPT FUNCTIONALITY AT TAPE LEVEL TO HANDLE THE READY-EXCEPTION SIGNALS BOTH IN POLLING OR INTERRUPT MODE PROGRAMMED BY SOFTWARE.
IT SUPPLIES ALSO THE CONTROL SIGNALS TO THE IMDC IN ORDER TO ALLOW THE INTERRUPT PROTOCOL.

REGISTERS PROGRAMMING

THE BUS INTERRUPTER MODULE CONTAINS 8 REGISTERS

HEX ADDR.	54001001 55001001	CR0	CONTROL REGISTER 0
HEX ADDR.	54001003 54001003	CR1	" 1
HEX ADDR.	54001005 55001005	CR2	" 2
HEX ADDR.	54001007 55001007	CR3	" 3

THESE FOUR REGISTERS CONTAIN THE INFORMATION TO PROGRAM THE INTERRUPT PROTOCO

F	FAC	$\overline{XIN}$	IRE	IRAC	L2	L1	L0

apparently 68153 Interrupter mod

L0 ÷ L2 LEVEL AT WHICH AN INTERRUPT WILL BE GENERATED

IRAC TO RESET AUTOMATICALLY IRE (NOT USE

IRE TO ENABLE THE INTERRUPT REQUEST

$\overline{XIN}$ TO SELECT INTERNAL OR EXTERNAL VECT

FAC TO RESET AUTOMATICALLY FLAG (NOT USE

FLAG TO ALLOW PROCESSOR COMMUNICATION AND RESOURCE ALLOCATION (NOT USED)

Interrupt
~~RAPR~~

HEX ADDR.	54001009 55001009	VR0	VECTOR	REGISTER	0
HEX ADDR	5400100B 5500100B	VR1	"	"	1
HEX ADDR.	54001009 5500100D	VR2	"	"	2
HEX ADDR.	5400100F 5500100F	VR3	"	"	3

THESE FOUR REGISTERS CONTAIN THE INTERRUPT VECTOR.

IN DCS JUST THREE INTERRUPT REQUEST ARE USED ON BIM AND ONLY THE LAST TWO VECTORS REGISTERS ARE USED FOR TAPE PURPOSES.

CHANNELS CONFIGURATION

DCS CONNECTS UP TO 3 HARD DISK IN DAISY CHAIN, 1 FLOPPY DRIVE AND 1 STREAMER TAPE.

HARDWARE AND SOFTWARE VISIBILITY AT CHANNEL LEVEL IS:

CHANNEL	0	HARD	DISK	} ANY MIX
"	1	"	"	
"	2	"	"	
"	3	FLOPPY	DISK	single/double s
"	X	TAPE	DRIVE	QIC02

DEVICE ADDRESSABILITY

AT SOFTWARE LEVEL THE DEVICE IS SELECTED FOR START THE I/O OPERATION USING THE DSCR REGISTER IN THE FOLLOW CONFIGURATION:

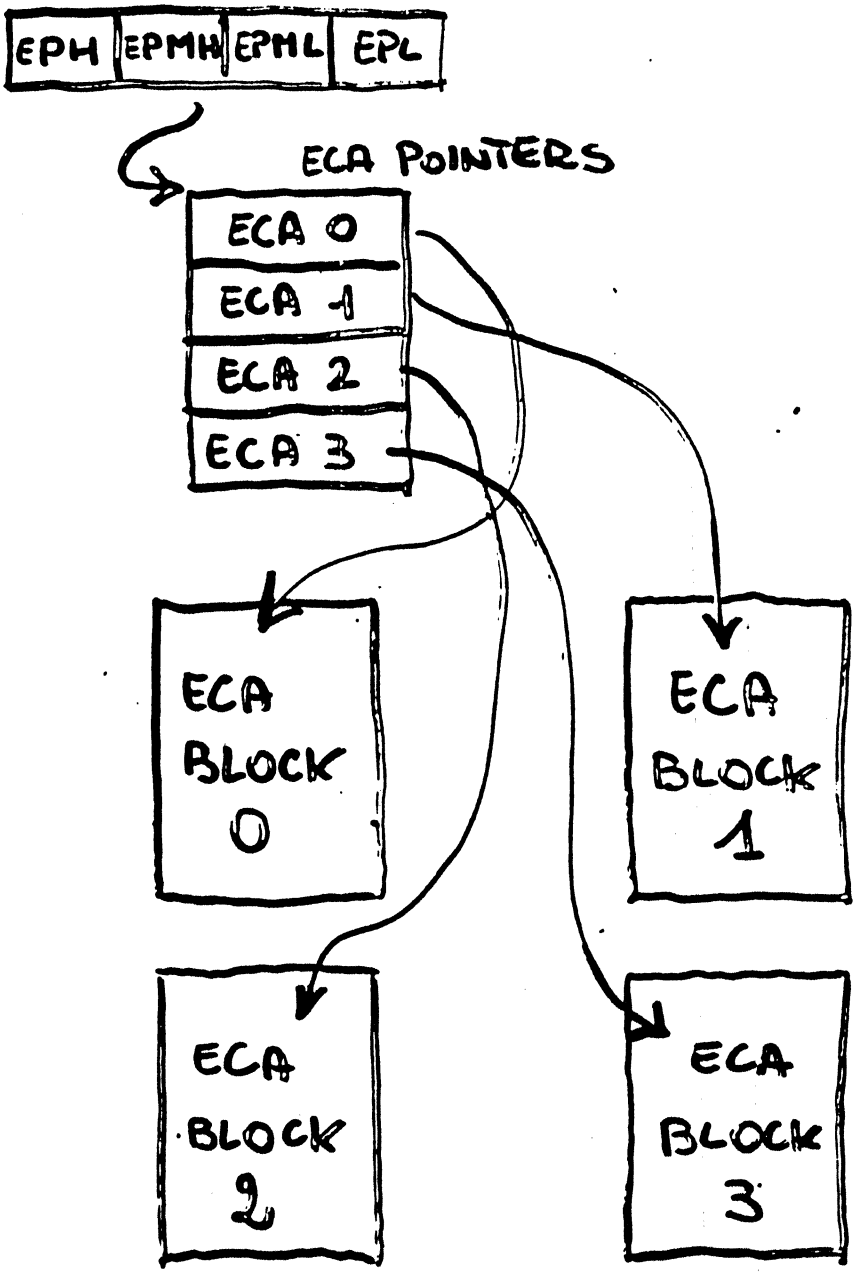
CHANNEL	0	SELECT	11	IN DSCR
"	1	"	21	"
"	2	"	41	"
"	3	"	81	"

FOR TAPE DRIVE NO SELECTION IS NEEDED

LOGICAL STRUCTURE

THE IMDC IS CAPABLE TO EXECUTE AN I/O OPERATION, AFTER THE PROPER BIT IN DSCR HAS BEEN SET, BY MEANS OF USE OF SOME TABLES IN MAIN MEMORY.

TABLES STRUCTURE



MAIN MEMORY TABLES

THIS TABLE IS ADDRESSED BY THE CONTENT OF THE EPH-EPHH-EPHL-EPL REGISTERS AND CONSISTS OF FOUR 2 WORDS ADDRESSES THAT POINT TO THE LOCATION OF FOUR ECS BLOCKS. THE POINTERS MUST BE ARRANGED IN ASCENDING ORDER BY DRIVE

ECA BLOCK

THE IHDC NEEDS THE ECA TABLE (EVENT CONTROL AREA) TO COMUNICATE WITH THE HOST SYSTEM.
AN ECA TABLE MUST BE SET FOR EACH DISK/DKT DRIVE TO BE CONTROLLED.
ECA TABLE CONTAINS INFORMATIONS ABOUT THE REQUESTED COMMAND AND THE DISK DRIVE.
THE INTERNAL MICROPROGRAM WILL USE THESE TABLES TO GENERATE THE DISK INTERFACE SIGNAL AND PERFORMS THE I/O OPERATION. THE IHDC ALSO WILL USE THESE TABLES TO RETURN THE STATUS INFORMATIONS TO THE SOFTWARE.
THE DISK DRIVE TO ECA BLOCK ASSIGNEMENT IS DETERNINED BY THE RELATIVE POSITION OF THE POINTER IN THE TABLE.

FUNCTIONS OF ECA BLOCK

00	COMMAND CODE	MAIN STATUS	
01	EXTENDED STATUS		
02	MAX # OF RETRIES	ACTUAL # OF RETRIES	
03	DMA COUNT	COMMAND OPTIONS	
04	BUFFER ADDRESS MOST SIGNIFICANT WORD		
05	BUFFER ADDRESS LEAST SIGNIFICANT WORD		
06	BUFFER LENGTH REQUESTED		1
07	# OF BYTES TRANSFERRED		
08	CYLINDER NUMBER		
09	HEAD #	SECTOR #	
10	CURRENT CYLINDER POSITION		
11	P.R.P CONTROL WORD		
12	SCWT MOST SIGNIFICANT WORD		
13	SCWT LEAST SIGNIFICANT WORD		
14	SCAN TERMINATOR	RFU	
15	MAXIMUM RECORD LENGTH - 1		
16	NO PRE INDEX GAP	N1 POST INDEX GAP	2
17	N2 SYNC BYTE COUNT	N3 POST ID GAP	
18	N4 POST DATA GAP	N5 ADD. MARK COUNT	
19	RFU	RFU	
20-22	ECC MASK 3 WORDS		1
23	MOTOR ON DELAY (μsec)	# OF HEADS	3
24	END SECTOR #	STEPPING RATE	
25	HEAD SETTLING TIME	HEAD LOAD TIME	
26	SEEK TYPE	PHASE COUNT	
27	LOW WRITE CURRENT TRACK		
28	PRECOMPENSATION TRACK		
29-31	ECC REMAINDER 3 WORDS		1
32	MAX NUMBER CYLINDER PER SURFACE		3
33	SECTOR LENGTH	FLAG BYTE	
34-35	B TREE POINTER		
36-45	IMDC WORKING AREA		

THREE DIFFERENT PARTS ARE FORESEEN IN ECA BLOCK

1) THIS FIRST AREA CONTAINS THE ECA DATA FIELD REQUIRED BY THE IMDC TO EXECUTE A COMMAND. THESE FIELDS CAN BE FURTHER SEPARATED INTO STATIC AND DYNAMIC FIELDS. THE STATIC FIELDS ARE VALID FOR THE DURATION OF A COMMAND. DYNAMIC FIELDS PROVIDE THE LOCAL STORAGE NEEDED TO EXECUTE MULTIPLE SECTOR COMMANDS.

2) FORMAT PARAMETERS
THESE PARAMETERS ARE USED BY THE IMDC DURING FORMAT OPERATION. THESE PARAMETERS DEFINE THE PHYSICAL TRACK FORMAT.

3) DRIVE CONTROL PARAMETERS
THESE FIELDS CONTAIN INFORMATIONS THAT DEFINE THE DISK INTERFACE TO THE IMDC

Not a buffured system
DCS is highest Priority on VME

DESCRIPTION OF MASTER-SLAVE FUNCTIONALITIES

THE DCS CAN BE EITHER MASTER OR SLAVE ON THE SYSTEM BUS DEPENDING BY THE FUNCTIONALITY EXECUTED.
DCS WILL BE SLAVE ANY TIME THAT THE HOST SYSTEM NEEDS TO READ OR WRITE AN INTERNAL REGISTER.
DCS WILL BE MASTER ANYTIME THAT THE IMDC NEEDS THE BUS TO TRANSFER DATA OR CONTROL INFORMATIONS WITH THE MAIN MEMORY. IN CASE OF IMDC DATA TRANSFER THE DCS WILL BE MASTER FOR THE DURATION OF THE DATA TRANSFER THAT COULD BE SINGLE WORD ACCESS OR BURST MODE. THE SOFTWARE CAN PROGRAMME THE DCS TO EXECUTE UP TO 16 CONTINOUS WORDS BURST MODE TRANSFER. THIS FEATURE IS IMPORTANT SINCE THE DCS IS A NON BUFFERED CONTROLLER AND THE DATA TRANSFER OCCOURS BETWEEN THE INTERNAL FIFO AND THE MAIN MEMORY.

THEORY OF OPERATIONS

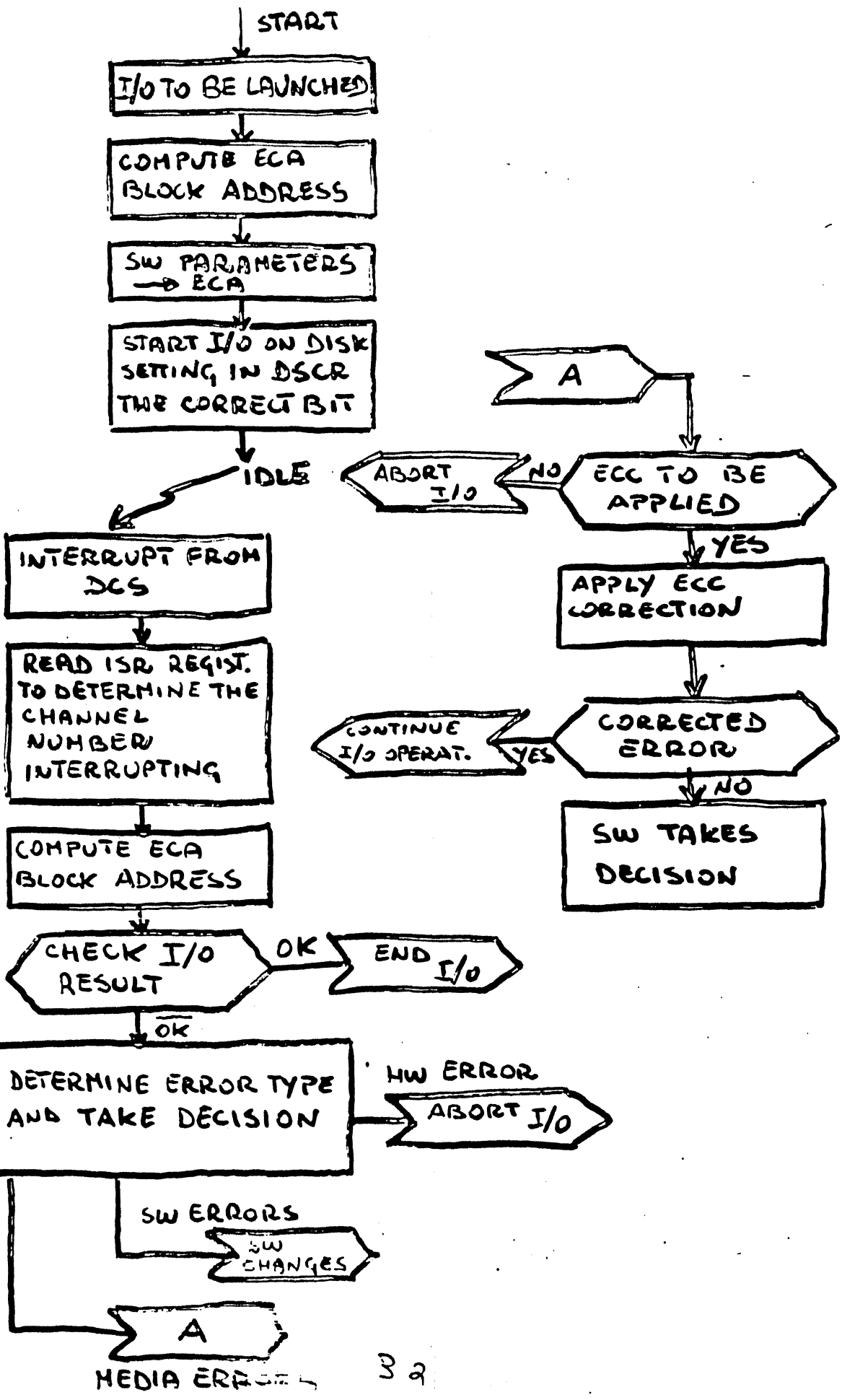
DCS IS CAPABLE TO EXECUTE A SET OF COMMANDS BY MEANS OF A COMMAND CODE BYTE IN ECA BLOCK.

TWO TYPES OF COMMANDS CAN BE DISTINGUISHED:

1) OFF LINE COMMANDS (not to M14)
2) DATA TRANSFER COMMANDS

- OFF LINE COMMAND SET
  - CHIP DIAGNOSTIC
  - RECALIBRATE
  - SEEK
- DATA TRANSFER COMMAND SET
  - FORMAT
  - WRITE MULTIPLE SECTORS
  - WRITE WITH DELETED DATA FLAG
  - READ MULTIPLE SECTORS
  - VERIFY
  - READ IDENTIFIER
  - READ TRACK
  - CORRECT DATA
  - PROGRAMMABLE RECORD PROCESSING

OPERATION STRUCTURE



FORMAT OPERATION

FORMAT COMMAND SERVES SEVERAL PURPOSES:
IT DEFINES THE STRUCTURE OF THE SECTOR,
THE LOCATION OF THE ACTUAL DATA AND PROVIDES GAPS
AND SYNC BYTES THAT ALLOW AN INTERVAL FOR ANY
SWITCHING REQUIRED BY THE DRIVE HARDWARE. THESE
INTERVALS IN TURN ALLOW THE IMDC TO COMPENSATE FOR
ANY VARIATIONS IN EITHER THE RECORDING MEDIA AND/OR
THE DRIVE HARDWARE. THE IMDC WITH ITS STRUCTURE
OFFERS A COMPLETE PROGRAMMABILITY BY MEANS OF SOME
FORMAT TABLES LOCATED IN MAIN MEMORY. IN FACT ANY FIELD
THAT COMPOSE THE SECTOR CAN BE PROGRAMMED IN TERM OF
LENGTH AND VALUE. THE IMDC WILL FORMAT THE TRACK
ACCORDING WITH THESE VALUES IN FORMAT TABLE GIVING
TO THE USER A GREAT FLEXIBILITY.

EXAMPLE OF FORMAT TABLES

FLOPPY DISK

00	02
A1	FE (sync field)
80	N3-3
4E	SECTOR LENGTH
00	02
A1	FB (data field)
fill byte	00
4E	N4-5
4E	00
00	N5-1
C2	FC
4E	00

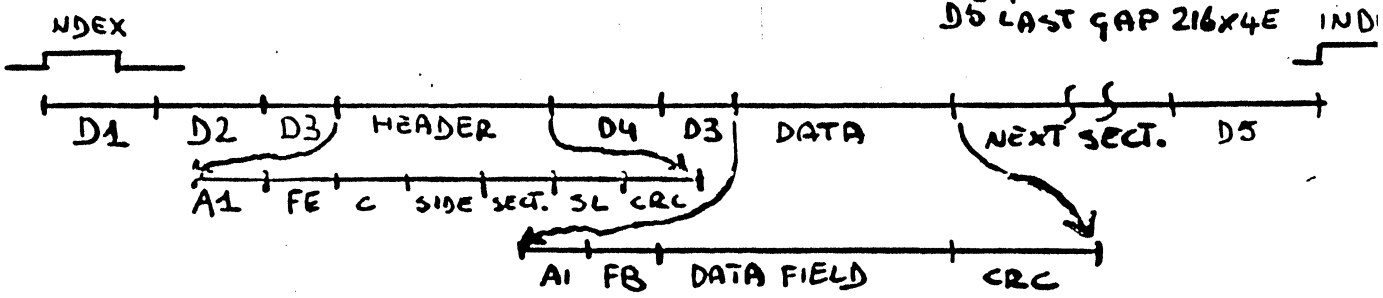
HARD DISK

00	00
A1	FE
00	N3-3
00	SECTOR LENGTH
00	N5-1
A1	FB
fill byte	00
00	N4-2
4E	00
4E	00
00	00
00	00

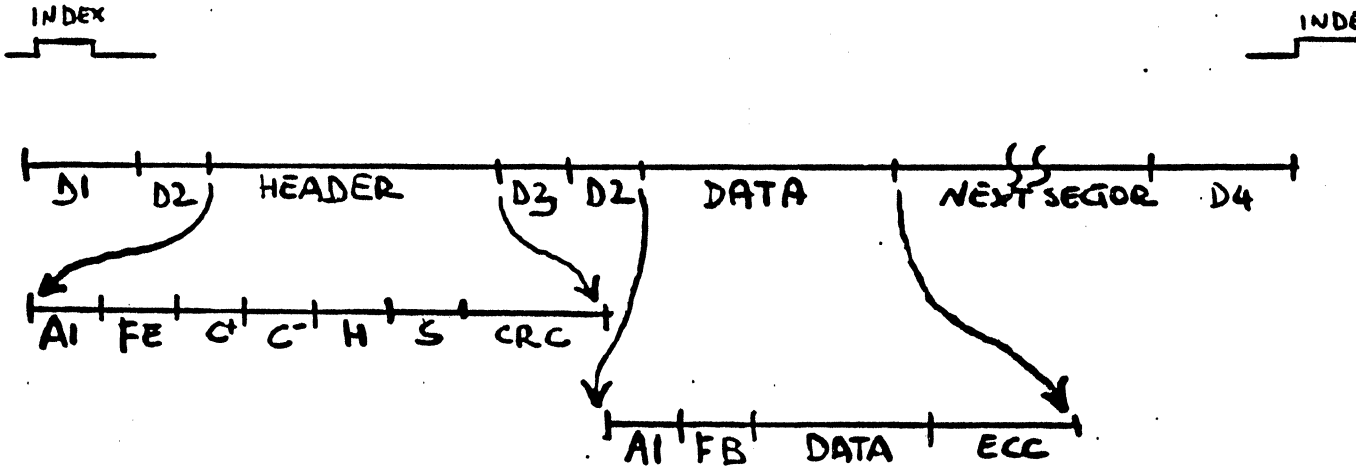
IMDC mfm 128 = 00
256 = 01
512 = 02

FLOPPY FORMAT

D1 POST INDEX GAP 80X4E
D2 INDEX ADDRESS MARK
D3 SYNC 13X00
D4 GAP 22X4E
D5 LAST GAP 216X4E



DISK FORMAT



D1 POST INDEX GAP 22X4
D2 SYNC 13X00
D3 PAD 3X00
D4 LAST GAP 346X4E

ERROR MANAGEMENT

(corrected in new version on INI (Capstone))

THE IMDC PROVIDES AN AUTOMATIC ERROR MANAGEMENT IF REQUIRED. IN FACT IN ECA BLOCK A FIELD IS FORESEEN (MAX # OF RETRIES) TO ALLOW THE HOST TO REQUIRE AUTONATIC RETRIES TO BE EXECUTED. IN CASE OF RETRIES NOT INVOKED THE IMDC WILL STOP THE OPERATION WHEN AN ERROR IS FOUND AND WILL COMPILE THE APPROPRIATE FIELD IN ECA BLOCK (MAIN-EXTENDED STAT TO INFORM THE HOST ABOUT THE ERROR DISCOVERED IN CASE OF RETRIES INVOKED THE IMDC WILL RETRY THE OPERATION FOR THE NUMBER OF TIMES REQUIRED BY THE HOST. IN CASE OF RETRIES NOT SUCCESSFULL THE IMDC WILL STOP THE OPERATION AN NOTIFY THE SW. IN CASE OF SUCCESSFULL RETRIES THE IMDC WILL CONTINUE THE OPERATION UNTIL A NEW ERROR IS FOUND OR THE END OF OPERATION OCCOURS.

THREE KIND OF ERRORS CAN BE FOUND

-) SOFTWARE ERRORS
-) HARDWARE ERRORS
-) MEDIA ERRORS

SOFTWARE ERRORS

IN CASE OF SOFTWARE ERRORS THE IMDC WILL NOT EXECUTE RETRIES INDIPENDENTLY FROM THE MAX NUMBER OF RETRIES VALUE IN ECA BLOCK.

CASE OF SOFTWARE ERRORS

- COMMAND REJECTED
  DUE TO UNKNOWN COMMAND
- DRIVE NOT READY
  THE SPECIFIC DRIVE IS NOT READY
- COMMAND ABORT
  I/O OPERATION ABORTED BY SW REQUEST
- WRITE PROTECT ON DISKETTE
  WRITE OPERATION CAN NOT EXECUTED DUE TO WRITE PROTEC
- POSITIONING ERROR
  CYLINDER/HEAD NUMBER REQUIRED BY SOFTWARE GREATER THAN THE MAXIMUM ALLOWED
- DELETED ADDRESS MARK
  DELETED DATA IS FOUND ON THE TRACK

HARDWARE ERRORS

IN CASE OF HARDWARE ERRORS THE IMDC WILL EXECUTE RETRIES REGARD TO THE MAX NUMBER OF RETRIES IN ECA BLOCK.

CASE OF HARDWARE ERRORS

- **WRITE FAULT**
  WRITE FAULT CONDITION EXISTS ON DRIVE
- **FIFO OVERRUN/UNDERRUN**
  THIS ERROR OCCOURS IF SYSTEM BUS IS LOST FOR A LONG PERIOD OF TIME
- **DATA PART TIME OUT**
  THIS ERROR OCCOURS IF THE IMDC LOST THE SYNCRONIS.. WITH THE TRACK
- **POSITIONING ERROR**
  POSITIONING ERROR DUE TO FAILED SEEK ON DRIVE
- **BUS ERROR**
  SYSTEM BUS ERROR CONDITION OCCOURS

MEDIA ERRORS

FOR THESE KINDS OF ERRORS SPECIAL ACTION SHOULD BE TAKEN BY THE SOFTWARE REGARD TO THE KIND OF SOFTWARE OPERATION

- CRC/ECC ERROR ON DATA FIELD

THIS ERROR OCCOURS IF A CRC/ECC ERROR IS DETECTED ON DATA FIELD.

- IN CASE OF NORMAL I/O THE SW SHOULD TRY TO CORRECT THE ERROR INVOKING THE ECC CORRECTION
- IN CASE OF MEDIA VERIFICATION THIS ERROR MUST PRODUCE A BAD SECTOR

- CRC ERROR ON HEADER FIELD

THIS ERROR OCCOURS IF A CRC ERROR IS DETECTED ON HEADER FIELD

- IN CASE OF NORMAL I/O THE SW SHOULD TAKE DECISION ABOUT THE I/O ERROR
- IN CASE OF MEDIA VERIFICATION THIS ERROR MUST PRODUCE A BAD SECTOR

- UNCORRECTABLE DATA ERROR

THIS ERROR OCCOURS AFTER A CORRECT COMMAND WHEN THE ERROR IS UNCORRECTABLE FROM THE ECC ALGORITHM

CRC - ECC POLYNOMIALS

THE CRC USED ON BOTH FLOPPY HEADER AND DATA FIELDS AND HEADER FIELD ON DISK IS THE SAME.
DCS USES STANDARD CRC POLYNOMIAL 2 BYTES WIDE.

$$X^{16} + X^{12} + X^{5} + 1$$

THE ECC USED ON DISK DATA FIELD IS THE GLOVER ECC 4 BYTES LONG.

$$X^{32} + X^{28} + X^{26} + X^{19} + X^{17} + X^{10} + X^{6} + X^{2} + 1$$

USING THIS POLYNOMIAL THE DCS IS CAPABLE TO CORRECT UP TO 11 BIT IN ERROR CONTINOUS.

ECC HANDLING

THE DCS SUPPLIES THE CAPABILITY TO CORRECT A DATA ERROR WITH A MINIMUM INTERVENTION FROM USER POINTOF

- DESCRIPTION OF ECC PROCEDURE -

IN CASE OF DATA ERROR THE IMDC AT THE END OF OPERATION DISCHARGES IN ECA BLOCK THE SYNDROME COMPUTED DURING THE READ OPERATION THAT GAVE ERROR. WHEN THE SOFTWARE RECOGNISE THE ERROR IT SHOULD TRY TO CORRECT THE ERR IN ORDER TO HAVE THIS CAPABILITY SOFTWARE WILL SEND TO THE IMDC A CORRECT COMMAND RELATED TO THE CHA THAT GAVE ERROR. THE IMDC TAKE IN ACCOUNT THE SYNDROM PREVIOUSLY LOADED IN ECA BLOCK AND BEGIN THE CALCULATIC IN ORDER TO DETERMINE IF THE ERROR IS A CORRECTABLE TYPE AND IF YES COMPUTE THE DISPLACEMENT AND THE ERROR BURST TO ALLOW THE CORRECTION. AN INTERRUP WILL NOTIFY THE SW ABOUT THE END OF COMPUTATION SETTING THE INFORMATION ABOUT THE TYPE OF ERROR CORRECTABLE OR NOT. IN CASE OF CORRECTABLE ERROR THE HOST WILL TAKE IN ACCOUNT THE DISPLACEMENT AND THE ERROR BURST IN ORDER TO CORRECT THE ERROR IN MAIN MEMORY.

BAD SECTOR

THE DCS THROUGH THE IMDC OFFERS THE CAPABILITY TO HANDLE BAD SPOT ON DISK IN A VERY EASY MANNER. IN FACT WHEN A BAD SPOT IS FOUND DURING MEDIA VERIFICATION THE HOST SHOULD REPLACE THAT SECTOR WITH AN ALTERNATIVE ONE. IN ORDER TO FORMAT A BAD SECTOR THE HOST SHOULD PRESET IN THE FORMAT BUFFER THE CORRECT IMAGE OF THE TRACK INCLUDING ALSO POSSIBLE BAD SECTOR. THE SW WILL ALSO DECLARE THE ALTERNATIVE SECTOR WRITING THIS INFORMATION IN THE FORMAT BUFFER.

EX. OF FORMAT BUFFER WITH SECTOR 1 BAD AND ITS ALTERNATINE:

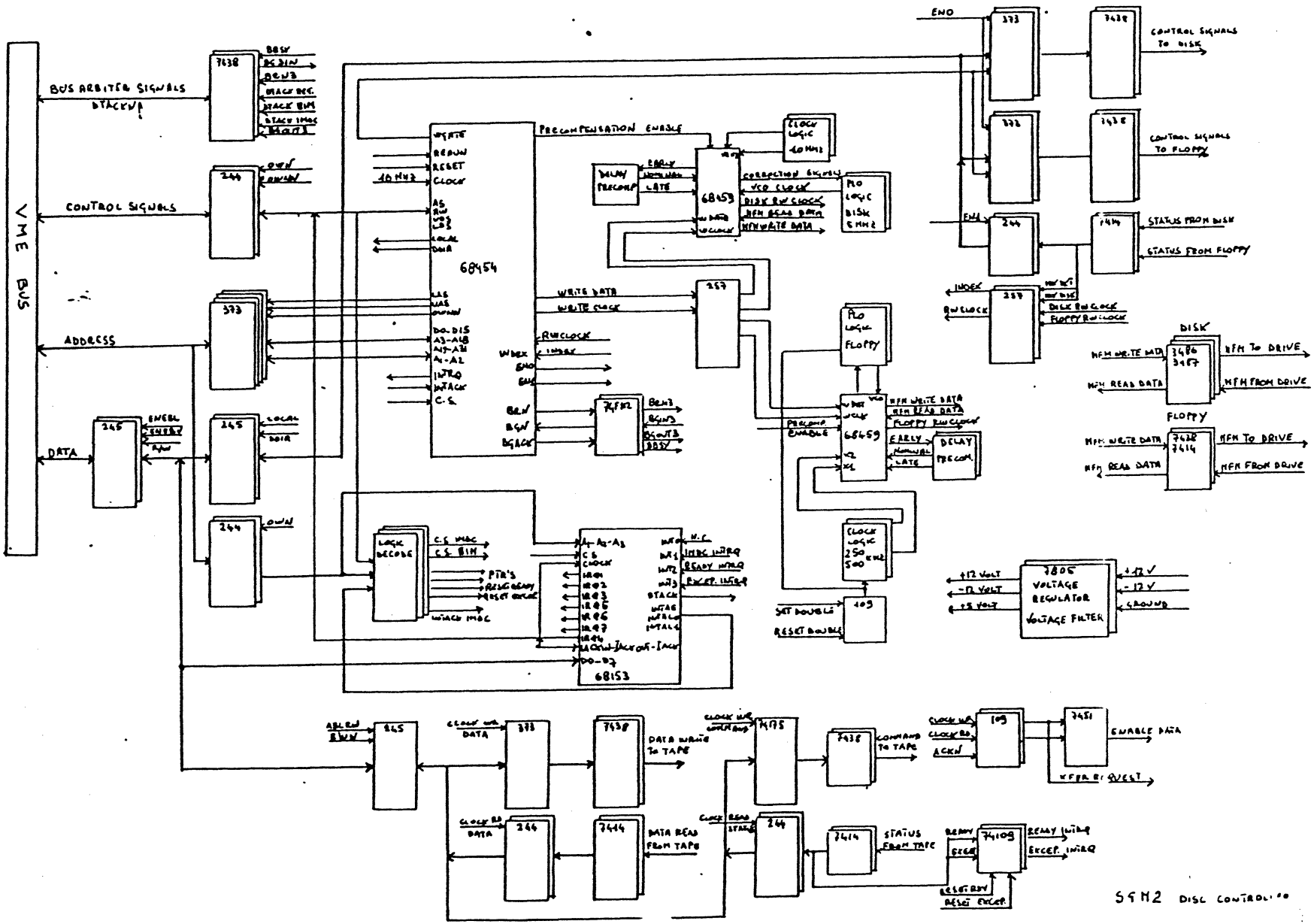
C^+	C^-	H	S	C^+	C^-	H	S	C^+	C^-	H	S	C^+	C^-	H	S	→ etc.
00	10	00	00	00	10	80	01	02	00	00	0X	00	10	00	02	

-BAD SECTOR HANDLING DURING NORMAL I/O-

DURING NORMAL READ/WRITE OPERATIONS THE IMDC WILL TAKE IN ACCOUNT THE INFORMATION WRITTEN IN THE HEADER FIELD AND IN CASE OF BAD SECTOR WILL AUTOMATICALLY REACH THE ALTERNATIVE SECTOR, WILL OPERATE THAT SECTOR AND WILL CONTINUE THE OPERATION RESTARTING FROM THE BAD SECTOR + 1.

OVERLAPPING FEATURES

THE OVERLAPPING FUNCTIONALITY IS VERY IMPORTANT FROM A PERFOMANCES POINT OF VIEW. THE DCS THROUGH THE IMDC OFFERS THE MAXIMUM OVERLAPPING POSSIBLE.
IN FACT WE CAN HAVE UP TO 4 I/O OPERATIONS RUNNING CONTEMPORARY ON DISK/DKT DRIVES IN TERM OF 3 CARRIAGE MOVEMENT, 1 DATA TRANSFER AND AN OPERATION ON TAPE DRIVE.
IT IS IMPORTANT TO NOTE THAT HOWEVER ONLY ONE DATA TRANSFER CAN RUN AT DISK/DKT LEVEL BUT A DATA TRANSFER ON TAPE CAN BE OVERLAPPED WITH THE PREVIOUS ONE ON DISK.



SYM2 DISC CONTROLLER

Signetics

SCN68454
Intelligent Multiple Disk Controller (IMDC)

Preliminary Specification

Microprocessor Products

2

DESCRIPTION

The SCN68454 Intelligent Multiple Disk Controller (IMDC) provides the traditional and advanced features required to control Winchester type rigid disks and floppy disks. It can control up to four rigid or floppy disk drives, in any combination. It can be used to control, with a minimum of external hardware, any drive that has an SA1000 interface standard or an ST506 Seagate interface standard.

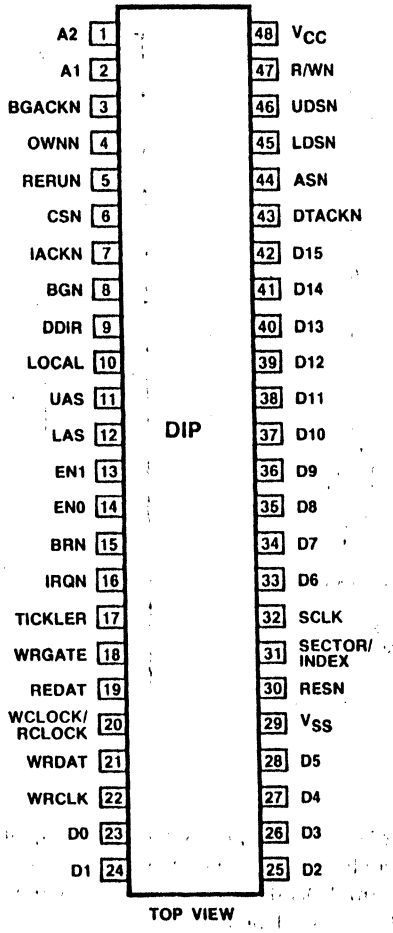
The IMDC supports soft or hard sectored disk track format and standard IBM track formats for floppy disks (both single and double density).

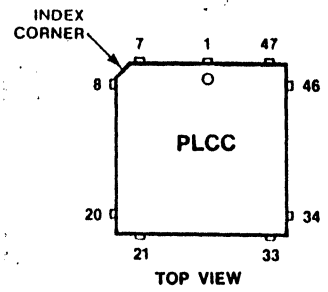
The controller is programmable via an external host processor by the use of high level commands. Data transfer on the host data bus can be 8 or 16-bit in parallel. The IMDC is capable of handling a serial data rate of up to 10Mbits per second. The SCN68454 is constructed using Signetics MOS-VLSI technology.

FEATURES

- **Bus compatible with SCN68000 microprocessor**
- **Automatic rerun on bus error**
- **Supports SCN68000 vectored interrupts**
- **31-bit address counter**
- **16/8-bit data transfers**
- **Supports up to 4 rigid disks and floppy disks in any combination**
- **Supports SA1000 and ST506 Winchester interfaces**
- **Data rates up to 10Mbits per second for MFM**
- **Data rates up to 2Mbits per second for FM**
- **Handles FM and MFM data encoding/decoding**
- **On chip DMA controller and FIFO buffer (128 bytes)**
- **Multiple sector read/write with implied seek**
- **Automatic bad sector handling**
- **32 and 40-bit ECC programmable polynomials**
- **Supports computer generated ECC polynomials**

PIN CONFIGURATION





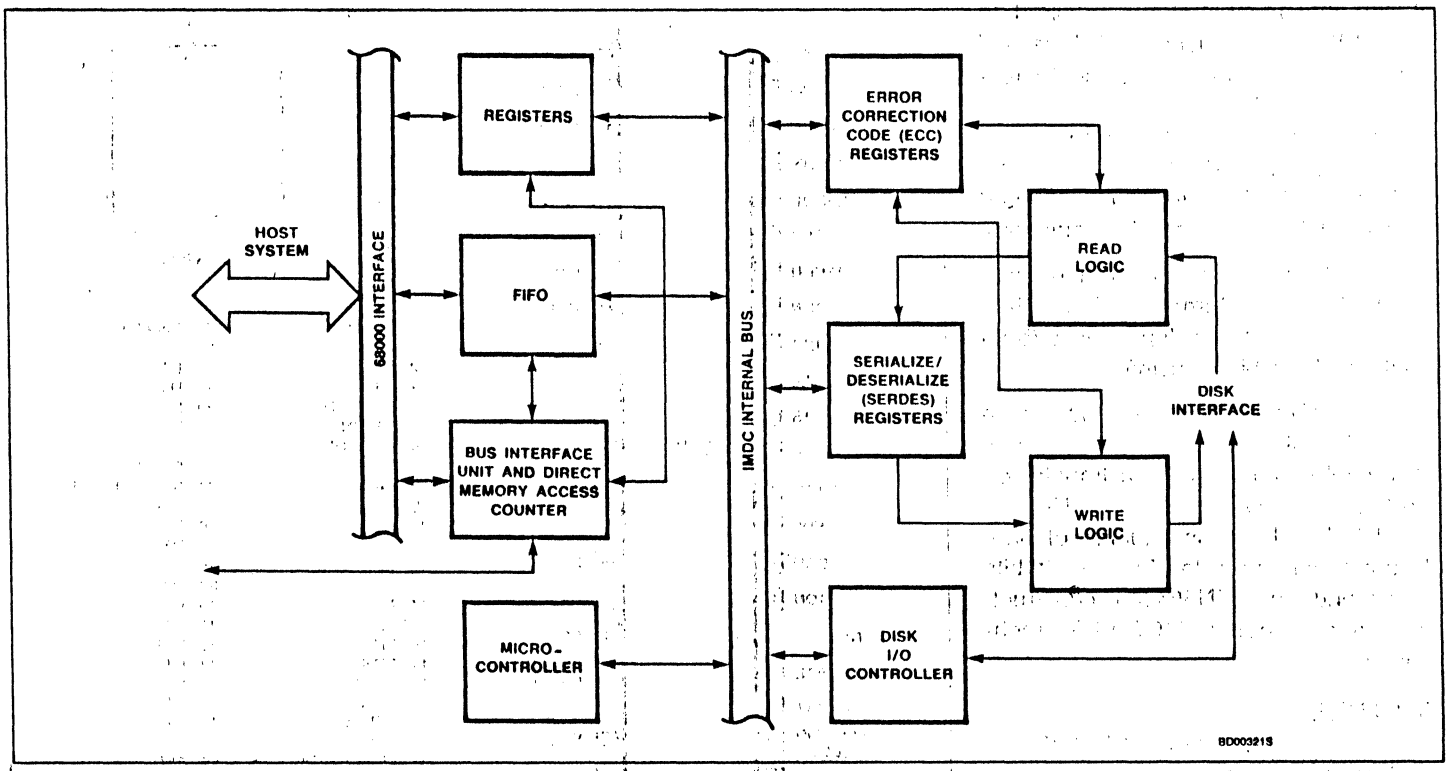
Pin	Function	Pin	Function
1	NC	27	NC
2	A2	28	D2
3	A1	29	D3
4	BGACKN	30	D4
5	OWNN	31	D5
6	RERUNN	32	V_{SS}
7	CSN	33	RESN
8	IACKN	34	SECTOR/INDEX
9	BGN	35	SCLK
10	DDIR	36	D6
11	LOCAL	37	D7
12	UAS	38	D8
13	LAS	39	D9
14	NC	40	NC
15	EN1	41	D10
16	EN0	42	D11
17	BRN	43	D12
18	IRQN	44	D13
19	TICKLER	45	D14
20	WRGATE	46	D15
21	REDAT	47	DTACKN
22	WCLOCK/RCLOCK	48	ASN
23	WRDAT	49	LDSN
24	WRCLK	50	UDSN
25	D0	51	R/WN
26	D1	52	V_{CC}

Intelligent Multiple Disk Controller (IMDC) SCN68454

ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0 to 70°C
Ceramic DIP	SCN68454C6I48
Plastic DIP	SCN68454C6N48
Plastic LCC	SCN68454C6A52

BLOCK DIAGRAM



PIN DESCRIPTION

The pin description table describes the function of each of the pins of the IMDC. Signal names ending in 'N' are active low. All other signals are active high. In the descriptions, 'REG mode' refers to the state when the IMDC is chip selected. The term 'DMA mode' refers to the state when the IMDC assumes ownership of the bus. The term 'LOCAL mode' refers to the state when the IMDC is transferring data to or from the disk interface. The IMDC is in the 'IDLE mode' at all other times.

In this data sheet, signals are discussed using the terms 'active' and 'inactive' or 'asserted' and 'negated' independent of whether the signal is active in the high (logic one) state or the low (logic zero) state. Refer to the individual pin descriptions for the definition of the active level of each signal.

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
A1, A2	1, 2	2, 3	I/O	**Address Lines:** Active high, three-statable. In the REG mode, these low order address lines specify which internal register of the IMDC is being accessed. In DMA mode, A1 – A2 are outputs which provide the low order address bits of the location being accessed. Three-stated in IDLE and LOCAL mode.
D0 – D15	23 – 28, 33 – 42	25, 26, 28 – 31, 36 – 39, 41 – 46	I/O	**Data Lines:** Active high, three-statable. In REG mode, the bidirectional data lines are used to transfer data between the CPU and the IMDC registers. In LOCAL mode, the bidirectional data lines are used to transfer data between the IMDC and the disk unit (three-stated in IDLE mode). In DMA mode, the data lines carry the address information. During the first part of the cycle, D0 – D15 provide high order address bits, A19 – A31, which are latched by UAS. The data lines then provide A3 – A18 address bits which are latched by LAS.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO. DIP	PIN NO. PLCC	TYPE	NAME AND FUNCTION
ASN	44	48	I/O	**Address Strobe:** Active low, three-statable. In REG and IDLE modes, ASN is an input which indicates that the current bus master has placed a valid address on the bus. It is monitored by the IMDC during bus arbitration to ascertain that the previous bus master has completed the current bus cycle. In DMA mode, it is an output indicating that the IMDC has placed a valid address on the bus.
UDSN	46	50	I/O	**Upper Data Strobe:** Active low, three-statable. In REG and IDLE modes, UDSN is an input which indicates that the upper data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning. In an eight bit system this input is tied to DSN.
LDSN	45	49	I/O	**Lower Data Strobe:** Active low, three-statable. In REG and IDLE modes, LDSN is an input which indicates that the lower data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning. In an eight bit system this input is tied to A0.
R/WN	47	51	I/O	**Read/Write:** Active high for read, low for write, three-statable. In REG mode, R/WN is an input which controls the direction of data flow through the IMDC's input/output data bus interface and through an external data bus buffer. R/WN high causes the IMDC to place the data from the addressed register on the data bus, while R/WN low causes the IMDC to accept data from the data bus. In DMA mode, R/WN is an output to memory and I/O controllers indicating the type of bus cycle. It is held three-stated during IDLE and LOCAL mode.
CSN	6	7	I	**Chip Select:** Active low. When low, places the IMDC into the REG mode. This input signal is used to select the IMDC for register data transfers. These transfers take place over the D0 – D15 lines as controlled by the R/WN and A1 – A2 inputs. The IMDC is deselected when CSN is high. CSN is ignored during DMA mode.
DTACKN	43	47	I/O	**Data Transfer Acknowledge:** Active low, three-statable. In REG mode, DTACKN is asserted on a write cycle to indicate that the data on the bus has been latched, and on a read cycle or interrupt acknowledge cycle to indicate that valid data is present on the bus. The signal is negated (driven high) when completion of the cycle is indicated by negation of the CSN or IACKN input. In DMA mode, DTACKN is an input monitored by the IMDC to determine when the addressed device (memory) has latched the data (write cycle) or put valid data on the bus (read cycle).
RERUNN	5	6	I	**Rerun:** Active low. This input is asserted by external error detect logic to indicate a bus error. In DMA mode, the IMDC stops operation and three-states the data, address, and control lines, except BGACKN. It remains IDLE until RERUNN becomes inactive, and then retries the last bus cycle. If RERUNN is asserted again, the IMDC sets the error code in the main status byte, stops DMA operation, releases the bus, and interrupts the CPU. Not monitored in REG, LOCAL and IDLE modes.
RESN	30	33	I	**Master Reset:** Active low. Assertion of this pin clears the internal registers and initializes the interrupt vector register to H'0F'. All bidirectional I/O lines are three-stated and the IMDC is placed in the IDLE mode.
SCLK	32	35	I	**Clock:** Active high. Usually the system clock, but may be any clock meeting the electrical specifications. Used by the IMDC to synchronize disk functions and external control lines, and may not be gated off at any time. The frequency should be 16MHz ± 1%.
IRQN	16	18	O	**Interrupt Request:** Active low, open drain. This output is asserted at the end of each command execution. The CPU can read the status register to determine the interrupting condition, or can respond with an interrupt acknowledge cycle to cause the IMDC to output an interrupt vector on the data bus.
IACKN	7	8	I	**Interrupt Acknowledge:** Active low. When asserted, indicates that the current cycle is an interrupt acknowledge cycle. The IMDC normally responds by placing the contents of the interrupt vector register on the data bus and asserting DTACKN.
BRN	15	17	O	**Bus Request:** Active low, open drain. BRN is asserted by the IMDC to request ownership of the bus for a DMA transfer. It is negated when the bus has been granted (BGN low) and BGACKN has been asserted.
BGN	8	9	I	**Bus Grant:** Active low. BGN indicates to the IMDC that it is to be the next bus master. After BGN is asserted, the IMDC waits until DTACKN, ASN, and BGACKN have become inactive before assuming ownership of the bus by asserting BGACKN.
BGACKN	3	4	I/O	**Bus Grant Acknowledge:** Active low, open drain. As an input, BGACKN is monitored by the IMDC during the bus arbitration cycle to determine when it can assume ownership of the bus (BGACKN negated). In DMA mode, it is asserted by the IMDC to indicate that it is the bus master. Three-stated in REG, LOCAL, and IDLE modes.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO. DIP	PIN NO. PLCC	TYPE	NAME AND FUNCTION
EN0	14	16	O	**Enable 0:** Active high. This signal is asserted during LOCAL mode when the IMDC transmits disk control signals on the data lines D0 – D15. The assertion or negation of EN0 is used to latch the control signals at the disk interface.
EN1	13	15	O	**Enable 1:** Active low. This signal is asserted during LOCAL mode when the IMDC reads the status of the disk drive. The assertion of EN1 is used to enable the status information onto data lines D0 – D15.
UAS	11	12	O	**Upper Address Strobe:** Active high. UAS is active only during DMA mode. It is used to latch the upper address bits A19 – A31 from the address/data lines.
LAS	12	13	O	**Lower Address Strobe:** Active high. LAS is active only during DMA mode. It is used to latch the lower address bits A3 – A18 from the address/data lines.
REDAT	19	21	I	**Composite Read Data:** This signal is the composite disk data synchronized to the RCLOCK signal generated by the external PLL.
WRGATE	18	20	O	**Write Gate:** Active high. This signal is asserted during disk write operations. Disasserted during all other modes.
WCLOCK/ RCLOCK	20	22	I	**Write Clock/Read Clock:** This clock input is generated by external logic such as the SCB68459 DPLL. During disk write operations, the input is defined as WCLOCK which provides the bit-cell frequency to the IMDC. The input is defined as RCLOCK during disk read operations. RCLOCK is twice the data frequency and is synchronized to REDAT.
WRDAT	21	23	O	**Write Data Pattern:** Active high. This signal provides the write data pattern for external logic, like the SCB68459 DPLL, to use with WRCLK and WCLOCK to generate the write data pulse stream to a disk unit. WRDAT is an non-return to zero (NRZ) signal which changes state on the rising edge of WCLOCK.
WRCLK	22	24	O	**Write Clock Pattern:** Active high. This signal provides the write clock pattern for external logic, like the SCB68459 DPLL, to use with WRDAT and WCLOCK to generate the write data pulse stream to a disk unit. WRCLK is an NRZ signal which changes state on the rising edge of WCLOCK.
TICKLER	17	19	O	**Tickler:** Active high. This signal is used to control an external PLL. When TICKLER is asserted, the external PLL should output the crystal controlled clock to the WCLOCK/ RCLOCK input. With TICKLER is low, the PLL should synchronize on the incoming disk data and generate read clock to the WCLOCK/RCLOCK input.
SECTOR/ INDEX	31	34	I	**Sector/Index:** Active high. The IMDC uses the rising edge of the signal to generate the read/ write sequence. Hard sectored disk drives output a pulse for each sector boundary for the IMDC. Soft sectored disk drives have a once around index pulse to define the start of a track.
LOCAL	10	11	O	**Local:** Active high. During LOCAL mode, this line controls the output enable on the bidirectional buffers of the address/data lines.
OWNN	4	5	O	**Own:** Active low. This output is asserted by the IMDC during the DMA mode to indicate bus mastership. It can be used to enable external address/data and control buffers. Inactive in REG and IDLE modes.
DDIR	9	10	O	**Data Direction:** Active high. This signal is active during the DMA and REG modes. It controls the direction of the data through the bidirectional buffers on the address/data bus. DDIR is asserted during a read operation of the IMDC.
V_{CC}	48	52	I	**+5 volt ± 5% power input.**
V_{SS}	29	32	I	**Power ground input.**

REGISTERS

Register Map

The IMDC is a memory transfer oriented device with minimal information transferred to the registers. The internal accessible register organization of the IMDC is shown in table 1. Register bit formats are shown in table 2. Each is 8-bits wide to allow interface to either 8 or 16-bit host systems. When the IMDC is interfaced to an eight bit system, A0 is tied to LDSN and the data strobe (DSN) is tied to UDSN.

Interrupt Source Register (ISR)

These bits are used to indicate which drive was the source of a command completion interrupt. Bit 4 reflects drive 0 as the source and bit 7 is for drive 3. The assertion of RESN will initialize all four bits to zero. The IMDC will not initiate the next command, if pending, until the host resets the interrupt source bit. The host should also reset the appropriate busy bit in the status and configuration register.

Drive Status and Configuration Register (DSCR)

[0]8/16 Bit Mode

This bit sets the length of memory and register data transfers. Byte transfers are initiated when bit 0 is set to zero. This bit is set to zero when RESN is asserted.

[7:4]Drive Busy

These bits are used by the host system to initiate an IMDC command operation for a particular drive. Bit 4 is used for drive 0 and bit 7 for drive 3. After the host system has set a busy bit to activate a drive, the IMDC

Table 1. IMDC ADDRESS MAP

ADDRESS BITS[1]	ACRONYM	REGISTER NAME	MODE	AFFECTED BY RESET
2 1 0				
0 0 0	EPH	ECA pointer high	R/W	Yes
0 0 1	EPMH	ECA pointer middle high	R/W	Yes
0 1 0	EPML	ECA pointer middle low	R/W	Yes
0 1 1	EPL	ECA pointer low	R/W	Yes
1 0 0	IVR	Interrupt vector register	R/W	Yes
1 0 1	ISR	Interrupt source register	R/W	Yes
1 1 0[2]	DSCR	Drive status and configuration register	R/W	Yes
1 1 1		Reserved		

NOTES:
1. A0 = 0 for UDSN asserted, A0 = 1 for LDSN asserted for 16- bit mode.
2. In 16-bit systems, the data for this register must be in data bits D0 – D7.

Table 2. REGISTER BIT FORMATS

INTERRUPT SOURCE REGISTER

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DR3 INT	DR2 INT	DR1 INT	DR0 INT	<——— NOT USED ———>			

DRIVE STATUS AND CONFIGURATION REGISTER

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DR3 BUSY	DR2 BUSY	DR1 BUSY	DR0 BUSY	<——— NOT USED ———>			8/16 MODE

responds by accessing the corresponding event control area (ECA) and performing the requested action. The host system can abort the current drive operation by resetting the busy bit before the operation is completed. The assertion of RESN will initialize the busy bits to zero (only time the IMDC writes into this register).

Event Control Area Pointer Registers (EPH/EPMH/EPML/EPL)

These four registers are used by the host system to direct the IMDC to a table of pointers. The table consists of four 2 word addresses that point to the location of the four ECA blocks. The pointers are arranged in ascending order by drive number. EPL is the least significant byte and EPH the most significant byte. All four registers are cleared to zero by RESN asserted.

Interrupt Vector Register (IVR)

The IVR contains the value to be placed on the data bus upon receipt of an interrupt acknowledge from the CPU. The contents of this register are initialized to H'0F' by a reset.

OPERATION

The IMDC is an intelligent controller with on-chip microprogrammed CPU and interface circuitry. Two interfaces are provided, one to the host CPU and the other to the disk drive. The host interface contains a complete DMA interface compatible with the SCN68000 bus. After the IMDC accepts a command, the DMA interface handles all transfers between the system memory and the IMDC. The IMDC signifies through an interrupt signal that the command is completed.

Operation Initiation

The host must initialize the IMDC before it can process any command requests. The initialization information must be passed from the host to the IMDC control registers. The information consists of the data byte or word transfer mode selected in the drive status and configuration register. The host must also load an interrupt vector to the interrupt vector register and load the event control areas registers with the start location of the ECA pointer table in system memory.

A new command is requested using the drive status and configuration register. The user sets a bit corresponding to the drive to be serviced. This causes the IMDC to start execution of the command for the requested drive. This can be performed as long as the bus is available. The IMDC accepts the request but does not necessarily begin to process it. If the IMDC is currently doing disk data transfers to another disk drive, it will accept the request and treat it as a pending request for disk drive service. It does this in order to prevent forcing the host processor to wait for it to complete its current processing. The IMDC will execute parallel seek operations for floppy and SA1000 type disk drives.

Because four different drives can be on line simultaneously, the possibility exists of more than one pending request to appear in the drive status and configuration register. In this situation the IMDC will process the drive requests in ascending order (i.e., 0, 1, 2, and 3) and then start again with drive zero.

It is also the responsibility of the host to insure the drives that are to be controlled are in a ready state prior to issuing commands. Commands issued to drives that are not ready will result in the command being aborted. Also if a command is requested on a drive that goes from ready to not ready, the IMDC will abort the execution of the command. In both cases, an interrupt cycle will be initiated.

When RESN is asserted, the IMDC will go through an internal initialization program which clears the control registers and the interrupt vector will be set to H'0F'. All bus and control lines will be cleared and the IMDC will enter an idle loop.

DMA Operation

After a command is started, it is executed without further communication with the host system. All memory data transfers are handled automatically by the on chip DMA controller. The IMDC indicates that it wishes to become the bus master by asserting its bus request (BRN) output. The processor ac-

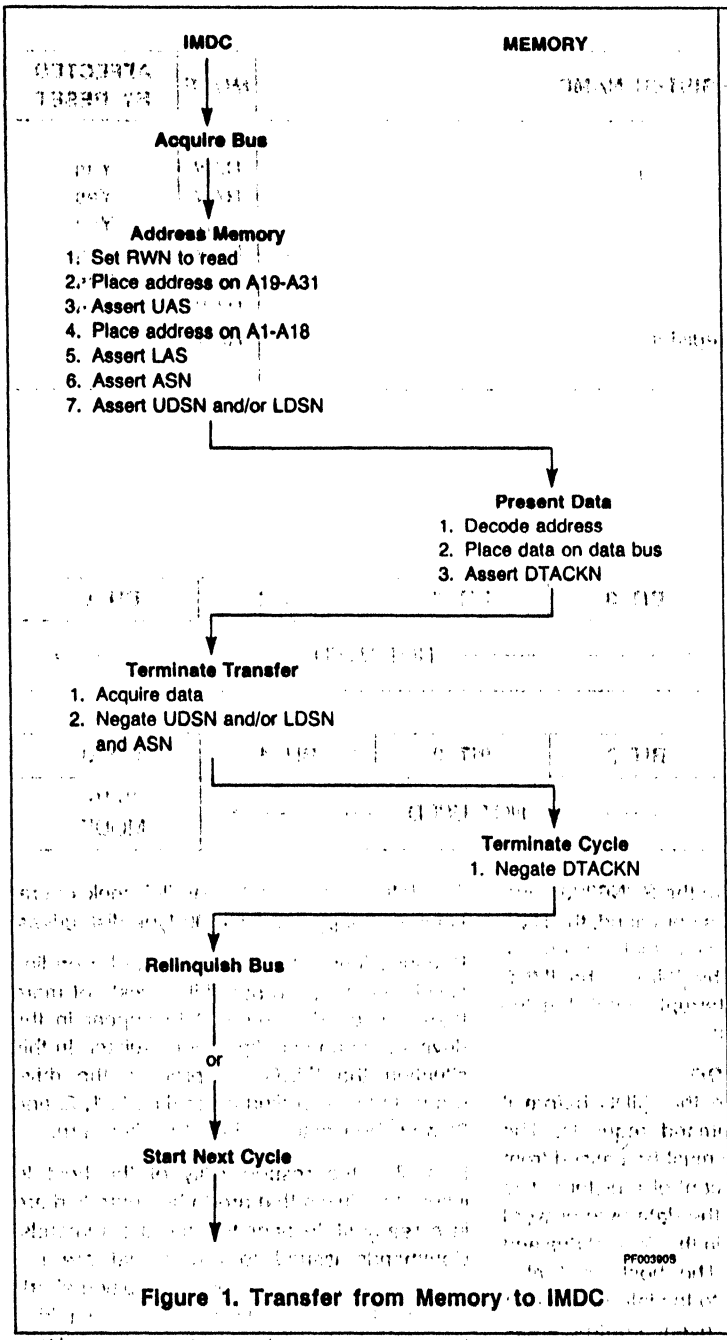


Figure 1. Transfer from Memory to IMDC

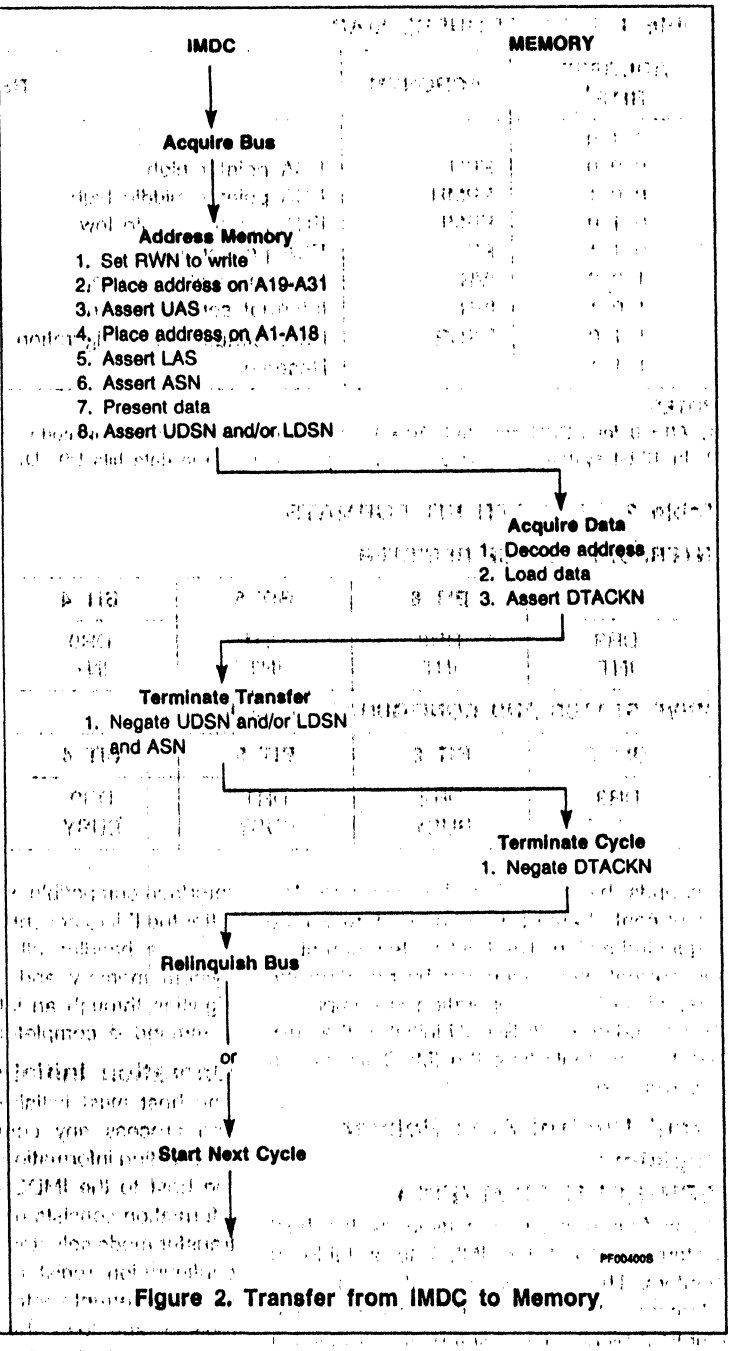


Figure 2. Transfer from IMDC to Memory

knowledges the request by asserting its bus grant (BGN) output which puts the bus up for arbitration.

The IMDC will be the next bus master when its BGN input is asserted. The IMDC then waits for address strobe (ASN), data transfer acknowledge (DTACKN), and bus grant acknowledge (BGACKN) to become inactive. It then asserts the BGACKN output to become the bus master and negates the BRN output. The IMDC then proceeds with the transfer of data between itself and memory. After the data transfer phase, the IMDC relinquishes bus mastership by negating the BGACKN output. Flow charts for the transfer operations are shown in figures 1 and 2. Refer to the timing section for the equivalent timing diagrams.

DMA Transfer Rates

The serial read/write I/O portion of the drive interface is a dedicated slave of the IMDC disk drive interface; that is, a serial string is sent or received by the interface without interruption. The DMA interface, however, must surrender the bus for arbitration after the requested number of operands have been transferred. If the number of transfers does not provide the IMDC with sufficient bus access time, it is conceivable that the DMA may fail to keep track with the disk I/O operation. This situation can occur if the bus is lost for a 'long' period of time. At the IMDC, the serial disk transfer rate is 1.6μsec per word at a 10MHz data rate. The IMDC processor operates using a 312.5nsec period clock. Three cycles or 937.5nsec are required to perform one DMA transfer. If the system bus is unavailable to the IMDC DMA for an average time greater than the difference between the drive and DMA transfer rates (662.5nsec), overflow or underflow of the FIFO buffer can result.

As an example, consider a Winchester disk transfer data of 5Mbits/second. If the transfer count is set to 16 words per transfer, the IMDC will transfer each 32 byte block in 15 microseconds. The transfer time will be 51.2 microseconds for the disk data to put 32

bytes into the FIFO. The difference between the two transfer block rates is 36.2 microseconds. This is the maximum time which the IMDC does not require the system bus.

To avoid this situation the DMA must not lose the bus for a period exceeding the difference in transfer rates. This can be accomplished in a variety of ways. One approach is to allow the DMA to have the bus mastership for the entire sector transfer. A second approach is delay the DMA until a sector has been read into the FIFO buffer and to delay writing to the disk until the FIFO is filled with one sector of data. The first approach severely restricts the availability of the system bus to other devices.

The latter approach requires a minimum FIFO buffer equal to the size of a sector and introduces a delay into the IMDC operation.

As a result, both approaches place stringent limitations on the system and reduce the throughput of the device.

Another alternative is to assign the bus arbitration such that the IMDC always has highest priority and that the bus must be returned in a time segment shorter than the transfer rate difference. This latter solution requires external circuitry to implement and still does not guarantee the IMDC bus access time is sufficient. To provide the user with maximum system flexibility, without severely restricting the bus to other devices, the IMDC has been structured to allow the user to specify the number of operands that the DMA can be master of the bus. The ECA Command Option field permits the user to select the number of operands that can be transferred before the bus is returned for arbitration. This technique allows the user to customize the DMA operation to the needs of the system.

I/O DESCRIPTION

The disk interface consists of two sections. The input and output ports that sense and generate slow changing or static control signals and the serial read/write data section. The parallel data is transferred over the address/ address/data lines (A3/A19/ D0 – A18/A31/D15) during LOCAL mode. Table 3 shows the data bus assignment during an input port bus cycle and table 4 shows the output port bus cycle assignment.

Table 3. INPUT PORT DEFINITION

BUS LINE	SIGNAL NAME	DEFINITION
D0	INDEX	Rigid disk index signal (hard sector mode) must be greater than 500ns
D1	READY	Drive ready
D2	SEEKC	Seek completed
D3	TR0A	Track zero first signal
D4	TR0B	Track zero second signal
D5	WFA	Write protection for floppies
D6	WFB	Write fault
D7 – D15		Not used

Table 4. OUTPUT PORT DEFINITION

BUS LINE	SIGNAL NAME	DEFINITION
D0	HEAD1	Head select 2**0/side select for floppy disks set to '0' for single sided floppy disks
D1	HEAD2	Head select 2**1
D2	HEAD4	Head select 2**2 that can be transferred.
D3	HEAD8	Head select 2**3
D4	HEAD16	Head select 2**4
D5		Not used
D6	STEP	Step pulse 10μsec long (period set in ECA)
D7	DIR	Direction of head movement (a high corresponds to head movement toward the disk spindle)
D8	LWC	Low write current
D9	MOT	Motor on
D10	PRECOM	Precompensation enable
D11	HDL	Head load for floppy disks
D12	SEL0	Drive 0 select
D13	SEL1	Drive 1 select
D14	SEL2	Drive 2 select
D15	SEL3	Drive 3 select

The two signals enable 0 (EN0) and enable 1 (EN1) are used to gate the signals off or onto the bus. EN0 is used to latch the bus during the output cycle and EN1 is used to enable the input data onto the bus.

Input Port

The input port consists of eight lines to input disk status information to the IMDC. All signals are active high. To accommodate different drives, signals TROA, TROB, WFA and WFB are used. The IMDC will respond to any one of the signals which becomes active high. The drive will not activate these signal simultaneously.

Output Port

The output port consists of the signals corresponding to bus lines D0 through D15. All signals are positive logic. A practical implementation of the output port would consist of standard octal registers.

EVENT CONTROL AREAS

The host system communicates with the IMDC through the event control areas (ECAs) which reside in system memory. An ECA parameter block is set up for each disk drive (up to four) to be controlled. These areas contain information that is required by the IMDC to execute a disk command. This information includes data about the requested command and the disk drive. The host prepares for IMDC operation by loading the ECA pointer table in system memory with the address of each of the ECA blocks. The disk drive to ECA block assignment is determined by the relative position of the pointer in the table. The first table entry corresponds to drive zero, the second entry to drive one, etc. (see figure 3).

The IMDC microprogram will use the data contained in the ECA block to generate the disk interface signals and perform the requested drive I/O. Prior to informing the host of a command completion, the

IMDC writes the return status information to the appropriate ECA block memory. Table 5 describes the format of the ECA block.

The communications between the IMDC and the host are established through the ECA. Alteration of the ECA by the host after a command has been accepted by the IMDC is not allowed.

The static and dynamic (see ECA fields) ECA command parameters should be verified by the host before new command execution is requested. During execution of a command involving multiple sectors, it is possible that dynamic values generated by the IMDC could be invalid. The host does not have access during this time to alter or even check these values, therefore, it is the responsibility of the

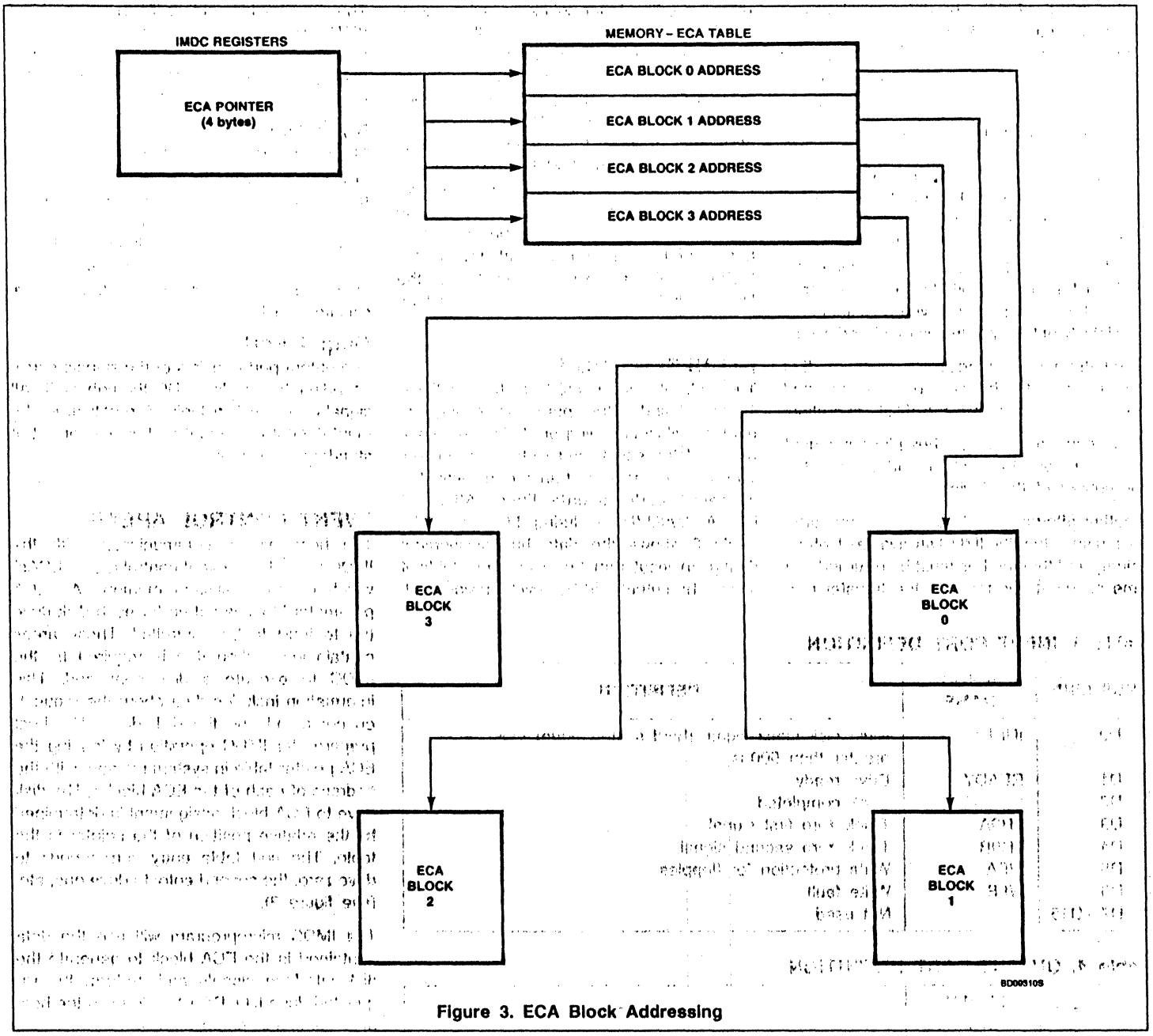


Figure 3. ECA Block Addressing

host to not only guarantee the integrity of the initial parameters but also to insure that values calculated by the IMDC during the life of a command remain valid. For example, the IMDC cannot know that a calculated DMA address is valid until it attempts to transfer data.

Implied in the execution of any disk operation is reading of the ECA data by the IMDC to load the drive control parameters. At the conclusion of a disk operation, the IMDC performs a write operation to the corresponding ECA parameters to store the results of the command. Both reading and writing of the ECA fields by the IMDC utilize the host/DMA interface to arbitrate for the system bus and to perform the required handshaking.

ECA FIELDS

As an IMDC command proceeds through its execution; it references and alters various fields of the ECA. To gain a better understanding of the interactions between the IMDC and the various ECA fields, the ECA data can be separated into four general categories.

1. Command, command status, and execution parameters
2. Programmable record processing fields
3. Disk Format Fields
4. Drive Control Parameters

These categories differentiate the ECA fields not only in content but in terms of the access and alterability.

Command, Command Status and Execution Parameters

These parameters consist of the ECA data fields required by the IMDC to execute a command. The fields can be further separated into static fields and dynamic fields. The static fields are valid for the duration of a command; the IMDC does not alter any information in these fields as long as the command is being processed. The dynamic fields provide the 'local storage' needed to execute multiple sector commands. The IMDC uses the dynamic fields of the ECA to maintain the current execution status of the on-line disk drives.

Table 5. ECA BLOCK FORMAT

WORD NO.	15 8	7 0	
00	Command code	Main status	
01	Extended status		
02	Max # of retries	Actual # of retries	
03	DMA count	Command options	
04	Buffer address most significant word		
05	Buffer address least significant word		
06	Buffer length request		
07	# of bytes transferred		
08	Cylinder number		*
09	Head number	Sector number	*
10	Current cylinder position		
11	PRP command control word		**
12	Location of SCWT, most significant word		**
13	Location of SCWT, least significant word		**
14	Scan terminator	Reserved	**
15	Maximum record length – 1		**
16	N0 – Pre index gap	N1 – Post index gap	***
17	N2 – Sync byte count	N3 – Post ID gap	***
18	N4 – Post data gap	N5 – Address mark cnt	***
19	Reserved		***
20 – 22	ECC mask (3 words)		
23	Motor on delay	# of heads	
24	Ending sector #	Stepping rate	
25	Head setting time	Head load time	
26	Seek type	Phase count	
27	Low write Current boundary track		
28	Precompensation boundary track		
29 – 31	ECC remainder (3 words)		****
32	Maximum number of cylinders per surface		
33	First sector/sector length	Flag byte	
34 – 35	B – tree pointer (2 words)		**
36 – 45	IMDC working area 10 words		

* Physical starting sector number.
** Programmable record processing parameters.
*** Track format fields.
**** ECC remainder will be aligned to the MSB byte of this field.

Command Code (1 Byte)
The command code indicates the command to be executed. Table 6 lists the valid IMDC commands and their associated codes. For a complete description of each command refer to the command description in the Command section.

Main Status (1 Byte)
This field contains the general oriented status information about the command execution. This encoded byte is updated at the completion of the current command and before the interrupt is asserted. Table 7 shows the possible values which may be returned by the IMDC.

Extended Status (2 Bytes)
This field contains specific bit oriented status information about the command execution. If errors result during the execution, the corresponding bit will be ORed into extended status and kept there. This OR function can be used by the host system for error logging. The extended status will be reset to zero by

Table 6. COMMAND CODES

COMMAND MNEMONIC	HEX CODE	DESCRIPTION
WRMS	00	Write multiple sector
WRDD	01	Write with deleted data flag
VER	10	Verify
REMS	11	Read multiple sector
PRP	12	Programmable record processing
TSR	13	Transparent sector read
RETD	20	Read identifier
FORM	40	Format
CALB	41	Recalibrate to track zero
CORR	81	Correct data
DIAG	80	Diagnostic

Table 7. MAIN STATUS CODES

DECIMAL VALUE	DESCRIPTION
0	Correct execution without error
1	Irrecoverable error which cannot be completed (auto retries are attempted, see extended status)
2	Drive not ready
3	PRP operation unsuccessful
4 – 5	Not used
6	Command rejected
7 – 9	Not used
10	Command abort (busy bit reset by host)
11 – 255	Not used

Table 8. EXTENDED STATUS

DATA BIT	DESCRIPTION
0	Write fault
1	CRC/ECC error on data or ID
2	FIFO overrun/underrun
3	No identifier found
4	Not used
5	Deleted data address mark
6	Write on write protected diskette
7	Positioning error
8	Data part timeout
9	Not used
10	Uncorrectable data error (ECC)
11	Not used
12	Not used
13	Positioning timeout
14	Not used
15	Bus Error Fault (DMA operation)

the IMDC at the beginning of the command execution. The error definitions are shown in table 8.

Maximum Number of Retries (1 Byte)
This parameter specifies the maximum number of retries per command (not per sector) that the IMDC attempts, after a disk operation error. Loading a zero value indicates that no retries should be attempted. This byte will not be used if a bus error or FIFO under/overrun error occurs during the command execution. These two errors will cause an immediate command abort. If an identifier is not found, 16 revolutions of the disk will be tried before exit.

Actual Number of Retries (1 Byte)
This byte is set by the IMDC to the actual number of retries executed per command. All retries are accumulated by the IMDC on a sector by sector basis.

DMA Count (1 Byte)
This byte contains a DMA transfer count. If a zero value is specified, the IMDC transfers only one operand and surrenders the bus. A transfer amount of 1 to 16 can be specified.

Command Options (1 Byte)
This byte contains options that are to apply to the current command. Valid options for handling of deleted-data address marks are described in table 9. Sectors with the deleted-data address mark (bits 0, 1 of the option field) will be handled as shown for read operations, excluding those associated with the PRP command. Some retries may be necessary in these steps. They all start with a new identifier search.

Buffer Address
This is a 31-bit starting buffer address for the DMA transfers. The LSB of the address field defines the even or odd address of the 16 bit words when in 16-bit mode. For the 68000, only the first 23 bits are used for addressing. The additional bits can be used to implement the function codes. It should be noted that the IMDC does not protect the upper address lines, AD25 – AD31, from counter overflow. It is up to the operator to prevent transfers between the IMDC and memory greater than the 68000 addressing range. For PRP commands, this field contains the pointer to the content of the matched record data (see Programmable Record Processing section).

Buffer Length Requested (2 Bytes)
This field contains the requested number of bytes to be transferred. This field implicitly defines the number of multiple sector transfers to be performed. A seek only will occur when a zero length is specified. For the format track command, this value is used to terminate the operation.

Table 9. COMMAND OPTIONS

BIT	FORMAT	OPTION DESCRIPTION
1–0	00	FM, IBM single density format
	01	MFM, IBM double density format
	10	Programmable disk format, 1 byte cylinder
	11	Programmable disk format, 2 byte cylinder
2		Not used
4–3	00	Select CRC-CITT (CRC-16) polynomial
	01	Select 32 bit ECC polynomial
	10	Select 40 bit ECC polynomial
	11	Not used
5	0	The sectors with deleted-data address mark will be skipped as if it did not exist. A successful CRC/ECC check is not required.
	1	The data of the sector with deleted-data address mark will be transferred to the host system and the operation terminated
6		Not used
7		Hard sectored disk

Actual Number of Bytes Transferred (2 Bytes)
This field indicates how many bytes have been actually read or written during a command execution. The value can be used, for example, for a location of a problem sector with an irrecoverable error. The value of this field can be changed by the IMDC during the execution of a command and used as an intermediate field.

Physical Starting Sector Number (4 Bytes)
Cylinder, head, sector – This field contains the physical disk sector location at which the command is to begin execution. The format of the field is shown below:

Cylinder High	Cylinder Low
Head Number	Sector Number

Current Cylinder (2 Bytes)
These two bytes will be updated by the IMDC after each positioning. At the beginning of the IMDC reset, they should be set by the host system to zero, because most of the hard disks have built-in automatic recalibration.

Programmable Record Processing Fields (10 Bytes)

The programmable record processing fields are described in the Programmable Record Processing section.

Disk Track Format Fields (8 Bytes)

The disk track format fields are described in the Disk Track Format section.

ECC Mask (6 Bytes)
These six bytes define the error correction polynomial. The standard notation of a polynomial is: $X^{40} + k_{39} X^{39} + \cdots + k_1 X^1 + k_0 X^0$

The coefficients k_{39} through k_0 may be any combination of 1s and 0s. For a given polynomial, an equal combination of bits in the mask register will initialize the IMDC logic to generate or check the same polynomial on the disk.

Drive Control Parameters

These fields contain information that define the disk interface to the IMDC. This information is provided by the host system and is not altered by the IMDC. The IMDC microprogram, that controls the disk drive I/O issued, continually references this information during the execution of a command. Because of the potentially large number of possible variations of interfaces, the IMDC cannot provide extensive verification of the parameters contained in these fields. Protection and verification of these fields are primarily the responsibility of the a host operating system and not the IMDC.

Motor on Delay (1 Byte)
This field contains the motor on delay (period) in ten millisecond units. A value of zero represents a zero delay timeout.

Number of Heads (1 Byte)
This value is the number of heads on the disk unit. The maximum value is 128.

Ending Sector Number (1 Byte)
This value is the last sector on the track (cylinder) for the disk unit.

Stepping Rate (1 Byte)
This field contains head stepping rate (period) in 500μsec units if applicable.

Head Settling Time (1 Byte)
This field contains head settling time in 500μsec units. If a non-zero head settling time is specified, the IMDC assumes that a seek complete is not available from the drive.

Head Load Time (1 Byte)
This field contains head load time in 500μsec units. For all hard disks, this value must always be set to zero by the host system.

Seek Type (1 Byte)
This field defines what type of seek positioning is to be performed.

VALUE	DESCRIPTION
0	Normal single step seek
1	ST506 with accelerated seek
2	Disk with buffered seek

Phase Counter (1 Byte)
This field contains the phase counter which is a status of the command execution set by the IMDC.

Low Write Current Active Track (2 Bytes)
This field defines, to the IMDC, the track at which low write current (LWC) output signal, on the output port (bit 8), is to be asserted.

Precompensation Active Track (2 Bytes)
This field defines to the IMDC the track at which precompensation signal (PRECOM) is to be asserted at the output port (bit 10).

ECC Remainder (6 Bytes)
This field contains the ECC remainder generated by the IMDC read operation. This returned value will be zero unless an error is detected.

Maximum Number of Cylinders (2 Bytes)
These bytes define, to the IMDC, the maximum cylinder count for the disk unit on line. If an operation is specified beyond this boundary, the IMDC will abort the command.

Sector Length (1 Byte)
This byte defines the sector length as a power of two multiple of 128 bytes, i.e., sector length = $(2^{\#}) * (128$ bytes). The minimum sector length of 128 bytes is specified by a zero value while the maximum length of 4096 is specified by 5. The most significant bit of this byte is used for the IMDC's starting sector number, either 0 or 1. The three least significant bits are used for the sector length.

Flag Byte (1 Byte)
This byte is placed in the ECA by the IMDC during a transparent sector read (TSR) command operation. It is the flag byte for the data field part of the sector.

B-Tree Pointer (4 Bytes)
The IMDC loads the physical sector number of the last record match when in the PRP command mode and in the B-tree scan function (see the Programmable Record Processing section for further details).

IMDC Work Area (20 Bytes)
These ten words are reserved for the IMDC microprogram. The meaning of certain bytes of this area will be determined by the command being executed.

COMMANDS

The following is a generalized description of the sequence performed by the IMDC to execute a command:

a. The IMDC checks if the drive is available. If it is not, the IMDC generates an error status and a completion interrupt.
b. The IMDC executes a track seek if necessary. For drives which have a seek complete signal (as indicated by a zero value in the head settling time field of the ECA), a command termination can be caused by the timeout of the drive signal SEEK COMPLETE. Drives for which the head settling time is non-zero, the IMDC waits the specified time after issuing the stepping pulses (that is, there is no timeout on the seeks).
c. The head will be selected, or for a floppy disk, loaded. The IMDC will wait four byte times to insure head switching has occurred.
d. The IMDC reads the sector identifier to locate the requested sector. If the IMDC is unable to perform the sector identifier read, a retry is attempted for as many times as specified in the ECA maximum number of retries parameter. If the retries are all unsuccessful or if no retries are specified, the command is terminated (step g) and the status 'no identifier' is returned. If the matched sector identifier indicates a bad sector (see Track Format description), the IMDC uses the replacement information to perform steps b and c for the new sector. This bad sector replacement is not counted as a retry.
e. After the requested sector has been located, the IMDC performs the sector disk I/O (read or write) and DMA operations. If the operation cannot be completed, the retry processing described in step d is attempted. A failure in this portion of the command is indicated by the return status 'data part time out', which distinguishes it from the read identifier failure of step d.
f. If multiple sectors have been specified in the command, the IMDC assumes contiguous physical sectors: steps b, c, d and e are repeated for each sector. The IMDC automatically performs any track seek required if the next physical sector is located on the next cylinder. When the sectors are interleaved, the IMDC repeats step d until the correct sector is located.
In the case of a bad sector replacement, the IMDC continues the multiple sector operation, after the replacement sector, at the sector physically located after the bad sector. It performs all the necessary track repositioning required to return the drive head to the next physical sector.
g. Upon command completion, all relevant ECA fields are updated and the appropriate bits of the 'interrupt state status' and 'drive status and configuration' registers are set and an interrupt signal is generated by the IMDC. Upon interrupt acknowledge, the IMDC presents the interrupt vector from the interrupt vector register on the data bus.

Command Description

In the command descriptions, the term FIFO is used to refer to the internal memory of the IMDC when it is used as a FIFO buffer.

Write Multiple Sector with Implied Seek (WRMS)
After the desired sector is located, the IMDC will write the complete data part; preamble, address mark, flag, data, CRC or ECC and postamble. The precise format is given by the drive type. The FIFO will be continuously filled with the new data from the buffer. The number of data bytes written in one sector is determined by the IMDC using information from the ECA fields. This operation is repeated until the number of sectors, implied by the buffer length field of the ECA, have been written.

Write with Deleted Word (WRDD)
This command differs from the normal write only in that the flag or address mark written is the deleted data address mark or flag.

Verify (VER)
Verify with implied seek is the same functionally as the read command, except the IMDC compares the ECC or the CRC for accuracy. No data is transferred to the system.

Read Multiple Sector with Implied Seek (REMS)
After a successful seek to the desired sector, the IMDC will start to read the data of the sector, fill the FIFO buffer and check the CRC or ECC code. This sector data is transferred through the DMA interface to the host system memory. This procedure is repeated until the number of sectors, implied by the buffer length field of the ECA, have been read. For sectors with the deleted data address mark, the IMDC will process them according to the options selected in the command option field of the ECA.

Programmable Record Processing (PRP)
Refer to the Programmable Record Processing section.

Transparent Sector Read (TSR)
This command will read a single error sector and transfer the data to the FIFO buffer regardless of a CRC or ECC check. Only the extended status will be correspondingly filled with all errors encountered. The transparent sector read can be used for diagnostic purposes and, with some manual help, for recovery of damaged data. In case of an incorrect CRC/ECC remainder, the IMDC discards the remainder into four bytes reserved in the ECA for this value.

Read Identifier (RETD)
This command will read identifiers as they come from the disk and fill the whole buffer with records consisting of:
- flag byte
- ID data
- CRC
- remainder generated by IMDC

A CRC error will not terminate the execution of the command. For soft sectored formats, the command execution will begin and end with the detection of an index pulse. For the hard sectored formats, the IMDC will detect and count sector pulses to determine the command termination.

Format a Track with Implied Seek (FORM)
The IMDC supports four different media formats (see Disk Track Format section). This command allows the user to write the format information, as specified by the ECA track format fields, to the recording media.

Recalibrate to Track Zero (CALB)
The function of this command is to retract the heads to track 0. The IMDC issues one step 'IN' and then steps 'OUT' until the signal track 0 becomes active or the maximum number of steps equal the number of cylinders from the ECA. The check on seek completed timeout will be made for the hard disk. The function recalibrate can be initiated automatically by two conditions:
- Encountered error in cylinder number by reading or writing.
- Writing zero into the current cylinder in ECA.

Correct (CORR)
This command provides an error mask which is used to correct the data in the memory. It uses the ECC remainder field of the ECA. The IMDC will put the error correction vector and its relative position from the end of the data buffer into the ECA.

The 24-bit ECC correction mask will be placed in the flag byte and first half of the B-tree pointer of the ECA block. This is the least significant byte of the 33rd word and the 34th word of the ECA block in memory as shown in table 5. The offset count will be placed in the second half of the B-tree pointer or the 35th word of the ECA block. This is the relative offset from the last byte transferred to memo-

ry, plus four for a 32-bit ECC or five for a 40-bit ECC.

Chip Diagnostic (DIAG)
The IMDC will exercise the 16-bit counters; the record counter, the record length counter, the buffer length counter, and the time out counter. If everything is functioning properly, the IMDC places a zero byte into the main status. Otherwise, an irrecoverable error bit value is set in the main status.

DISK TRACK FORMAT

The IMDC supports four different track formats:

- IBM 3740 single density
- IBM System 34 double density
- Programmable soft sectored
- Programmable hard sectored

In each of the formats, data on the physical media is separated into blocks of information or sectors. The format serves several purposes in this arrangement; it defines the structure of the sector, the location of the actual data, and provides gaps and sync bytes that allow an interval for any switching required by the drive hardware. These intervals, in turn, allow the IMDC to compensate for any variations in either the recording media and/or the drive hardware. Although each of these formats is well defined, variations of parameter values within a given format require that the IMDC provide the user with the capability to program them.

Because there is no one standard which defines track format parameters, a description of them and other pertinent definitions are included in this section. See table 10 for track format definitions. Tables 11 and 12 provide a summary of the formats and the programmable values. The following definitions apply:

The format command operation is performed by the IMDC as integral operations on a per track basis, as opposed to normal disk I/O which is on a sector basis. This technique was selected as a compromise to satisfy two conflicting – requirements command efficiency versus equal access of on-line drives to the IMDC resources. Obviously, these commands would be most efficient if allowed to monopolize the IMDC resources. However, this situation would prevent any other drive from being serviced until the command had been completed. Normally these commands are background tasks with the other disk operations having a higher priority. For this reason, allowing the IMDC to concentrate completely on either command is not a good system practice.

The IMDC becomes available for other processing after a full track has been processed. In the worst case situation for the format command, the time the IMDC will not be able to process the other drives will not exceed two disk revolutions. This would occur for disks in which the index pulse was just missed and a complete revolution is required to find the pulse.

The programmable soft sectored disk format, which is used mainly for hard disk drives, is a MFM format which is nearly identical to the IBM double density format. The hard sectored disk format is used for rigid disks with an internal sector clock and is similar to the soft sectored format, except that each sector starts with the sector pulse rather than a related byte count from the index pulse.

The format track command allows the user to write formatting information to the recording media. Specification of format parameters is accomplished by changing appropriate ECA fields. This structure gives the user a tremendous flexibility to accommodate, through changes in the ECA, variations that occur within a given format.

For non-hard sectored disk formats, the IMDC writes the sector identifier and fills the data part with the fill byte starting with the leading edge of the first index pulse and ending with the leading edge of the next index pulse. For hard sectored disk formats, the format information is written between sector pulses.

Tables 11 and 12 show the possible layout of the track information in table format. Table 11 is for the floppy parameters and table 12 is for the rigid disk format parameters. In table 11 each of the different fields on each track is described. Associated with each field is where the IMDC gets the data, either from the ECA register locations or through the DMA process from the system memory.

ECA Track Format Fields

For the four formats supported by the IMDC, eight parameters are required to specify the format of the recording media to the IMDC. The format parameters are programmed by changing the values of the appropriate ECA fields. The layout of the track format portion of the ECA is given in table 13.

N0 and N1 are ignored by the hard sectored format and only N1 is applicable to the programmable soft sectored format. N5 contains the number of address marks contained in the address mark subfields for the active

Table 10. TRACK FORMAT DEFINITIONS

NAME	SUBFIELD	FORMATS	DESCRIPTION
N0	Index	IBM only	Pre-index Gap. This gap represents the number of bytes that appear prior to the index pulse.
N1	Index	All except hard sect	Index Gap. This gap represents the number of bytes that appear after the index pulse and prior to the ID subfield.
N2	ID, data	All	Preamble count or sync. This is the number of index sync bytes that precede the address mark.
N3	ID	All	Post ID gap. This count is the number of bytes that separate the ID subfield from the data subfield.
N4	Data	All	Post data gap. This count is the number of bytes that separate the data subfield to the beginning of the ID subfield of the next sector.
N5	ID, data	IBM	Address Mark Count. This contains the number of index address marks contained by the subfields. For single density formats, the count is one and for double density the count is three.
	ID, data	Prog	The number of data part address marks is a 1, 2 or 3. The number of ID address marks is always 1.

Table 11. SUMMARY OF FLOPPY FORMAT PARAMETERS

DESCRIPTION	IMDC USES	HEX DATA VALUE	FM CNT	HEX DATA VALUE	MFM CNT	
Pre-index gap	N0	FF	40	4E	80	
Sync field	N2	00	6	00	12	
Index mark	N5	FC/D7[1]	1	C2[2]	3	
Index flag		–	–	FC	1	
Index gap	N1	FF	26	4E	50	
Sync field	N2	00	6	00	12	Repeat as required
ID address mark		FE/C7[1]	1	A1[3]	3	
ID address flag		–	–	FE	1	
Cylinder	DMA		1		1	
Side	DMA		1		1	
Sector	DMA		1		1	
Record length	DMA	01[4]	1	01[4]	1	
CRC-CCITT			2		2	
Post ID gap	N3	FF	11	4E	22	
Sync field	N2	00	6	00	12	
Data address mark	N5	FB/C7[1]	1	A1[3]	3	
Data address flag		–	–	FB	1	
DATA (see note 4)		Fill byte	256	Fill byte	256	
CRC-CCITT			2		2	
Post data gap	N4	FF	27	4E	54	
Inter-record gap[5]		FF	170	4E	598	

NOTES:
1. Shows data pattern and clock pattern (clock pattern normally FF).
2. Shows data pattern; clock pattern should suppress clock bit between data bit 3 and 4.
3. Shows data pattern; clock pattern should suppress clock bit between data bit 4 and 5.
4. This example is for a 256 byte sector, others will have different values.
5. This is an approximate count.

Table 12. SUMMARY OF RIGID FORMAT PARAMETERS

DESCRIPTION	IMDC USES	HARD SECTOR		SOFT SECTOR		
		Hex Data Value	Pgm Cnt	Hex Data Value	Pgm Cnt	
Index gap	N1		22	4E	22	
Sync field	N2	00	13	00	13	Repeat as required
ID address mark	N5	A1[1]	1	A1[1]	1	
ID address flag		FE	1	FE	1	
Cylinder	DMA		1or2		1or2	
Head	DMA		1		1	
Sector	DMA		1		1	
CRC-CCITT			2		2	
Post ID gap	N3	00	3	00	3	
Sync field	N2	00	13	00	13	
Data address mark	N5	A1[1]	1	A1[1]	1	
Data address flag		FB	1	FB	1	
DATA (note 2)		Fill Byte	256	Fill Byte	256	
CRC-CCITT		note 3	2,4,6	note 3	2,4,6	
Post data gap	N4	00	3	00	3	
Inter-sector gap		4E	15	4E	15	
Inter-rec gap[4]				4E	346	

NOTES:
1. Shows data pattern; clock pattern should suppress clock bit between data bit 4 and 5.
2. This example is for a 256 byte sector; others will have different values.
3. Can be either a CRC-CCITT, 32-Bit ECC, or 40-Bit ECC Field.
4. Approximate count.

format; valid values are either one, two or three. N2 must be at least four.

Format Parameters

Although exhibiting some differences, the parameters that constitute each of the four formats supported by the IMDC are basically similar. In each format the data on the diskette or disk is separated into a logical data block or sector. The sector becomes the smallest block of information that can be addressed directly by the IMDC.

The programmable soft sectored and the two IBM formats organize the physical disk into a circular path or track, which, in turn, is separated into several sectors. In this scheme, tracks on the media are referenced with respect to a physical index mark. An index mark pulse is generated by the media to indicate the beginning of a track. By specification of a track and sector, the location of a sector on the media is uniquely addressed (for at least one side of the media). The programmable hard sectored format also uses this track and sector structure, however, it differs from the other formats in that in addition to the index pulse, each sector is preceded by a sector pulse.

In each of the formats, a sector can be further separated into two parts or subfields – an ID and a data subfield. The ID subfield contains a unique identifier (or address) and description of the sector. The data subfield contains the actual data of the sector. Both subfields contain three types of information:

1. Address mark – unique character which precedes the data in the subfield. For the MFM formats, (non IBM single density) it is followed by a single character, the address mark flag, which further describes the subsequent data.
2. Data (either actual or about the sector).
3. Error report – pattern generated by the IMDC that is used to verify the data transmitted.

Both fields contain sequences of characters called 'gaps' and 'syncs' that are used to differentiate the sector subfields and to provide an interval that allows any hardware switching to be performed. As a result, these sequences provide any timing compensation due to variations in either the recording media or the disk drive.

For the programmable hard sectored and IBM formats, (formats that use the index pulse), a third subfield, the index subfield, is utilized.

Table 13. ECA TRACK FORMAT

Word	15 – 8	7 – 0
16	N0 – Pre Index Gap	N1 – Post Index Gap
17	N2 – Sync Byte Cnt	N3 – Post ID Gap
18	N4 – Post Data Gap	N5 – Address Mark Cnt

This subfield appears prior to the first physical sector of a track on the recording media and has subfields which contain gap and sync sequences. Unlike the other two fields it occurs only once per track and contains only the address mark information.

Format Table

Table 14 contains the tables used by the IMDC during the format operation. Each table section is unique for the four different format types the IMDC can execute. The table is addressed by the SCWT pointer in the ECA block in RAM.

Provisions for a Bad Sector Substitution by Formatting

The IMDC command provides a convenient mechanism for handling bad sectors on the recording media. For example, consider the identifier (ID subfield) layout for a good sector using the programmable soft sectored format.

	Bytes	Coding Data/Clock
Preamble(sync)	N2	00
Address mark	1	A1/A0
Flag	1	FE (normal)
Cylinder number	1 or 2	
Head number	1	
Sector number	1	
Sector length	1	
CRC-CCITT	2	
Postamble	3	
Post ID gap	N3	

The data part of the sector is:

	Bytes	Coding Data/Clock
Preamble(sync)	N2	
Address mark	N5	A1/A0
Flag	1	FB (normal) or F8 (deleted)
Data	128X**L	L is the number between 0 and 5.
ECC	N6	
Postamble	3	
Post data gap	N4	

The media sectors with defects cannot be used for recording data. The host can replace any bad sectors it encounters during formatting with good sectors. Sector replacement is accomplished via the coding of identifiers (ID subfield) during the formatting. A bad sector is identified by the host placing an X'FF' in the sector length field of the ID part of the sector. The identifier for the bad sector is then extended with information pointing to the replacement sector.

Usually, there will be more than one identical bad sector identifier on the track for the same media defect. In this way, a later correct reading will be possible independent of the position of the media defect. During the read and write, the IMDC will automatically issue a seek to the replacement sector cylinder. There is no restriction on the placement of the substituting sectors. The identifier for a bad sector for the previous example would then appear as:

	Bytes	Coding Data/Clock
Preamble	N1	00
Address mark	1	A1/A0
Flag	1	FE (normal)
Cylinder number	1 or 2	
Head number	1	MSB set to 1 for bad sector
Sector number	1	
Cylinder number	2	
Head number	1	
Sector number	1	
CRC-CCITT	2	
Postamble	4	
Post ID gap	N3	

Table 14. FORMATS

IBM FM	
15	0
01	00
FE	00
80	N3-3
FF	Sector length
01	00
FB	00
Fill byte	00
FF	N4-5
FF	00
01	N5-1
DC	00
FF	00

IBM MFM	
15	0
00	02
A1	FE
80	N3-3
4E	Sector length
00	02
A1	FB
Fill byte	00
4E	N4-5
4E	00
00	N5-1
C2	FC
4E	00

PROG 1 AND 2 CYL	
15	0
00	00
A1	FE
00	N3-3
00	Sector length
00	N5-1
A1	FB
Fill byte	00
00	N4-2
4E	00
4E	00
00	00
00	00

Note that these tables are aligned to an even address boundary.

PROGRAMMABLE RECORD PROCESSING

The following definitions apply:

Key
Character string that the IMDC is to locate and match.

Scan
Search operation performed by the IMDC in trying to match the key.

Field (data item)
Smallest unit of named data. It consists of a string of characters that has a user defined significance. Records are formed by joining several fields together.

Record
Named collection of data items (fields) that has significance to the user.

Key Field
Field that contains the key data.

File
Named collection of all occurrences of given type of record.

The IMDC can be instructed to locate a specified string of characters within a logical data block on the recording media. After the string has been matched, the IMDC can perform the following additional functions:

- Retrieve and store the content of the logical data block
- Process a pointer from the data block
- Locate all data blocks that satisfy some search criteria

These primitive functions constitute the programmable record processing (PRP) capability of the IMDC, which can be used to form the basic support of more sophisticated applications such as:

- Directory processing
- Data block retrieval for data base management systems
- Processing of complex file structures (linked and mapped file structures)
- Searching of multilevel tree or network data structures

The programming for the PRP functions consists basically of two parts. One part involves loading of ECA fields with parameters that define the PRP operations and the physical representation of the data to be processed. The second part consists of table(s) which describe the search criteria on a character by character level.

PRP ECA Fields

The IMDC requires the following information to define the PRP operation:

1. PRP command control
2. Location of scan control word table
3. Location of matched record storage (not a separate PRP field; the IMDC uses the ECA buffer address field for this purpose)
4. Scan termination character
5. Maximum record length

PRP Command Control (1 Word)
This field contains the execution control parameters for the PRP command. The format of the command control field is:

B-tree scan (bit 1) — This parameter enables the B-tree scan function on the IMDC.

No data transfer — Returns pointer in ECA B-tree pointer.

Record type (bit 2) — This parameter contains a code which specifies the record type:

0 Fixed length. The record length is given in the maximum record length (MAXLEN) field of the ECA.

1 Variable length. The record is of variable length and uses the content of the scan terminator field of the ECA to determine the end of the record. If this terminator is not found, the IMDC uses the MAXLEN parameter to terminate the record search.

Location of Scan Control Word Table (2 Words)
This field contains the address of the start of the scan control word table in system memory.

Scan Terminator (1 Byte)
This character specifies the end of the scan field in the record.

Maximum Record Length (MAXLEN – 1 Word)
This field plus one is the maximum length of variable length records and is the length of the fixed length records.

Scan Control Word Table

The scan control word (SCW) table's primary task is to provide the character by character comparison template. It also is used to locate the key field and provide processing instructions for the IMDC as shown below. It must be located on an even word boundary.

15	0
Program length	Record start offset
Number of record to be scanned	
Scan control word(s)	

Bits 8 – 15
DATA – This 8 bit field is used in the following manner depending on the usage defined by the operation.

1. It contains the data to be matched against the key
2. It contains a count for repeating a previous instruction.
3. It contains the character used with the continue opcode.

Bit 5
SUB – This bit is used to indicate the last character of a subkey or the last character of the key to the IMDC. This flag is used by AZFF and SUCFF to control generation of the automatic initialization state for these devices. This bit must be set for the last byte of the key field, i.e., when bits 6,7 are 0,0.

Bit 4
ORENA – This bit is used to indicate to the PRP processor that the result of the previous subkey comparison is to be 'ORed' with the next logical comparison. This bit must be set for the last byte of the key field, i.e. when bits 6, 7 are 00.

Bits 6, 7
OP – This 2 bit field is a command to the IMDC to perform one of the following operations independent of the resultant logical processing being performed.

7 6
0 0 Indicates the last byte of the key field.
0 1 Read this character (data) into system memory from this character until either:
  1. Stop read command is issued, or
  2. End of key, and no match has occurred.
1 0 Stop reading
1 1 No-op

Bits 0 – 3
OP CODE – This field is used to indicate either the logical comparison or some special control function is to be performed.

CODE	OPERATION
0	Repeat previous instruction (data + 1) times
3 – 1	Not used
4	SUCFF is unchanged
5	Continue scanning, no-op until 'data' byte is read
	Not used
	If ORFF is set, then store this character into the FIFO. If ORFF is not set, then use last point in FIFO as address of next sector to be scanned on the next level of the tree or plex.
8	Reset SUCFF if (CHAR GT Data) • (AZFF = 1).
9	Reset SUCFF if (CHAR GT or EQ Data) • (AZFF = 1).
10	Reset SUCFF if (CHAR LT Data) • (AZFF = 1).
11	Reset SUCFF if (CHAR LT or EQ Data) • (AZFF = 1).
12	Reset SUCFF if CHAR = Data.
13	Reset SUCFF if CHAR NEQ Data.
14, 15	SUCFF is unchanged

The number of scan control word tables required to describe a PRP function is determined by the data structure organization. For records organized in a relational structure, a single SCW table is required to provide the pertinent parameters. A complete description of the PRP operation is defined by the single SCW table and the ECA fields.

For records organized into a tree or plex structure, each level generally requires a new SCW table. Because each level may have a different record structure, certain ECA parameters describing the record structure must be updated (see PRP ECA fields discussion).

Note that a repeat operation cannot follow a continue operation.

Location of Matched Record Storage (2 Words)
The IMDC will use the previously defined 'buffer address' field.

PRP Status Flip Flops

To understand the operation of these flip flops, one must differentiate between the 'automatic initialization state' and the 'logical operation' of each of these internal devices. If this distinction is not made, some apparent conflicts in their logical state appears to exists. For example, a contradictory condition appears to occur for the accumulated zero flip flop (AZFF) for the last character of the subkey. According to the description of the flip flop, it is reset if the comparison fails. However, in the same description, it is stated that the AZFF is set if the character is the last of the subkey string. This apparent contradiction is easily explained; it is simply a problem of failure to recognize the sequence of events that is involved in execution of the 'logical operation' and the 'automatic initialization state'. The logical operations are performed by the flip flops prior to the automatic initialization state.

In order to help in making this distinction, the operation of each of the flip flops is described in terms of the logical operation and the automatic initialization state rather than in terms of the set/reset conditions.

Accumulated Zero Flip Flop (AZFF)
This flip flop is used to report the status of each character by character comparison. It is used to identify the first unsuccessful character by character match.

Logical Operation — The flip flop is reset on the first unequal character comparison.

Automatic Initialization State — The operation of the flip flop is initialized to the set condition:

1. At the beginning of a new record.
2. At the end of a subkey prior to the next subkey. The SUB bit of the SCWT indicates this condition.

Success Flip Flop (SUCFF)
This flip flop can be used to indicate the logical status of the subkey match operation. The AZFF cannot be used for this purpose since its primary task is to respond to the character by character search operation.

Logical operation — The SUCFF is reset by the interaction of the SCWT operation (OP code) and the status of the AZFF.

Automatic initialization state — This flip flop is set for the following conditions:

1. At the beginning of a new record.
2. At the end of a subkey if the ORENA bit of the SCWT is set.

OR Flip Flop (ORFF)
This flip flop is used to indicate the resultant scan status for the entire key field. The ORFF will always be in the set condition at the conclusion of a successful key scan, regardless of the logical operation performed between the subkeys.

Logical operation — The ORFF is set at the end of a successful subkey comparison; this condition is indicated by the SUCFF and the ORENA bit of the SCW table both being set.

Automatic initialization state — RFF is reset for the following conditions:

1. At the beginning of the first record.
2. At the beginning of all records to be scanned.

Programmable Record Processing Operation

Although the programmable record processing command utilizes most of the basic disk read sequence described earlier, its execution involves significantly more character processing than is performed for the basic read operation. For the PRP command, the function of the IMDC memory is split; one portion is allocated to store the PRP program contained in the SCWT table and the remaining portion is used as a FIFO buffer. The term 'FIFO' for this discussion of the PRP command refers to the memory available after the SCWT has been loaded. The command is 'normally' executed in the following sequence:

1. Transfer the SCW table data from system memory to a portion of the IMDC memo-

ry. The remainder of the IMDC memory is used as a FIFO to store the data from the record.
2. Perform disk positioning I/O to locate the drive read/write heads to the first sector at which the PRP is to operate.
3. Perform a read of the sector ID part to locate the data part.
4. Read an operand from the data part of the sector. The IMDC performs processing of this operand in order to locate the key field. This step is repeated until the key field is located.
5. After the key field is located, it is compared character by character with the SCWT template. This step is repeated until a match or no match with the key is determined.
6. If the scan does not produce a match the current record is skipped and the next record is located for processing by repeating steps 3, 4 and 5.

Programming Parameters Supplemental Information

The following describes programming parameters in more detail.

Record Format Types

Records to be processed by the IMDC, regardless of the overall data organization, can be formatted in one of two types.

1. Fixed – Records consist of a fixed number of characters. The user must pass a record length parameter value to the IMDC.
2. Variable – Records are of variable length. Requires the user to specify a unique termination character to identify the end of each record.

Since all records organized in the relational data structure are identical, the selected record format is applicable to all the records being processed. For data organized in a tree or plex structure, each level may have a different format type, therefore, each requires this parameter to be specified.

Key Field Location

The IMDC allows the user to specify the location of the key field in the record. The SCW allows the user to program the necessary information to locate the key field. The record format type determines how the key field is specified.

1. For fixed length records, the key start occurs at some fixed number of characters from the beginning of the record. For these records, the user must provide the IMDC with this count.
2. For variable length record, the key field location is specified by the number of data fields occurring prior to it in the record. Data fields consist of data that is bordered by a 'marker' character. Normally, this character will be some unique control character like a tab, line feed or return code. The PRP hardware will count the occurrence of the markers to locate the key field. Specification of the variable length field requires the user to provide the 'marker' as a parameter to the PRP.

Record Offset Start

The recording media often contains some header or identification information prior to the first record. The IMDC allows the user to specify where the PRP control is to become active. This allows the IMDC to skip any information that is not to be processed by the PRP. This parameter is referred to as the 'record start offset'.

Number of Records to Process

Because a user does not always know how many records are to be processed, the IMDC provides mechanisms to limit the number of records that are to be processed. First, the user is required to specify the 'number of records to process' parameter which is a count of the maximum number of records to be processed. In addition to this count, the user must also specify either an 'end of record' character for variable length fields or a record length count for fixed length records. The IMDC will monitor the number of records processed and will terminate the processing when the total equals the 'number of records to process', or if the 'buffer length' is exceeded.

Programming the PRP Feature

One of the most powerful aspects of the PRP feature is the capability it provides the user to effectively specify the search criteria on a character by character basis. The PRP feature not only allows the user to establish the criteria for success, but also to specify the action to be taken after completion of a search. The PRP program consists of two major parts. The first part essentially defines the basic requirements of the processing and the structure of the records; it is given once during initialization of the feature to the IMDC. It consists of loading several ECA fields dedicated to defining the parameters of the PRP operation to the IMDC.

Scan Control Word Table

The second part of the PRP program consists of a template which is used to perform the character by character match of the record data as it read from the media. The template consists of the character string data to be matched and processing instruction for each character of the string. The IMDC utilizes a 16-bit word to implement this comparison/instruction. It is loaded by the host into system memory and is transferred by the IMDC into its memory before execution of the PRP command. The SCWT is used in conjunction with three internal IMDC statuses to control and monitor the operation of the PRP search. Because of the interaction, they will be described prior to presentation of the table format.

PRP Key Status

To implement the PRP feature, the IMDC performs a character string search of the key field against the scan control word data. The IMDC generates and stores the status of this comparison in three flip flops dedicated to reporting the status of this operation. The logical operation of these flip flops is as follows:

For the most general case, the key field is composed of substrings. For convenience these substrings shall be referred to as subkeys for the remainder of this section. The desired logical result of the scanning of the entire key field is the logical combination of the constituent subkeys. For this general situation, there is the possibility of three levels of comparison statuses that occur in attempting to match a single key field – character, subfield, and entire key field each has a status that must be monitored by the IMDC. The IMDC has three flip flops that generate a status that allows these conditions to be monitored. Consider the example:
YYYYYY985DEF12580XXXXXXXXXXXX

This key field consists of five subkeys; YYYYYY, 985, DEF, 12580 and 'XXXXXXXXXXXXX', referenced arbitrarily as subkeys A, B, C, D, and E, respectively. Suppose that each subkey has the following logical condition associated with it.
A – Value is not processed by the PRP
B – Value greater than 980
C – Value equal to DEF
D – Value less than 20000
E – Value is not processed by the PRP

To perform the scan operation for this example requires that the IMDC be able to generate and monitor the condition of the character by character comparison (level 1), each subkey (level 2) and finally the entire key (level 3). The SCWT has been structured to allow specification of such combinations.

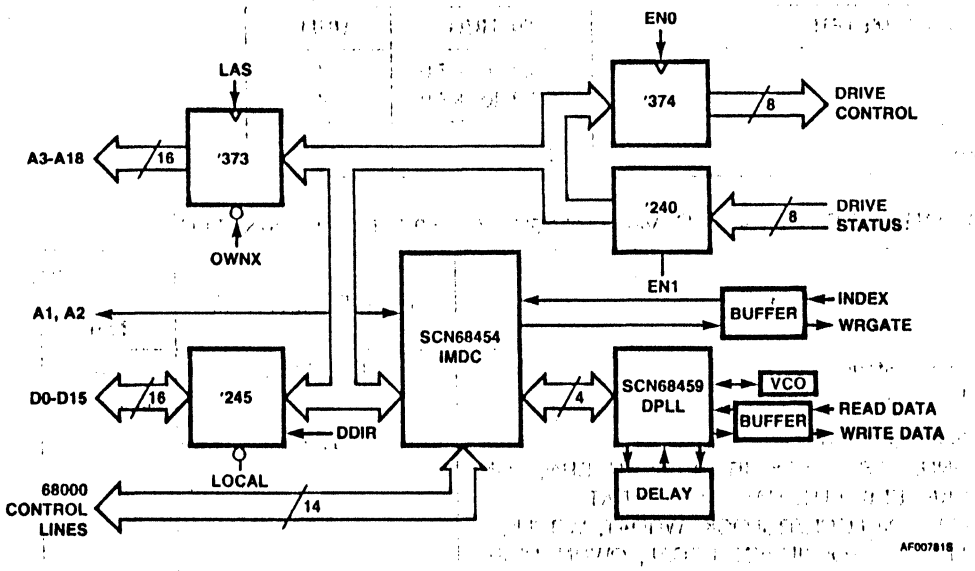


Figure 4. IMDC Application

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltage	−0.3 to +7.0	V
Input voltage[2]	−0.3 to +7.0	V
Operating temperature range[3]	0 to +70	°C
Storage temperature	55 to +150	°C

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5V ± 5%, V_{SS} = 0V; T_A = 0°C to +70°C[4,5]

	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IH} V_{IL}	Input high voltage Input low voltage	I_{IL} = 20μA	2.0 GND − 0.75	V_{CC} 0.8	V V
I_{IN}	Input leakage current RERUNN, RESN, SCLK, IRQN, IACKN, BRN, BGN, BGACKN, EN0, EN1, UAS, LAS, REDAT, WRGATE, WCLOCK/RCLOCK, WRDAT, WRCLK, TICKLER, SECTOR/INBEN, LOCAL, OWNN, DDIR	5.25V		20	μA
I_{TSI}	Three-state (off state) input current A1, A2, D0 − D15, ASN, UDSN, LDSN, R/WN, DTACKN	2.4V/0.4V		20	μA
V_{OH}	Output high voltage A1, A2, D0 − 15, ASN, UDSN, LDSN, R/WN, DTACKN, EN0, EN1, UAS, LAS, WRGATE, WRDAT, WRCLK, TICKLER, LOCAL, OWNN, DDIR	I_{OH} = −400μA I_{OL} = 6.3mA	2.4	V	
V_{OL}	Output low voltage A1, A2, D0 − D15, ASN, UDSN, LDSN, R/NN, DTACKN IRQN, BRN, BGACKN, EN0, EN1, UAS, LAS, WRGATE, WRDAT, WRCLK, TICKLER, LOCAL, OWNN, DDIR			0.5	V
P_D	Power dissipation			1.5	W
C_{IN}	Capacitance	V_{in} = 0V, T_A = 25°C f_o = 16MHz		10	pF

NOTES:

1. Stresses above those listed under absolute maximum rating may cause permanent damages to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltage larger than the rated maximum.
3. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8 and 2.0V as appropriate.

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$; $T_A = 0°C$ to $+70°C$[4,5] (see figures 5 – 18)

NO.	FIGURE	CHARACTERISTICS	TENTATIVE LIMITS		UNIT
			Min	Max	
1	5	Cycle time		62.5	ns
2	5	Clock pulse width low	21		ns
3	5	Clock pulse width high	21		ns
4	5	Rise time		10	ns
5	5	Fall time		10	ns
6	6	Reset pulse width	1		µs
7	7, 8	A1 and A2 set-up to CSN low	0		ns
8	7, 8	Local low after CSN low		1.2	µs
9	7	D0 – D15 valid data from ASN, CSN, and UDSN or LDSN low		1.2	µs
10	7, 9	DTACKN low after D0 – D15 valid data	62.5		ns
11	7	CSN high after ASN, UDSN, LDSN, A1 and A2		10	ns
12	7, 8	D0 – D15 hold after CSN high	0		ns
13	7, 8	Local high after CSN high		100	ns
14	7	DTACKN high after CSN high		100	ns
15	7, 8	CSN low time	1.2		µs
16	7, 8	A1 and A2 HOLD after CSN high	0		ns
17	8	DDIR low after CSN low		1	µs
18	8	DTACKN low after CSN low		1.4	µs
19	8	R/WN low before CSN low	0		ns
20	8	R/WN low after CSN high	0		ns
21	9	IACKN low after last of ASN AND LDSN		30	ns
22	9	IRQN high after IACKN low		140	ns
23	9	Local low after IACKN low		1.2	µs
24	9	D0 – D15 valid after last low of ASN, LDSN, IACKN		1.2	µs
25	9	IACKN low time	1.2		µs
26	9	D0 – D7 hold after LDSN high	10		ns
27	10	BRN high after BGACKN low		1.2	µs
28	11	OWNX low after BGACKN low		1.2	µs
29	11, 12	D0 – D15 valid before either UAS or LAS low	65		ns
30	11	DDIR low after OWNX low		500	ns
31	11, 12	D0 – D15 valid after either UAS or LAS low	5		ns
32	11, 12	LAS low before ASN low	0		ns
33	11	ASN, data strobes width low (read)/ASN write	440		ns
34	11	Local low after data strobes low		20	ns
35	11, 12	D0 – D15 valid after data strobes high	0		ns
36	11	Local high after data strobes high	0		ns
37	11	DDIR high after ASN high	20		ns
38	11, 12	OWNX high after ASN high		400	ns
39	11, 12	BGACKN high after OWNX high		30	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTICS	TENTATIVE LIMITS		UNIT
			Min	Max	
40	12	Data strobes with low	80		ns
41	12	R/WN low before ASN low	35		ns
42	12	D0 – D15 valid before data strobes low	62.5		ns
43	12	R/WN low after data strobes high	30		ns
44	13	D0 – D15 valid before EN0 high	40		ns
45	13	D0 – D15 valid after EN0 low	40		ns
46	14	D0 – D15 valid after EN1 low		40	ns
47	14	D0 – D15 valid after EN1 high	0	50	ns
48	15	WRGATE high after SECTOR/INDEX high	TBD		
49	15	WRDAT or WRCIK valid after WRGATE high	TBD		
50	16	REDAT high before WCLOCK/RCLOCK high	20		ns
51	16	REDAT high after WCLOCK/RCLOCK high	20		ns
52	18	RERUNN low before DTACKN low		20	ns
53	18	RERUNN low after DTACKN high	320		ns
54	11	DTACKN low after D0 – D15 valid		–62.5	ns
55	8	D0 – D15 valid after data strobes low		62.5	ns
56	12	Local low before data strobes low	50		ns

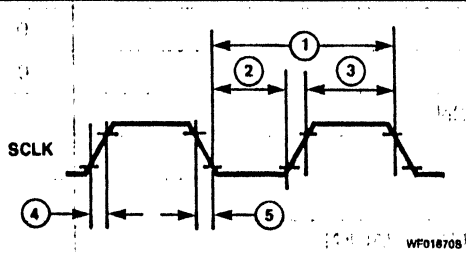


Figure 5. Input Clock Timing

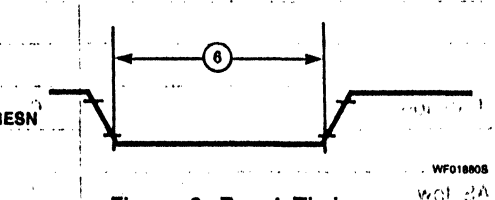


Figure 6. Reset Timing

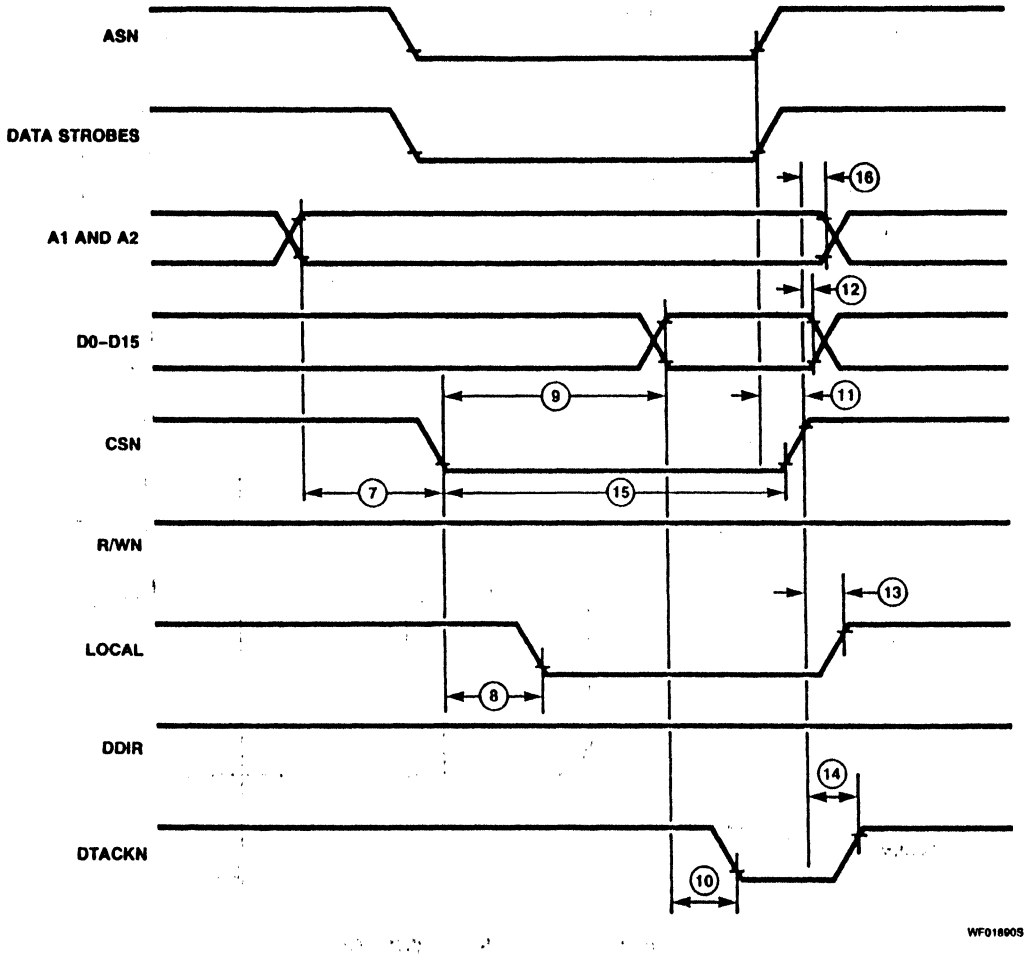


Figure 7. IMDC Register Read Cycle

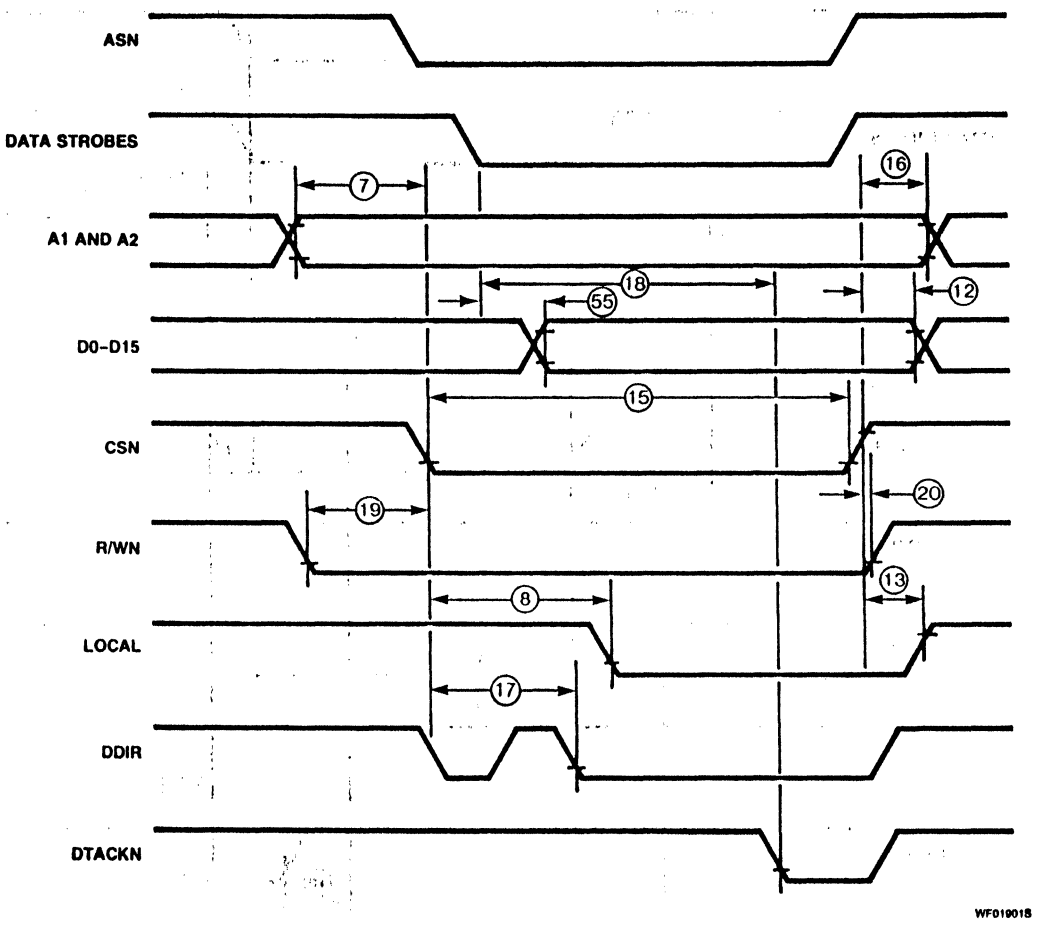


Figure 8. IMDC Register Write Cycle

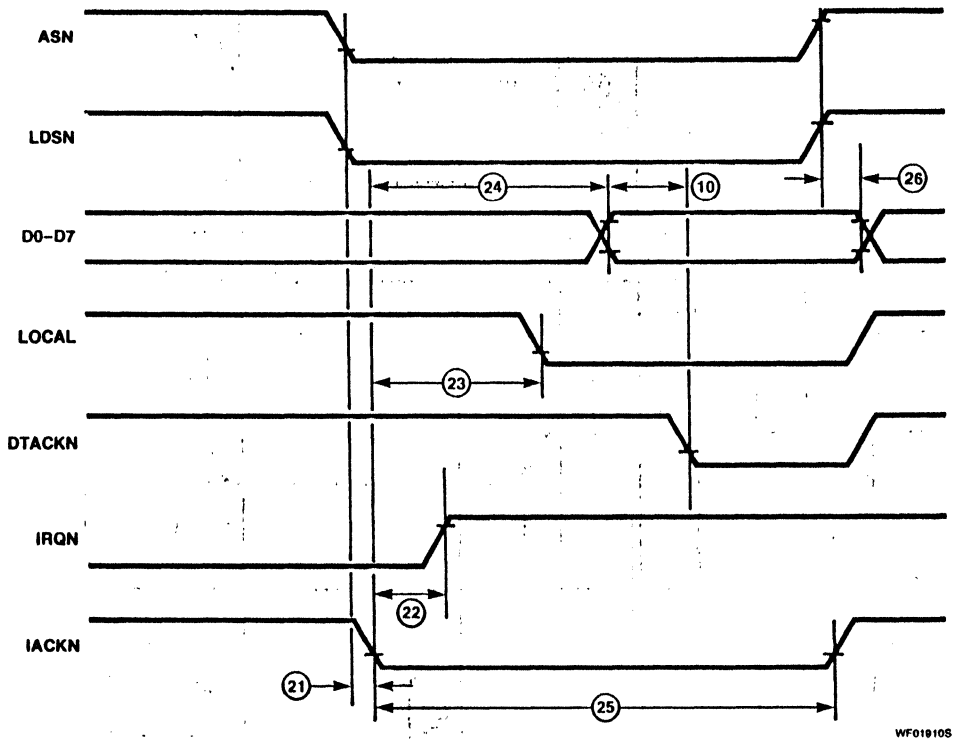


Figure 9. Interrupt Acknowledge Cycle

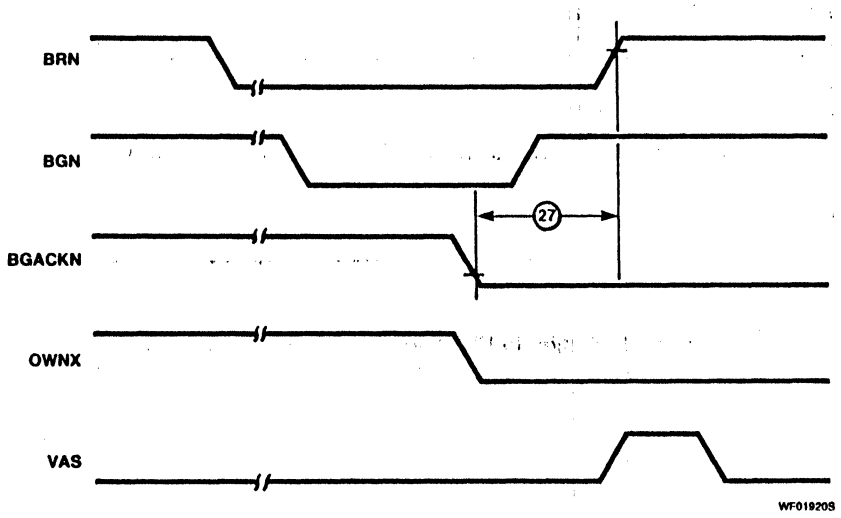


Figure 10. Bus Arbitration

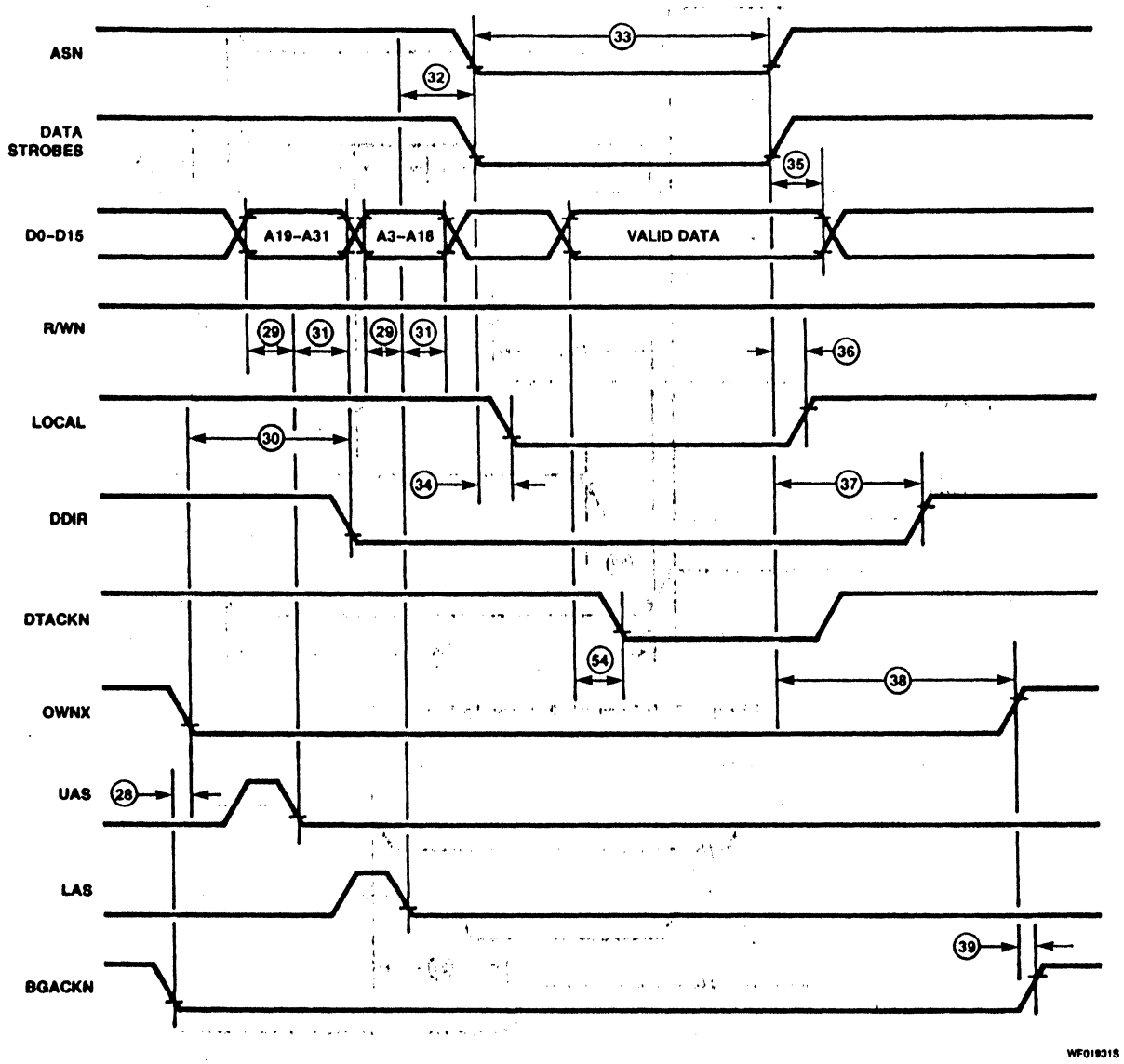


Figure 11. Single 16-Bit DMA Read From Memory Cycle

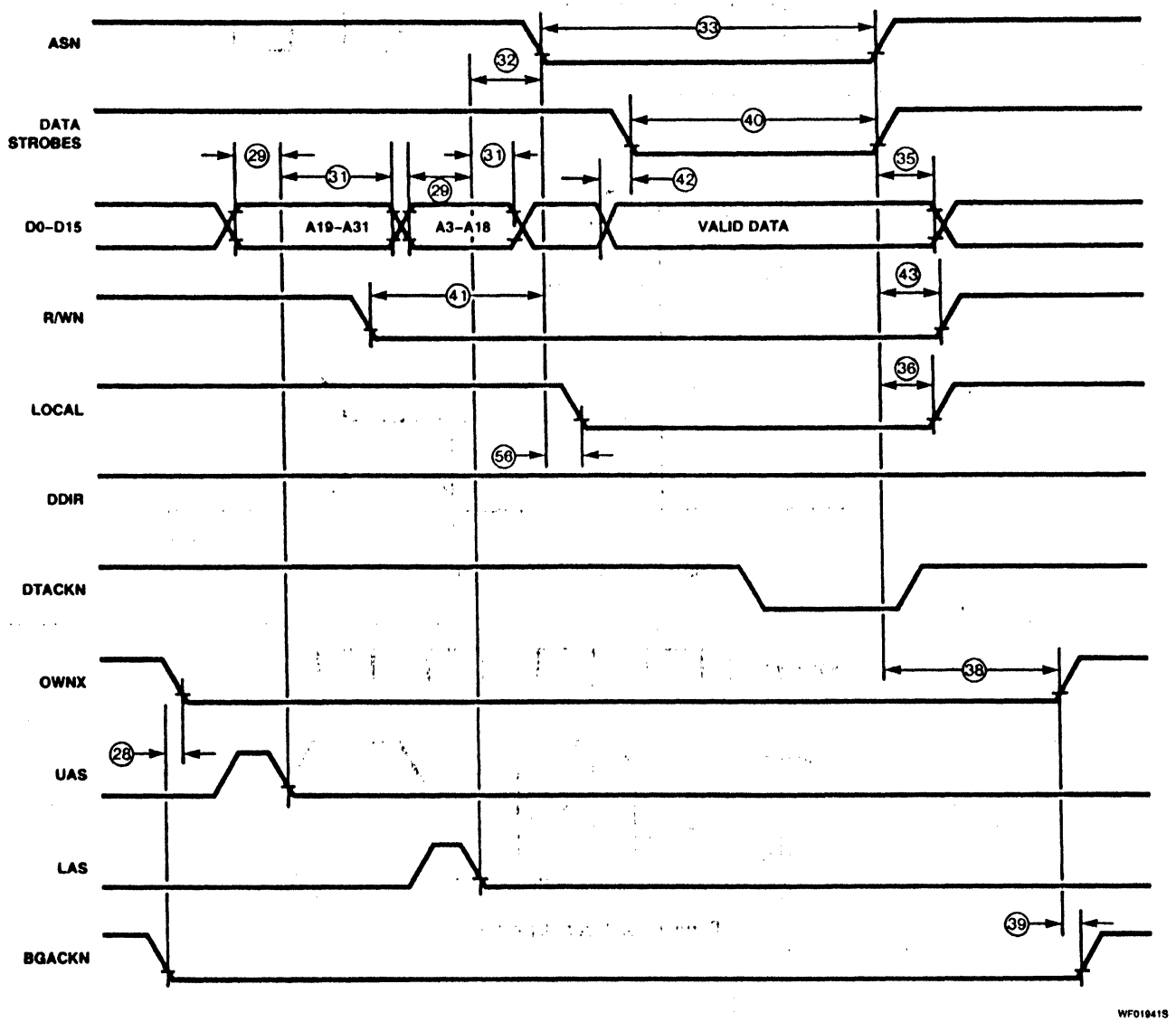


Figure 12. Single 16-Bit DMA Write to Memory Cycle

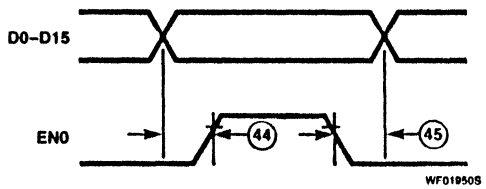


Figure 13. IMDC Output Port Timing

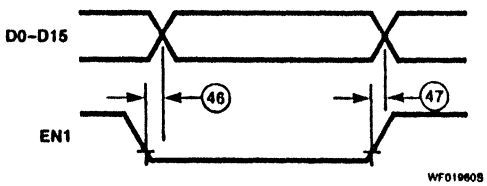


Figure 14. IMDC Input Port Timing

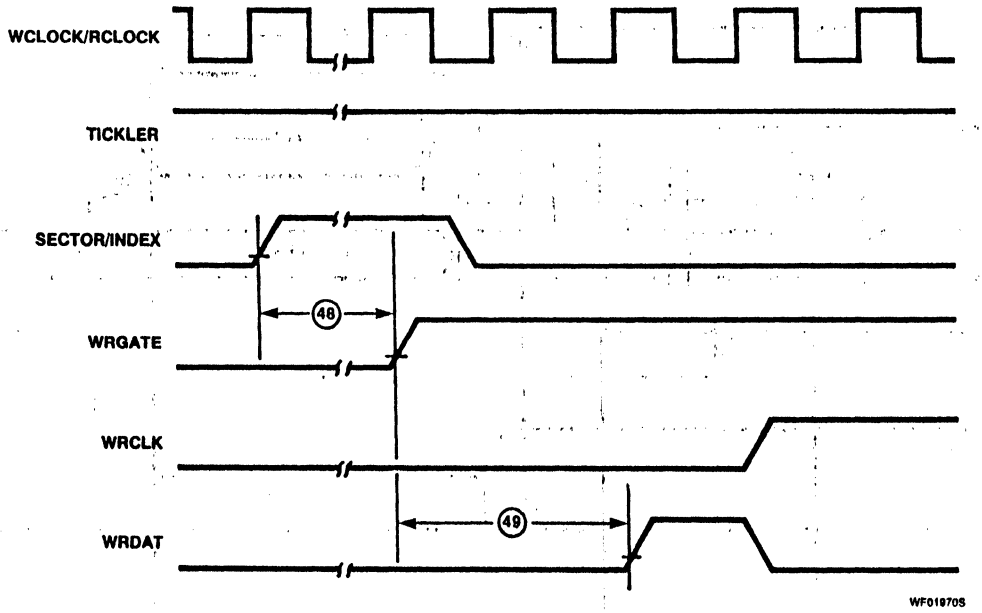


Figure 15. Write Operation

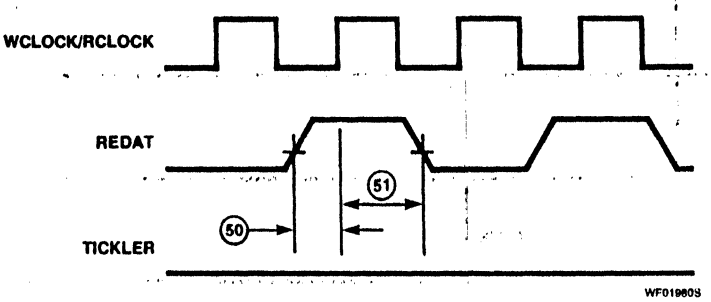


Figure 16. Read Operation

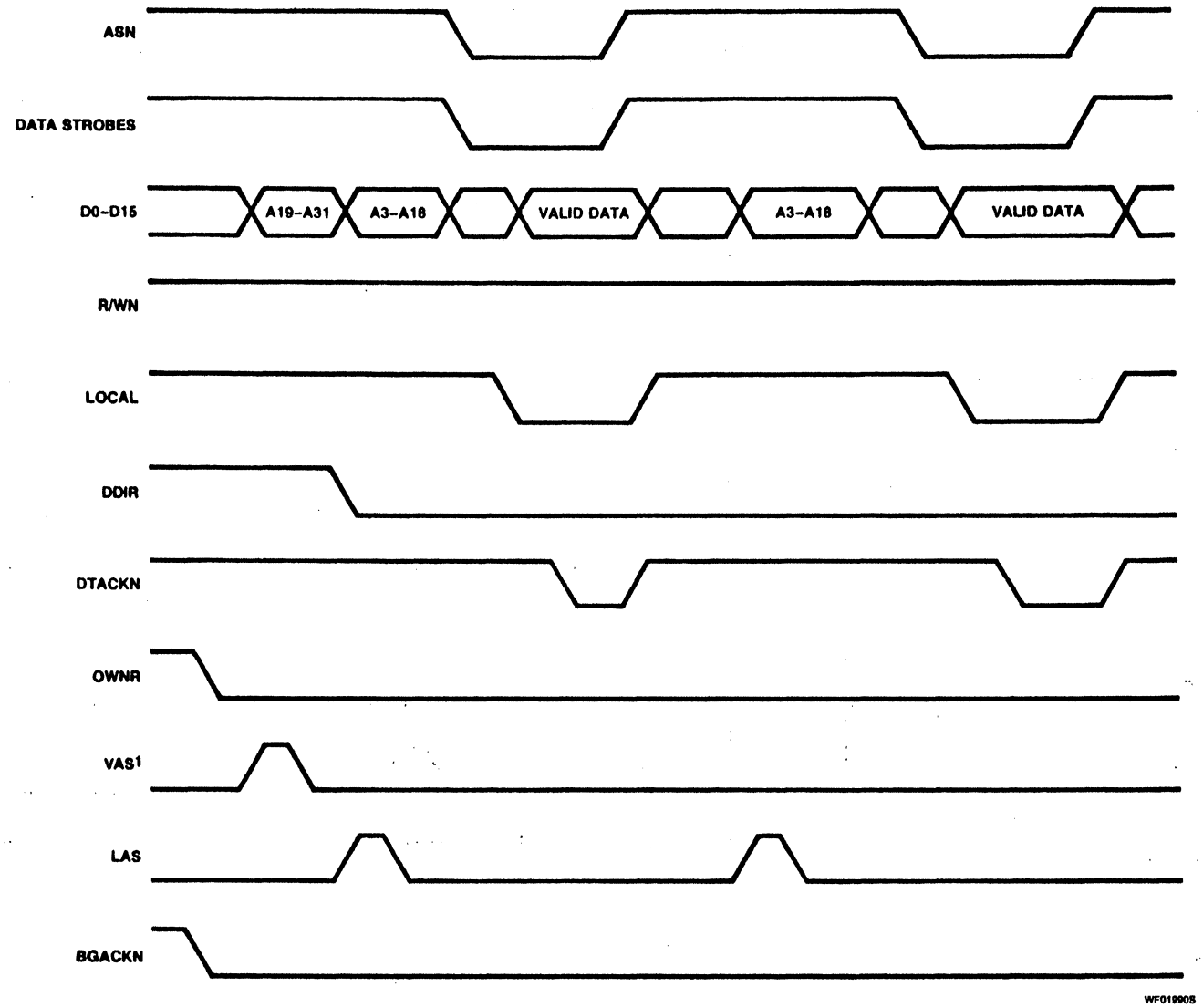


Figure 17. Example of Multiple DMA Read Cycles

[1]If the high order address does not change, VAS will not be asserted.

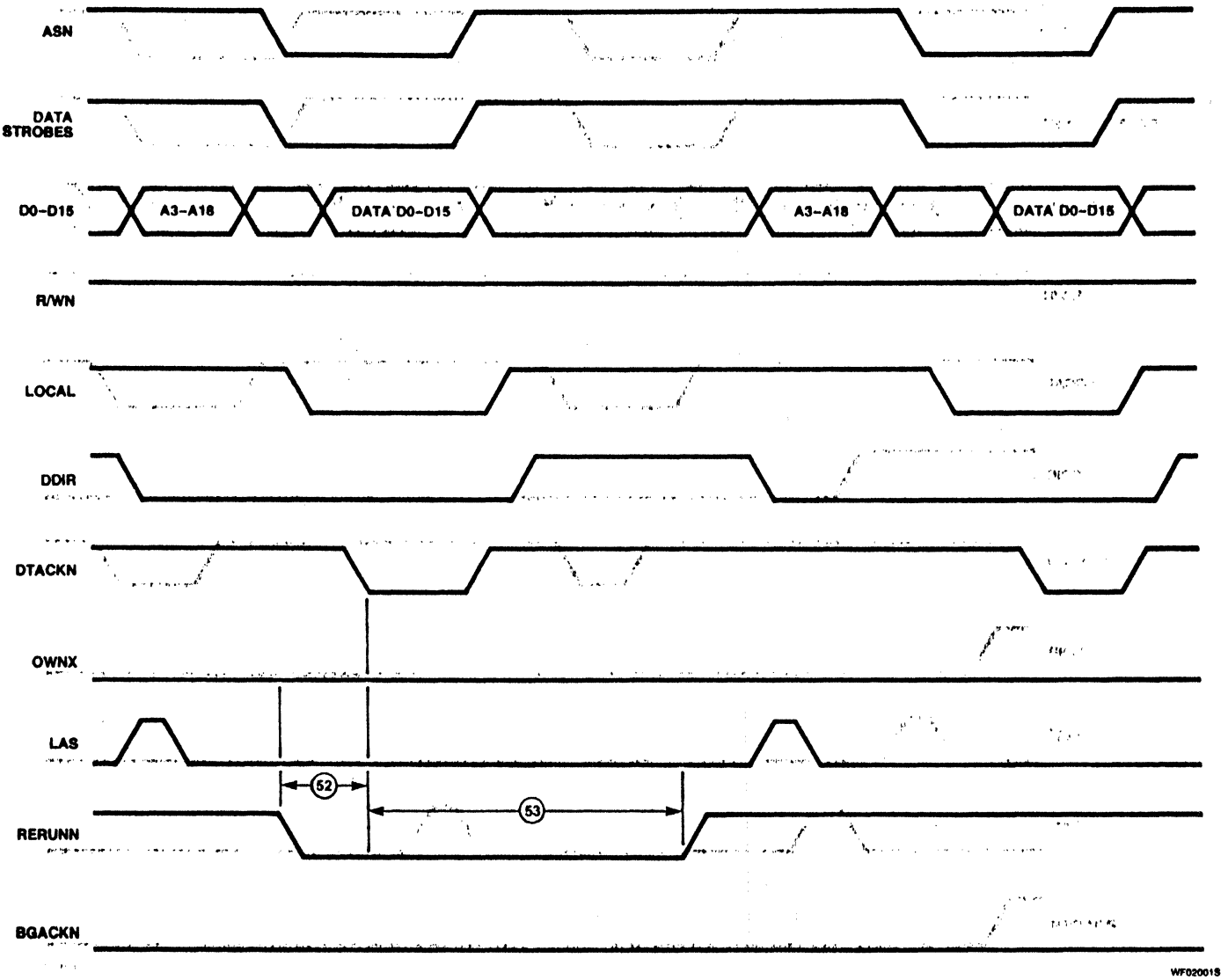


Figure 18. RERUNN Asserted During DMA Read Cycle