XEROX

Preliminary
008050-002

Dove IOP Technical Reference Manual

February 1, 1985

NOTICE: FOR INTERNAL XEROX USE ONLY
Warning: This Document is in ROUGH Draft form. Technical errors are a high probability. Much of the document has not been edited,and information is still being compiled.

Office Systems Division
Systems Development Department
Xerox Corporation
3450 Hillview Avenue
Palo Alto, California 94304

To: John Holzemer

From: Mary Lou Nohr
Roy Ogus

Re: IOP Technical Reference Manual 008050-002 (dated February 1, 1985)

February 1, 1985

Attached is the manual that Hardware Documentation agreed to deliver to you on this date for release to field service.

We want to emphasize caution in the use of the manual for the following reasons.

- The manual is incomplete. Some omissions have been noted in the text. Some omissions are inadvertent, and we hope these will be pointed out to us.
- Not all sections of the manual have been reviewed by the designers, and it is highly likely that the unreviewed sections will contain inaccuracies. The Table of Contents indicates which sections have not been reviewed.
- The manual may be inaccurate because of design changes after the manual was printed.

We will welcome comments and corrections from field service.

Roy C. Ogus

Roy Ogus, Hardware Design Manager

Mary Lou Nohr

Mary Lou Nohr, Hardware Documentation Team Leader

Preface

P.1 Purpose

The IOP Technical Reference Manual describes the hardware on the Dove IOP board. The manual describes the theory of operation of the components, and presents information important for programmer interface.

P.2 Audience

The following groups are expected to refer to this publication:

Engineers
Field Service personnel
Microprogrammers
System designers

P.3 Organization

After a brief overview, each major part of the IOP board is discussed in a discrete section. Components are described as to hardware, theory of operations, and programmer interface.

Hardware consists of: board layout and component discription including pins and signals

Theory of Operations describes system operating modes.

Programmer Interface describes register assignments and timing.

Appendices contain parts lists for each board, as well as schematics. (to be provided).

P.4 References

The following documents contain supplementary information.

INTEL Microsystems Component Handbook Volumes I and II.-1984

INTEL-The Complete VLSILAN Solution-1982

NEC Microcomputers, Inc- Application Note 8-A Single/Double Density Floppy Disk Controller using the PD765.

SIGNETICS- Spec sheet on the 8x305 microcontroller

Standard Microsystems Corporation-Data Catalog- 1983

United Technologies Mostek- Memory Components, 512 x 9 BiPort Parallel IN-Out FIFO

IOP Table of Contents

Appendice

Figures

Tables

1 IOP Board

The I/O subsystem is one of the major subsystems of the Dove Workstation. The subsystem is common to both the Daisy and Daybreak implementations of the workstation.

1.1 Overview

Figure 1.1 illustrates the I/O subsystem, and shows its internal elements as well as its relationship to the system as a whole. This manual describes the elements that constitute the I/O subsystem.

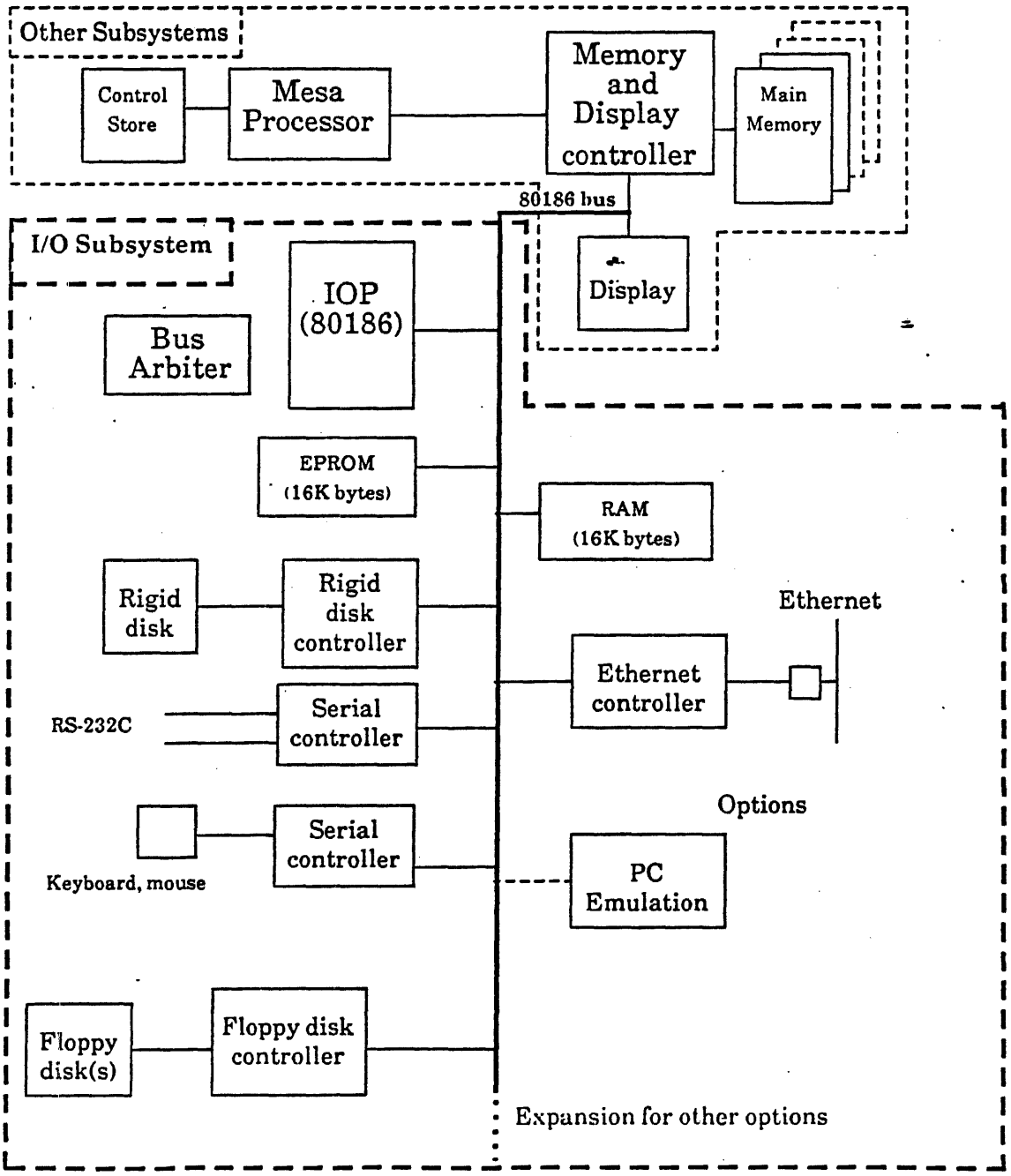


Figure 1.1. Dove workstation block diagram

The main functions of the I/O subsystem are:

- Controls all the I/O devices associated with the Dove workstation during system operation. With the exception of the bitmap display controller, all hardware associated with the peripheral devices is embodied in the I/O subsystem. All software that directly controls the I/O devices runs on the IOP. The display controller is programmed by the IOP, but most of the hardware associated with it is found outside the I/O subsystem.

- Controls the Mesa processor during power-up and initialization. The IOP is responsible for bringing the Mesa processor up to a functional state after the machine is powered up or booted. This process ensures that the various processor states are correctly initialized, and then starts the Mesa processor.

- Writes and reads the Mesa processor control store. The write function is used to initialize control store with microcode before the Mesa processor operates. The read function is used primarily as a diagnostic function to check the correctness of control store.

- Provides the system booting function. This function involves the multi-stage bringup of the system using the boot files stored on one of several boot devices. The IOP runs the software that bootstraps from the raw machine to a fully-functional Star workstation.

- Forms the basis of the diagnostic capability of the Dove workstation. Since the IOP can, to some degree, control all other subsystems, it can selectively exercise and diagnose problems in these subsystems.

- Provides the hardware and most of the software for the PC emulation function.

- Provides the framework by which optional devices can be attached to the Dove workstation. Control of the various Options slots is exercised by the IOP.

The I/O subsystem is controlled by the I/O Processor (IOP), a commercial VLSI microprocessor. The subsystem has a traditional microprocessor bus architecture to which memory devices and I/O controllers can be connected. All Dove I/O devices, except the display controller, interface to this bus. The devices include high speed devices like the Ethernet controller and rigid disk controller, medium speed devices like the floppy disk controller and serial communication controllers, and low speed devices like the keyboard and mouse.

The microprocessor bus also extends to the Options slots, where additional peripheral controllers can be added.

The physical location of the IOP peripheral devices is as follows:

- the rigid disk drive is housed in the System Unit.
- the floppy disk drive is located in a separate module on top of the Systems Unit.
- the keyboard and mouse are located on the user's desktop near the display unit.
- the Ethernet transceiver is located at some remote distance from the Systems Unit, typically in the area above the ceiling where the Ethernet cable is situated.
- External I/O connections to the electronics are made at the rear of the system.

1.2 Hardware

All components of the I/O subsystem are common to both Daybreak and Daisy, and the boards plug into an identical backplane. The IOP electronics are housed in the System Unit. Appendix A contains a parts list for the main IOP board.

1.2.1 Printed Wiring Board Assemblies (PWBA)

IOP electronics are located on six PWBAs: the 16" x 10.9" main IOP board; a 5" x 10.9" IOP expansion board; the 5" x 10.9" PCE board; and three 5" x 10.9" options boards. Figure 1.2 illustrates the location of the I/O subsystem PWBAs on the backplane.

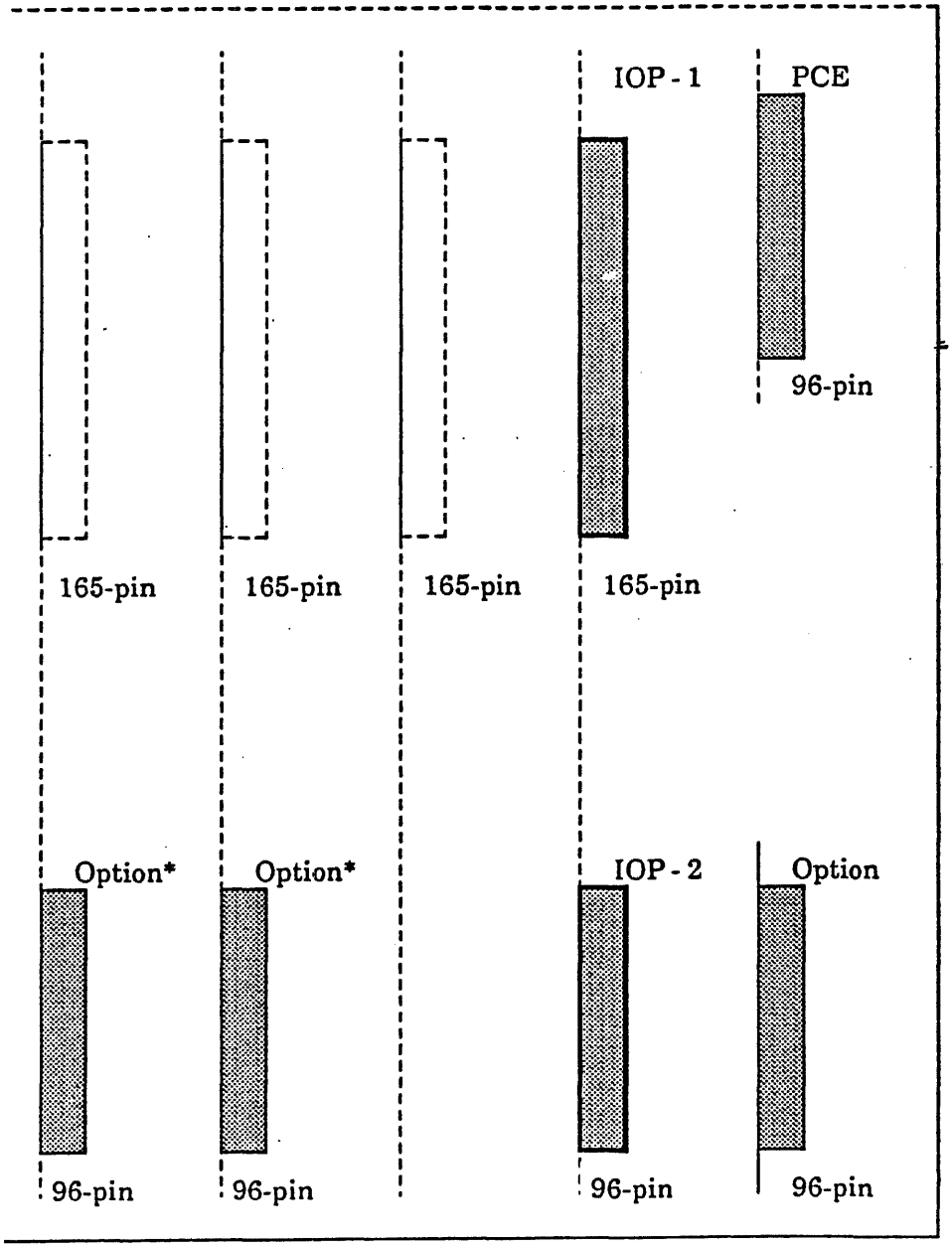


Figure 1.2. I/O subsystem backplane

Figure 1.3 illustrates the board layout for the main IOP board. Board layouts for the other boards is given in the manual describing those options. (To be provided)

Figure 1.3. IOP board layout

1.1.2 Interfaces to Backplane

Table 1.1 lists the IOP board 165-pin interface on the backplane. On the backplane, pins are grouped in six rows of three columns each. The table reflects the grouping, and is the front view of the backplane; that is the side from which the boards plug in.

Table 1.2 lists the 96-pin IOP expansion channel interface on the backplane. The table follows the same conventions as Table 1.1.

Backplane interfaces of the PCE and Options boards are presented in the manuals for those options.

Table 1.1. IOP Backplane Pin Assignment (Front View).

Outmost — Inmost

1) 80186 Bus

Spare-1 J4.001	GND J4.002	Spare-2 J4.003
A/AD.07 (bi) J4.004	A/DT/R' (tri, o) J4.005	A/AD.15 (bi) J4.006
A/AD.06 (bi) J4.007	A/DEN' (tri, o) J4.008	A/AD.14 (bi) J4.009
A/AD.05 (bi) J4.010	GND J4.011	A/AD.13 (bi) J4.012
A/AD.04 (bi) J4.013	A/MemRdy (i) J4.014	A/AD.12 (bi) J4.015
A/AD.03 (bi) J4.016	A/ALE' (i) J4.017	A/AD.11 (bi) J4.018
A/AD.02 (bi) J4.019	A/IOPMemWr' (o) J4.020	A/AD.10 (bi) J4.021
A/AD.01 (bi) J4.022	Spare-3 J4.023	A/AD.09 (bi) J4.024
A/AD.00 (bi) J4.025	GND J4.026	A/AD.08 (bi) J4.027
GND J4.028	A/CLK (o) J4.029	VCC J4.030

J4.031 GND	J4.032 GND	J4.033 GND
J4.034 (tri, o) A/AA.19	J4.035 (tri, o) A/S.2'	J4.036 (tri, o) A/AA.23
J4.037 (tri, o) A/AA.18	J4.038 (tri, o) A/S.1'	J4.039 (tri, o) A/AA.22
J4.040 (tri, o) A/AA.17	J4.031 (tri, o) A/S.0'	J4.042 (tri, o) A/AA.21
J4.043 (tri, o) A/AA.16	J4.044 (tri, o) A/BHE'	J4.045 (tri, o) A/AA.20
J4.046 Reserved-0	J4.047 GND	J4.048 (o) A/IOR'
J4.049 (o) A/EPROMCs'	J4.050 (o) LocRamCS'	J4.051 Spare-4
J4.052 (o) IOPLockO'	J4.053 GND	J4.054 (o) A/IOW'
J4.055 (i) A/PCHldAtoArb	J4.056 (i) A/IOPMemRd'	J4.057 Spare-5
J4.058 –5V	J4.059 –5V	J4.060 –5V

GND J4.061	GND J4.062	GND J4.063
Spare-6 J4.064	A/IOP-PCS2' (o) J4.065	A/IOPResetPC' (o) J4.066
Spare-7 J4.067	A/IOP-PCS3' (o) J4.068	GND J4.069
SpareID' (i) J4.070	GND J4.071	A/ArbHoldPC (o) J4.072
DBRK/Daisy' (i) J4.073	IOP-PCE-spare1 (i) J4.074	Reserved-5 J4.075
A/PEINT' (i) J4.076	IOP-OCE-spare2 J4.077	MPB-DCM-spare2 J4.078
A/MEBIntr' (i) J4.079	GND J4.080	A/PCESpker' (i) J4.081
A/VREINT' (o) J4.082		
GND J4.085		
A/RawCLK (o) J4.088		

2) IOP-PCE

3) Rigid Disk

RD1-RdData – J4.083	RD1-WrData – J4.084
RD1-RdData + J4.086	RD1-WrData + J4.087
GND J4.089	VCC J4.090

- more -

Table 1.1. IOP Backplane Pin Assignment (continuation)

Outmost — Inmost

4) IOP-Mesa

J4.091	GND	J4.092	GND	J4.093	GND
J4.094 (o)	A/IOPIntMP'	J4.095 (o)	CSWREN	J4.096	RD1-DriveSeled'
J4.09 (i)	A/MPIntIOP	J4.098 (o)	CSLOAD/SHIFT'	J4.099	RD0-RdData - [18]
J4.100 (o)	A/IOPRdNIA	J4.101 (o)	CSBUFFEREN	J4.102	RD0-RdData + [17]
J4.103 (o)	A/Halt'	J4.104 (o)	CSDATAIN	J4.105	RD0-WrData - [14]
J4.106	IOP-S-spare1	J4.107 (o)	CSSHIFTCLK	J4.108	RD0-WrData + [13]
J4.109 (o)	A/ResetMPB'	J4.110 (i)	CSDATAOUT	J4.111	RD0-DrvSel' [1]
J4.112	VCC	J4.113	VCC	J4.114	VCC
J4.115	VCC	J4.116	VCC	J4.117 (i)	VCC
J4.118	VCC	J4.119	VCC	J4.120	VCC

3) Rigid Disk (continued)

	GND J4.121		GND J4.122		GND J4.123
	GND J4.124		GND J4.125		GND J4.126
	GND J4.127		GND J4.128		GND J4.129
[34]	RD/DirIn' J4.130	[30]	RD/DrvSel2' J4.131	[28]	RD/DrvSel1' J4.132
[26]	RD/DrvSel0' J4.133	[32]	Rd/DrvSel3' J4.134	[24]	RD/Step' J4.135
[22]	RD/Ready' J4.136	[20]	RD/Index' J4.137	[18]	RD/HeadSel1' J4.138
[14]	RD/HeadSel0' J4.139		GND J4.140	[12]	RD/WriteFault' J4.141
[10]	RD/Track00' J4.142	[8]	Rd/SeekComp' J4.143	[6]	RD/WriteGate' J4.144
[4]	Rd/HeadSel2' J4.145		GND J4.146	[2]	RD/ReduceWrI' J4.147
	-12V J4.148		-12V J4.149		-12V J4.150

J4.151	GND	J4.152	GND	J4.153	GND
J4.154	GND	J4.155	GND	J4.156	GND
J4.157	+ 12V	J4.158	+ 12V	J4.159	+ 12V
J4.160 (i)	POWERNORMAL	J4.161 (o)	A/Reset'	J4.162 (o)	LED1
J4.163 (i)	BootButton	J4.164 (o)	LED3	J4.165 (o)	LED2

Number in brackets [n] is the pin number for the rigid disk drive connecter.

Table 1.2. IOP Expansion Channel Pin Assignment

Outmost				Inmost	
A.15	J13.1	Reset-Exp'	J13.2	A.07	J13.3
A.14	J13.4	GND	J13.5	A.06	J13.6
A.13	J13.7	8MHz-Exp	J13.8	A.05	J13.9
A.12	J13.10	GND	J13.11	A.04	J13.12
A.11	J13.13	I/ORd'	J13.14	A.03	J13.15
A.10	J13.16	GND	J13.17	A.02	J13.18
A.09	J13.19	I/OWrH'	J13.20	A.01	J13.21
A.08	J13.22	Exp-spare1	J13.23	A.00	J13.24
Exp-spare2	J13.25	IOPALEn	J13.26	Exp-spare3	J13.27
VCC	J13.28	VCC	J13.29	VCC	J13.30

J13.31	GND	J13.32	GND	J13.33	GND
J13.34	Data.15	J13.35	IOPDEn'	J13.36	Data.07
J13.37	Data.14	J13.38	I/ORdv	J13.39	Data.06
J13.40	Data.13	J13.41	I/OWrL'	J13.42	Data.05
J13.43	Data.12	J13.44	GND	J13.45	Data.04
J13.46	Data.11	J13.47	IOPDT/R'	J13.48	Data.03
J13.49	Data.10	J13.50	ExpChanSel'	J13.51	Data.02
J13.52	Data.09	J13.53	GND	J13.54	Data.01
J13.55	Data.08	J13.56	XIntrReq1	J13.57	Data.00
J13.58	+12V	J13.59	+12V	J13.60	VCC

GND	J13.61	GND	J13.62	GND	J13.63
ExpDmaReq'	J13.64	XIntr-Req2	J13.65	XIntr-Req5	J13.66
Reserved	J13.67	XIntrReq3	J13.68	XIntrReq6	J13.69
XIntrRe00	J13.70	XIntrReq4	J13.71	XintrReq7	J13.72
-12V	J13.73	-12V	J13.74	-12V	J13.75
Exp-Spare4	J13.76	Exp-Spare5	J13.77	Reserved-1	J13.78
Exp-Spare6	J13.79	GND	J13.80	Reserved-2	J13.81
Exp-spare7	J13.82	Exp-spare8	J13.83	Reserved-3	J13.84
Exp-spare9	J13.85	GND	J13.86	Reserved-4	J13.87
-5V	J13.88	-5V	J13.89	VCC	J13.90

J13.91	GND	J13.92	GND	J13.93	GND
J13.94	Exp-spare10	J13.95	Exp-spare11	J13.96	Exp-spare12

Reserved-1 = ResetExpChan4
Reserved-2 = ResetExpChan3
Reserved-3 = ResetExpChan2
Reserved-4 = ResetExpChan1

1.1.3 Power

Power requirements for the main IOP board - TO BE PROVIDED

2 IOP Processor

The IOP processor is based on the 8 MHz Intel 80186 microprocessor and uses a traditional microprocessor bus architecture that is expanded by the special functions necessary to support the Dove I/O requirements.

Figure 2.1 illustrates the basic components of the Intel 80186 used as the I/0 Processor (IOP).

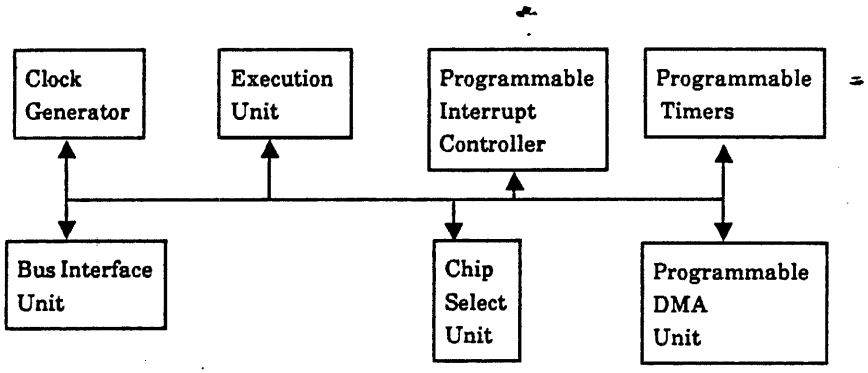


Figure 2.1. IOP 80186 block diagram

Features of this subsystem include:

- access to its local memory as well as to the main system memory via the memory controller.
- fully interrupt driven.
- structure can be extended through additional option slots.
- supports three external bus masters.
- controls and communicates with the Mesa processor. The control function involves resetting, halting, and starting the Mesa processor, as well as reading and writing the control store.
- interrupts the Mesa processor and may be interrupted by the Mesa processor.

2.1 Hardware

The IOP processor consists of a 80186 microprocessor chip. Figure 2.x (to be provided) illustrates the 80186 chip; Table 2.1 lists the pins and signals for this chip.

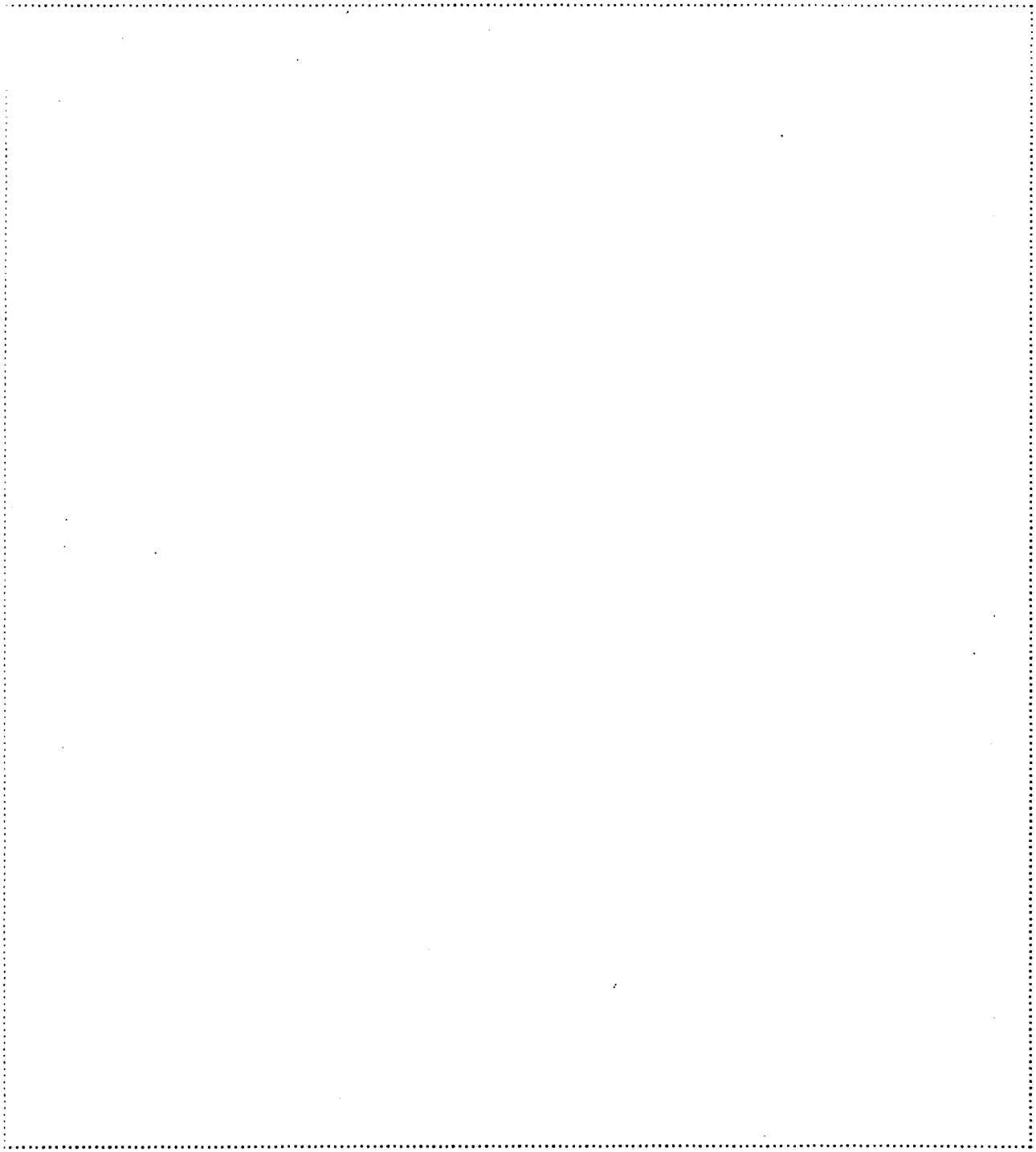


Figure 2.x. 80186 pins and signals (TO BE PROVIDED)

Table 2.1 80186 Pin Assignment

(Reprinted by permission of Intel)

Symbol	Function
Vcc, Vcc	System Power: = 5 volt power supply
Vss. Vss	System Ground
Reset	Reset Output indicates that the 80186 CPU is being reset, and can be used as a system reset. It is active HIGH, synchronized with the processor clock and lasts an integer number of clock periods corresponding to the length of the RES' signal.
X1, X2	Crystal Inputs, X1 and X2, provide an external connection for a fundamental mode parallel resonant crystal for the internal crystal oscillator. X1 can interface to an external clock instead of a crystal. In this case, minimizes the capacitance on X2 or drive X2 with complemented X1. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT)
CLKOUT	Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT has sufficient MOS drive capabilities for the 8087 Numeric Processor Extension.
RES'	System Reset causes the 80186 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80186 clock. The 80186 begins fetching instructions approximately 7 clock cycles after RES' is returned HIGH. RES is required to be LOW for greater than 4 clock cycles and is internally synchronized. For proper initialization, the LOW-to-HIGH transition of RES must occur no sooner than 50 microseconds after power up. This input is provided with a Schmitt-trigger to facilitate power-on RES' generation via an RC network. When RES' occurs, the 80188 will drive the status lines to an inactive level for one clock, and then tri-state them.
TEST'	Test' is examined by the wait instruction. If the TEST input is HIGH when "WAIT" execution begins, instruction execution will suspend. Test' will be resampled until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80186 is waiting for TEST', interrupts will be serviced. This input is synchronized internally.
TMR IN O, TMR IN1	Timer inputs are used either as clock or control signal, depending upon the programmed timer mode. These inputs are active HIGH (or Low-to-High transitions are counted) and internally synchronized.
NMI	Non-Maskable interrupt is an edge-triggered input which causes a type 2 interrupt. NMI is not maskable internally. A transition from a LOW to HIGH initiates the interrupt at the next instruction boundary. NMI is latched internally. An NMI duration of one clock or more will guarantee service. This input is internally synchronized.
INTO, INTO1 INT2/INTAO' INT3/INTA1'	Maskable Interrupt Request can be requested by strobing one of these pins. When configured as inputs, these pins are active HIGH. Interrupt requests are synchronized internally. INT2 and INT3 may be configured via software to provide active-LOW interrupt-acknowledge output signals. All interrupt inputs may be configured via software to be either edge-or level-triggered. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowledged When iRMX mode is selected, the function of these pins changes (see interrupt Controller section of this data sheet).
A19/S6, A18/S5, A17/S4, A16/S3	Address Bus Outputs (16-19) and Bus Cycle Status (3-6) reflect the four most significant address bits during T1. These signals are active HIGH during T2, T3, Tw, and T4. Status information is available on these lines as encoded below Low High S6 Processor Cycle DMA Cycle S3, S4, and S5 are defined as Low during T2-T4.

Table 2.1 80186 Pin Assignment

(Reprinted by permission of Intel)

<table>
<tr><th>Symbol</th><th>Function</th></tr>
<tr><td>AD15-AD0</td><td>Address/Data Bus (0-15) signals constitute the time multiplexed memory or I/O address (T1) and data (T2, T3, TW, and T4) bus. The bus is active HIGH. A0 is analogous to BHE' for the lower byte of the data bus, pins D7 through D0. It is LOW during T1 when a byte is to be transferred onto the lower portion of the bus in memory or I/O operations.</td></tr>
<tr><td>BHE'/S7</td><td>During T1, the Bus High Enable signal should be used to determine if data is to be enabled onto the most significant half of the data bus pins D15-D8. BHE' is LOW during T1 for read, write, and interrupt acknowledge cycles when byte is to be transferred on the higher half of the bus. The S7 status information is available during T2, T3, and T4. S7 is logically equivalent to BHE'. The signal is active LOW, and is tri-stated OFF during bus HOLD.
BHE' and A0 Encodings
BHE' Value | A0 Value | Function
0 | 0 | Word Transfer
0 | 1 | Byte Transfer on upper half of data bus (D-15-D8)
1 | 0 | Byte Transfer on lower half of data bus (D7-D0)
1 | 1 | Reserved</td></tr>
<tr><td>ALE/QSO</td><td>Address Latch Enable /Queue Status 0 is provided by the 80186 to latch the address into the 8282/8283 address latches. ALE is active HIGH. Addresses are generated off the rising edge of the CLKOUT immediately preceding T1 of the associated bus cycle, effectively one-half clock cycle earlier that in the standard 8086. The trailing edge is generated of the CLKOUT rising edge in T1 as in the 8086. Note the ALE is never floated.</td></tr>
<tr><td>WR'QS1</td><td>Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. WR' is active for T2, T3, and Tw of any write cycle. It is active LOW, and floats during "HOLD." It is driven HIGH for ONE clock during Reset, and then floated. When the 80186 is in queue status mode, the ALE/QSO and WR'QS1 pins provide information about processor/instruction queue interaction.
QS1 | QS0 | Queue Operation
0 | 0 | No queue operation
0 | 1 | First opcode byte fetched from the queue
1 | 1 | Subsequent byte fetched from the queue
1 | 0 | Empty the queue</td></tr>
<tr><td>RD'QSMD</td><td>Read Strobe indicates that the 80186 is performing a memory or I/O read cycle. RD' is active LOW for T2, T3, and Tw of any read cycle. It is guaranteed not to go LOW in T2 until after the Address Bus is floated. RD' is active LOW, and floats during "HOLD". RD' is driven HIGH for one clock during Reset, and then the output drive is floated. A weak internal pull-up mechanism on the RD line holds it HIGH when the line is not driven. During RESET the pin is sampled to determine whether the 80186 should provide ALE, WR' and RD', or if the Queue-Status should be provided. RD should be connected to GND to provide Queue-Status data.</td></tr>
<tr><td>ARDY</td><td>Asynchronous Ready informs the 80186 that the addressed memory space or I/O device will complete a data transfer. The ARDY input pin will accept an asynchronous input and is active HIGH. Only the rising edge is internally synchronized by the 80186. This means that the falling edge of ARDY must be synchronized to the 80186 clock. If connected to Vcc, no WAIT states are inserted. Asynchronous ready(ARDY) or synchronous ready (SRDY) must be active to terminate a bus cycle.</td></tr>
<tr><td>SRDY</td><td>Synchronous Ready must be synchronized externally to the 80186. The use of SRDY provides a relaxed system-timing specification on the Ready input. This is accomplished by eliminating the one-half clock cycle which is required for internally resolving the signal level when using the ARDY input. This line is active HIGH. If this line is connected to Vcc, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active before a bus cycle is terminated. If unused, this line should be tied LOW.</td></tr>
</table>

Table 2.1 80186 Pin Assignment

(Reprinted by permission of Intel)

Symbol	Function
LOCK'	LOCK' output indicates that other system bus masters are not to gain control of the system bus while LOCK' is active LOW. The LOCK' signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction and is activated at the beginning of the first data cycle associated with the instruction following the LOCK prefix. No prefetches will occur while LOCK' is asserted. LOCK' is active LOW, is driven HIGH for one clock during RESET, and then floated. If unused, this line should be tied LOW.
S0', S1', S2'	Bus cycle status S0'–S2' are encoded to provide bus-transaction information. **80186 Bus Cycle Status Information** S2' S1' S0' Bus Cycle Initiated 0 0 0 Interrupt Acknowledge 0 0 1 Read I/O 0 1 0 Write I/O 0 1 1 Halt 1 0 0 Instruction Fetch 1 0 1 Read Data from Memory 1 1 0 Write Data to Memory 1 1 1 Passive (no bus cycle) The status pins float during "HOLD". S2' may be used as a logical M/IO' indicator, and S1' as a DT/R' indicator. The status lines are driven HIGH for one clock during Reset, and then floated until a bus cycle begins.
HOLD (input) HLDA (output)	HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. HOLD may be asynchronous with respect to the 80186 clock. The 80186 will issue a HLDA (HIGH) in response to a HOLD request at the end of T4 or T1. Simultaneous with the issuance of HLDA, the 80186 will float the local bus and control lines. After HOLD is detected as being LOW, the 80186 will lower HLDA. When the 80186 needs to run another bus cycle, it will again drive the local bus and control lines.
UCS'	Lower Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1k-256K block) of memory. This line is not floated during bus HOLD. The address range activating LCS' is software programmable.
LCS'	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion of memory (8K–512K). This line is not floated during bus HOLD. The address range activating UCS' is a software programmable.
LCS	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K-256K) of memory. This line is not floated during bus HOLD. The address arrange activating LCS' is software programmable.
PCS0' PCS1-4	Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select or to provide an internally latched A1 signal. The address range activating PCS5 is software programmable. When programmed to provide latched A1, rather than PCS5, this pin will retain the previously latched value of A1 during a bus Hold. A1 is active HIGH.
PCS6'/A2	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating PCS6' is software programmable. When programmed to provide latched A2, rather than PCS6, this pin will retain the previously latched value of A2 during a bus HOLD. A2 is active HIGH.
DT/R'	Data Transmit/Receive controls the direction of data flow through the external data bus transceiver. When LOW, data is transferred to the 80186. When HIGH the 80186 places write data on the data bus.
DEN'	Data Enable is provided as a data bus transceiver output enable. DEN' is active LOW during each memory and I/O access. DEN' is HIGH whenever DT/R' changes state.

2.1.1 Clock Generator

The 80186 includes a clock generator and crystal oscillator. The crystal oscillator is used with a parallel resonant, fundamental mode crystal at 2x the desired CPU clock speed, or with an external oscillator also at 2x the CPU speed *(Which one do we use??)*. The output of the oscillator is internally divided by two to provide fifty percent duty cycle CPU clock from which all 80186 system timing derives. The CPU clock is externally available, and all timing parameters are referenced to this externally available signal. The clock generator also provides ready synchronization for the processor. (Is this how we use the clock generator?)

2.1.2 Execution Unit

The execution unit is a 16-bit ALU that performs 8- or 16-bit arithmetic and logical operations. *(Which do we use??)* This unit provides for data movement among the IOP system registers, memory and I/O space. In addition, the CPU allows for high speed data transfer from one area of memory to another using string move instructions to or from an I/O port and memory using block I/O instructions. The CPU also provides a large amount of conditional branch and other control instructions. *(Is this how it applies to out IOP board??)*

2.1.3 Programmable Interrupt Controller

To be provided-introduction lead in here and an explanation of this figure.

Figure 2.2 illustrates the interrupt arrangement for the IOP.

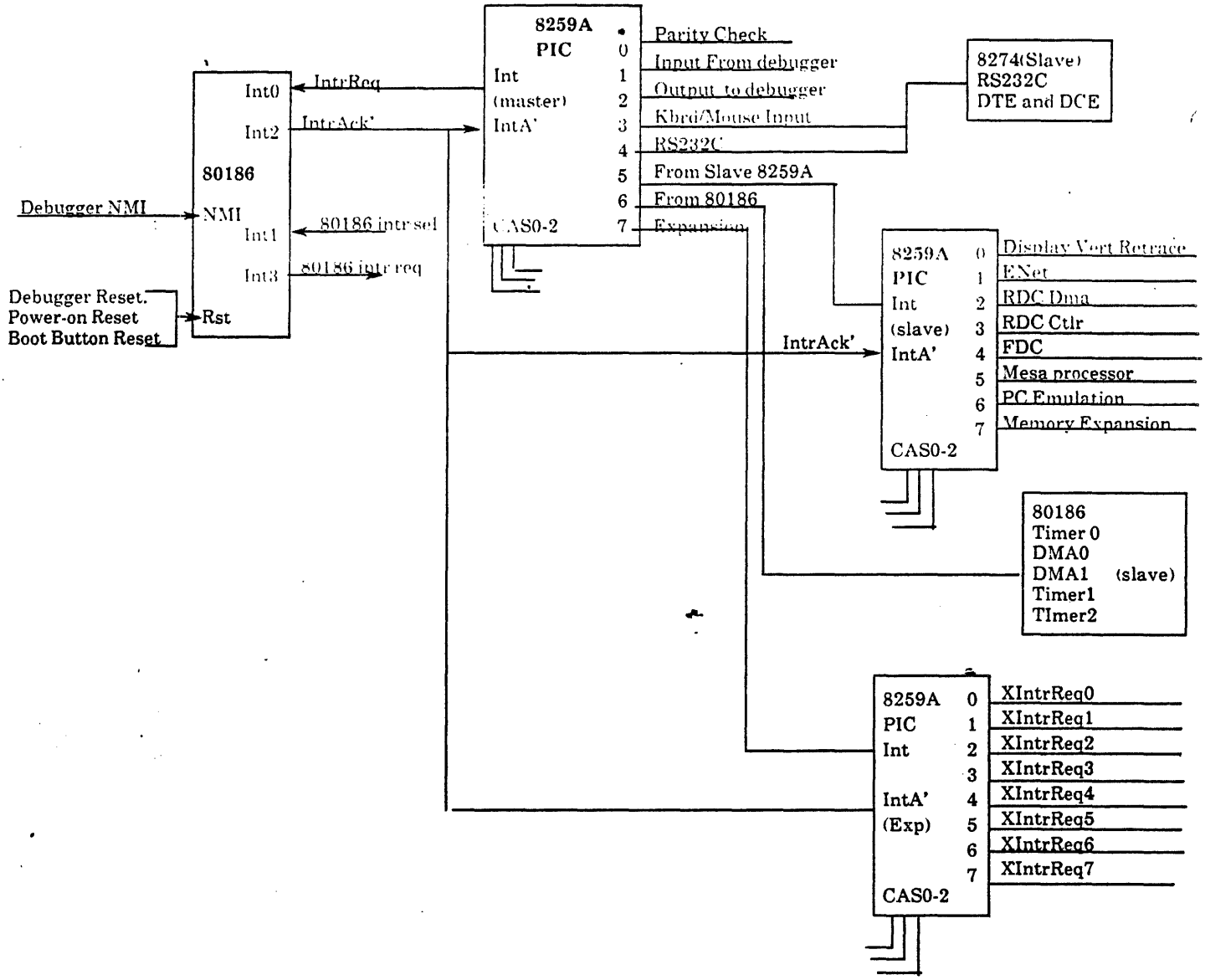


Figure 2.2. IOP interrupt structure

80186 operates in:

iRMX mode

This mode allows the 80186 internal interrupt controller to work as a slave to external Master Interrupt Controllers.

Master PIC 8259A operates in:

non-buffered mode, cascade mode, and special fully nested mode.

Inputs for this mode are edge-triggered.

Slave PIC 8259A operates in:

non-buffered mode, and cascade mode.

Inputs for this mode are edge-triggered.

2.1.3.1 Expansion PIC Vector Types

Up to 256 vectors are allowed in the 80186; the vectors are limited by 8 bits per vector. The vectors include predefined and reserved vectors. Vectors 0-19 are pre-defined for the 80186; vectors 20-31 are reserved by Intel. IOP interrupt types start at 32.

The iRMX mode interrupt vectors for the 80186 internal peripherals are programmable. The 80186 interrupt vector register specifies the 5ms bits. The lower 3 significant bits are determined by the priority level of the internal device that causes the interrupt in the iRMX mode.

2.1.4 Programmable Timers

The 80186 has three programmable timers: Timer 0, Timer 1, and Timer 2.

80186:	Timer0:	Speaker Circuit	(Clock-In = Internal 2 MHz clock)
(Internal timers,	Timer 1:	FDC Terminal Count	(Clock-In = 8272A FDC Dma Ack)
16bits wide.)	Timer2:	Reserved for Operating system	(Clock-In = Internal 2 MHz clock, no extra output)

2.1.5 Bus Interface Unit (To be provided)

2.1.6 Chip Select Unit (To be provided)

2.1.7 Programmable DMA Unit (To be provided)

2.1.8 External and Internal interfacing

A short paragraph is needed here to explain this part of the hardware.

2.2 Theory of Operations

A short paragraph will be provided to explain how the 80186 works in the overall IOP logic.

2.2.1 Initialization and Processor Reset

Processor initialization or startup begins by driving the RES' input pin LOW. RES' forces the 80186 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as RES' is active. After RES' becomes inactive and an internal processing interval elapses, the 80186 begins execution with the instruction at physical location FFF0(H). RES' also sets some registers to predefined values as shown in Table 2.2.

Table 2.2. 80186 Initial Register State after Reset

Name	Location
Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

Notes: Fetch 1st instruction at CS : IP = FFFF :0

Relocation register should = 040FF for iRMX mode and peripheral control blocks to start at 0FF00H in I/O space.

2.2.2 Local Bus Controller and Reset

After receiving a reset pulse from the RES' input, the local bus controller performs the following six actions:

1) Drive DEN', RD', and WR' HIGH for one clock cycle, then float. RD' are also provided with an internal pull-up device to prevent the processor from inadvertently entering Queue Status mode during reset.

2) Drive S0'-S2' to the passive state (all HIGH) and then float.

3) Drive Lock' HIGH then float.

4) Tri-state AD0-15, A16-19, BHE' DT.R'.

5) Drive ALE LOW (ALE is never floated).

6) Drive HLDA LOW. If Hold = H after Reset' = H, then HLDA will go high 2 clock cycles later.

2.2.3 Chip Select/Ready Logic and Reset

After reset occurs, the Chip Select/Ready Logic performs the following actions.

1) All chip-select outputs are driven HIGH.

2) After leaving RESET, the UCS' line is programmed to provide chip selects to a 1K block with the accompanying READY control bits at 011 to allow the maximum number of internal wait states in conjunction with external Ready consideration; that is, UMCS resets to FFFBH.

3) No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the PCS' lines will become active.

2.2.4 DMA Channels and Reset

After Reset occurs, the DMA channels perform the following actions:

1) The Start/Stop bit for each channel is reset to STOP.

2) Any transfer in progress is aborted.

2.2.5 Interrupt Controller and Reset

After RESET occurs, the interrupt controller performs the following actions.

1) All SFNM bits reset to 0, implying Fully Nested Mode.

2) All PR bits in the various control registers set to 1. This places all sources at the lowest priority. (level 111).

3) All LTM bits reset to 0, resulting in edge-sense mode.

4) All interrupt Service bit reset to 0.

5) All MSK (Interrupt Mask) bits set to 1 (mask).

6) All C (Cascade) bits reset to 0 (non-cascade).

7) All PRM (Priority Mask) bits set to 1, implying no levels masked.

8) Initialized to non-iRMX 86 mode.

2.2.6 Timers and Reset

After Reset, the timers will perform the following actions:

1) All EN (Enable) bits are reset, preventing timer counting.

2) All SEL (Select) bits are reset to zero. This selects MAX COUNT register A, resulting in the Timer Out pins going HIGH upon RESET.

2.2.7 Ready Logic

The IOP is a Normally-Ready system. The 80186 only uses the ARdy input; SRdy is not used. ARdy is separated into MemRdy and I/ORdy in order to reduce loadings and faster rise time. I/ORdy is connected to the Expansion Slots only as all expansion devices must be in I/O space.

2.2.8 Reset Logic

Power Normal becomes high between 50 and 250 microseconds after all power supply outputs exceed 94 percent. Pressing the Boot Button will generate a low pulse of ~ 0.5 s. Software resets of the peripheral controllers should be at least 20 µs long reset. When the floppy disk controller is reset or during machine power up, the write current is cut-off. When the rigid disk controller is reset or during machine power up the write current is cut off.

2.3 Programmer Interface

A short paragraph will be provided here to explain the programming interface.

2.3.1 Memory Mapping <Daisy>

The 80186 processor can directly address only one of the four Mbytes in Daisy. An address mapping circuit built into each A chip allows indirect access to an expanded real memory space. The mapping circuit provides two substantially different translations between the logical and real address spaces, and is programmable via I/O space transfers from the IOP 80186.

The logical address space is divided into eight banks of 128 Kbytes. Maps in the A chips are programmed so that exactly one A chip responds to addresses in a particular bank. That A chip maps the logical bank to one of its eight real banks. The logical-to-real translation is programmable and is independent for each bank within each A chip. The net effect is that addresses in a particular logical bank may refer to the real banks in the A chips.

High speed I/O devices can directly access all of real memory. The Intel Ethernet controller and the Dove rigid disk controller can each supply 24 bits of address, sufficient to address all four MBytes of real memory. The devices can also use either of the 80186 maps via special combinations of the upper four address bits.

Bus masters other than the 80186 (for example, the Intel 82586 integrated Ethernet controller) provide 24 bits of address. The devices address all of memory directly and can mix unmapped and mapped memory references, using special combinations of the upper four address bits.

2.3.2 RAM/ROM Access <Daisy>

Local RAM in Daisy is accessed via the 80186 maps. Access to the local ROM is restricted to the IOP 80186.

The 80186 has free access to the local RAM and ROM. The 80186 accesses the local ROM by asserting UCS' (active low), which prevents the A chips from responding. The A chips provide for access to local RAM by examining the IOP bus address, and then generating LCS' (active low) when the address falls in the local RAM address space. The A chips do not otherwise respond to local RAM addresses.

Local RAM is accessible to other bus masters when they make mapped references; addresses that fall in the local RAM space access local RAM. Since only the 80186 can control the UCS' line, the other bus masters cannot access local ROM.

The following figures illustrate the addresses and registers used in the IOP processor. Text to explain these figures will be provided.*

- Figure 2.3 illustrates the memory address space.
- Figure 2.4 illustrates the memory address space bit assignments.
- Figure 2.5 illustrates the IOP address space.
- Figure 2.6 illustrates registers in the IOP address space.
- Figure 2.7 illustrates the integrated peripheral control block.
- Figure 2.8 illustrates the relocation register layout.
- Figure 2.9 illustrates chip select control registers.

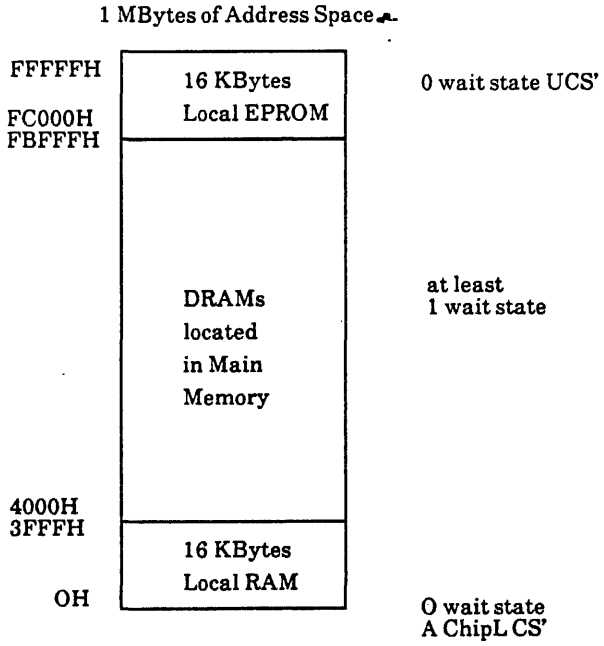


Figure 2.3. 80186 memory address space

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1M	512K	256K	128K	64K	32K	16K	8K	4K	2K	1K	512K	256	128	64	32	16	8	4	2

Figure 2.4. IOP 80186 memory address space bit assignment

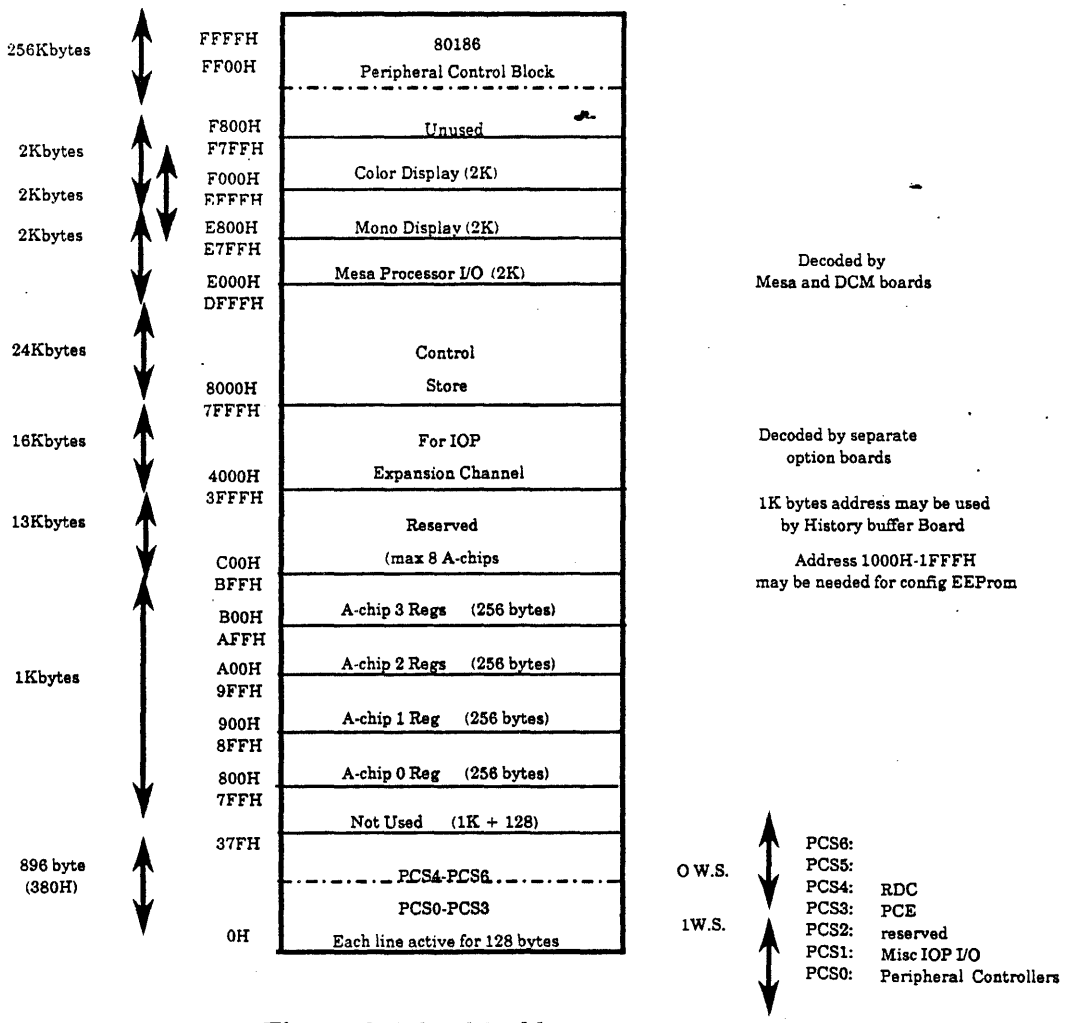


Figure 2.5. 80186 address space

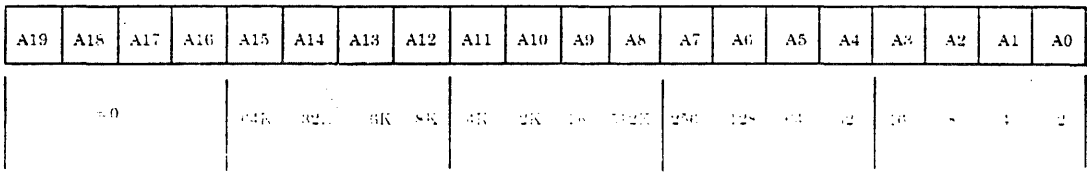


Figure 2.6. IOP 80186 address space

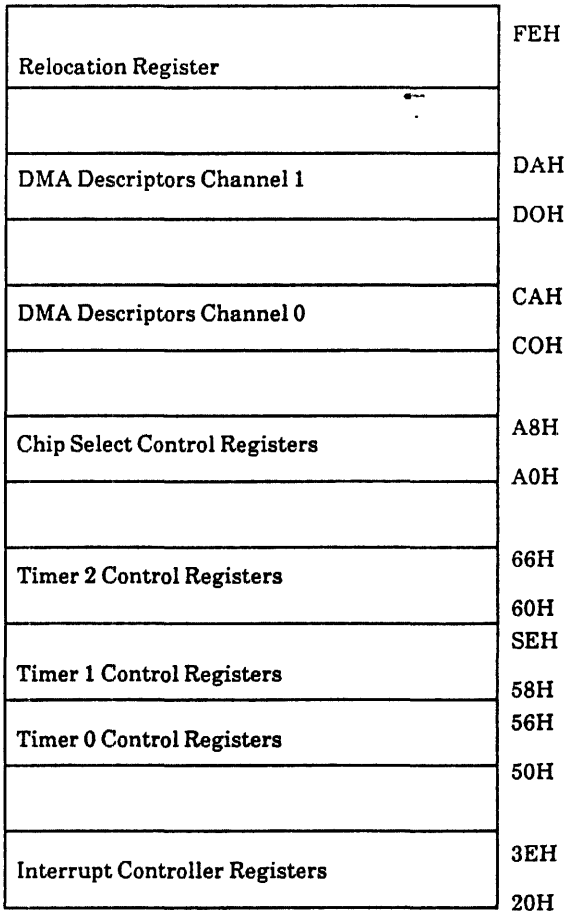
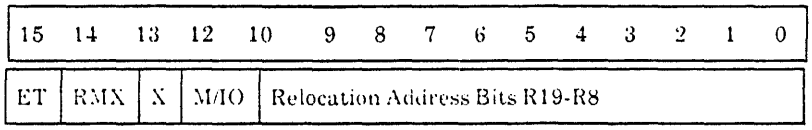


Figure 2.7. 80186 integrated peripheral control block



ET = ESC Trap/ No ESC Trap (1/0)
M/IO = Register block located in Memory, I/O Space (1/0)
RMX = Master Interrupt Controller mode/iRMX compatible Interrupt Controller mode (0/1)

Figure 2.8. 80186 relocation register layout

OFFSET:

A0H	Upper Memory Size				1	UMCS
A2H	Lower Memory Size				2	LMCS
A4H	Peripheral Chip Select Base Address				3	PACS
A6H	Mid-Range Memory Base Address				4	MMCS
A8H	Mid-Range Memory Size	EX	MS		5	MPCS

6

1. Upper memory ready bits
2. Lower memory ready bits
3. PCS0-PCS3 ready bits
4. Mid-range memory ready bits
5. PCS4-PCS6 ready bits
6. MS: 1 = Peripherals active in memory space
   0 = Peripherals active in I/O space
   EX: 1 = 7 PCS lines
   0 = PCS5 = A1, PCS5 = A1, PCS6 = A2

Not all bits of every field are used

Tables 2.3 and 2.4 list the controller addresses for the IOP processor..

Table 2.3. IOP I/O Controller Addresses: PSC0

PCS.0

Address (Hex)	Read	Write
8259A Master Intr Ctlr		
0	IRR,ISR	ICW1,OCW2OCW3
2	IMR	ICW2, ICW3,ICW4, OCW1
8259A Slave Intr Ctlr		
10	IRR, ISR	ICW1, OCW2, OCW3
12	IMR	ICW2, ICW3, ICW4, OCW1
8254 Timer		
20	Timer Counter 0	Timer Counter 0
22	Timer Counter 1	Timer Counter 1
24	Timer Counter 2	Timer Counter 2
26	———	Timer mode
8251 Uart (Keyboard/Mouse)		
30	Uart Rx Data	Uart Tx Data
32	Uart Status	Uart Ctl Word
8274 RS232C Ctlr		
40	Ch A Rx Data	Ch A Tx Data
42	Ch A Status	Ch A Cmd/Parm
44	Ch B Rx Data	Ch B Tx Data
46	Ch B Status	Ch B Cmd/Parm
8272A FDC		
50	FDC Main Status Reg	(illegal)
52	FDC Data Reg Stack	FDC Data Reg Stack
54	Dma Read from FDC	Dma Write to FDC
56	Terminal Count to FDC	(for polling only)
8259A Expansion Intr Ctlr		
60	IRR, ISR	ICW1, OCW2, OCW3
62	IMR	ICW2, ICW3, ICW4, OCW1
8255A-5 PIOBurdock Interface		
70	PortA	Port A
72	Port B	Port B
74	Port C	Port C
76	(illegal)	Control Word

Table 2.4. IOP I/O Controller Addresses: PSC1

	PCS.1	
Address(Hex)	Read	Write
80	Input Port	Control Reg
90	Host Address PROM (90, 92, 94, 96, 98, 9A, 9C, 9EH)	LED
A0	Clear Ring Latch	ENet Attn
B0	Clear Mesa Interrupt Latch	Reserved (May be used for WrConfigReg')
D0	Clear Vertical Retrace Interrupt Latch	
E0	X	Allow PCCmd'
F0	HoldIOPCmd'	AllowRDCCmd'

Figure 2.10 illustrates the I/O system address space.

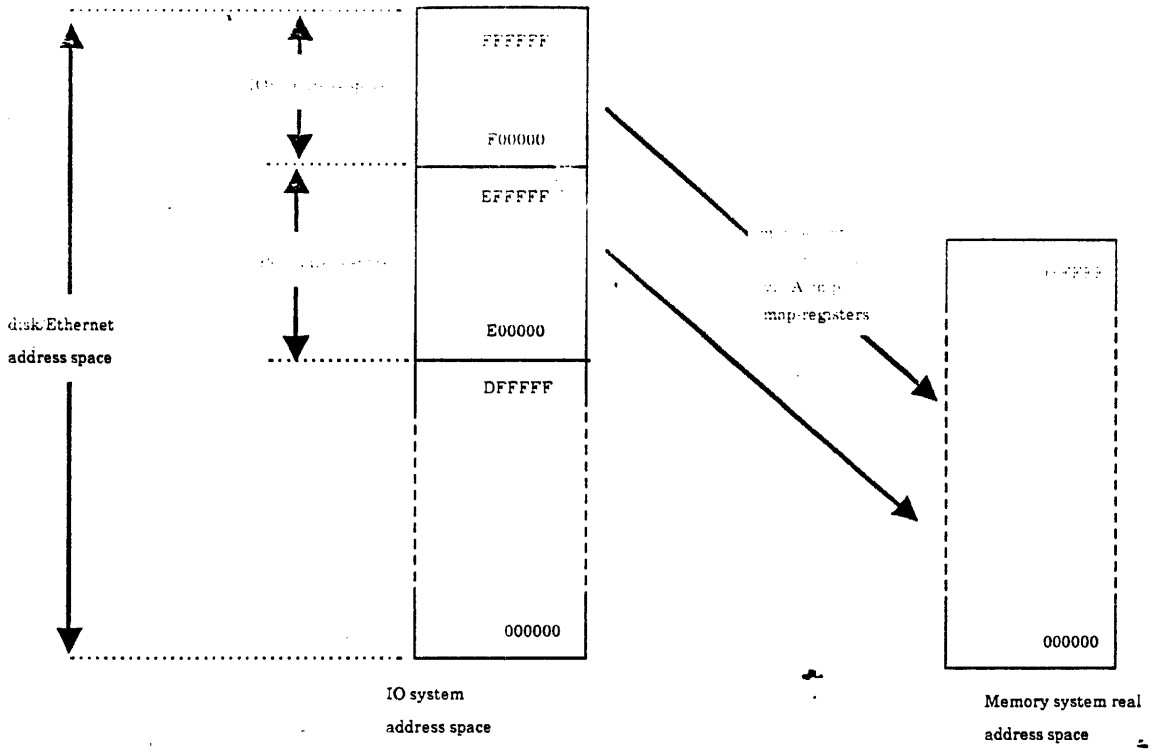


Figure 2.10. The I/O system address space

Table 2.5 lists register bit assignments. Additional text To Be Provided.

2.3.3 Timing- TO BE PROVIDED

Table 2.5. Register Bit Assignment

	Read			Write				
	Input Port 80H	Host Addr PROM (9xH)		Control Reg (80H)	LED (90H)	Mesa Proc/Control Store Reg (B0H)	Reset Control Reg (C0H)	EEProm Config Reg (D0H)
Bit 15	8272A FDC Interrupt request				LED digit 3	Control Store Wr Enable	Enable Ethernet Int Loopback	Serial EEProm Data-in
Bit 14	8272A FDC Dma request			Enable 186 Timer 0	LED digit 3	Control Store Load/Shift'	Enable Keyboard Int Loopback	
Bit 13	Data From Control Store (1 bit only)			FD Motor on	LED digit 3	ControlStore Buffer Enable	reserved	Serial EEProm Clock
Bit 12				FD InUse	LED digit 3	Data To Control Store (1 bit only)	reserved	Serial EEProm Enable
Bit 11	Data From Config EEProm (1 bit only)			Allow Timer 1 to generate 8272A TC in real Dma	LED digit 2	Control Store Shift Clock	reserved	reserved
Bit 10	RS232 Ch A DSR' (L active)			FD H = Low speed L = High Speed	LED digit 2	IOPRdNIA	Reset ExpansionChannel'	Diag LED3
Bit 9	RS232 Ch A Ring Indicator' (L active)			Sel RS232 Ch A to use internal clock	LED digit 2	Halt Mesa Proc' (L active)	Reset Dma-FIFO'	Diag LED2
Bit 8	RS232 Ch B DTR' (Low active)			Enable RS232 Ch B to send clock signals	LED digit 2	Interrupt Mesa Processor	Reset RDC'	Diag LED1
Bit 7	Speaker Timer			FD Drive Sel 4	LED digit 1		Reset PCE-80186'	
Bit 6	Serial EEPromRdy			FD Drive Sel 3	LED digit 1		Reset Mesa Proc'	
Bit 5	DipSw.5	Host Address		FD Drive Sel 2	LED digit 1		Reset Keyboard'	
Bit 4	DipSw.4	PROM connected		FD Drive Sel 1	LED digit 1		Reset Burdock Ctlr 8255-5'	
Bit 3	DipSw.3	to Bit 07-00		FD H = 5¼"	LED digit 0		Reset Keyboard Uart 8251A'	
Bit 2	DipSw.2	only		FD PreComp 2	LED digit 0		Reset FDCtlr 8272A'	
Bit 1	DipSw.1			FD PreComp 1	LED digit 0		Reset RS232 Ctlr 8274'	
Bit 0	DipSw.0			FD PreComp 0	LED digit 0		Reset ENet Ctlr 82586'	

3 Local Memory

3.1 Hardware	3.2 Theory of Operations	3.3 Programmer Interface 3.3.1 EPROM 3.3.1.1 Booting Proceedures 3.3.1.2 Initialization Software 3.3.1.3 Diagnostics 3.3.1.4 Debugger Kernal 3.3.2 RAM 3.3.2.1 Operating System Software 3.3.2.2 Interrupt Vector Table 3.2.2.3 Local Vectoring

The IOP contains a small amount of local memory. The local memory consists of 16 Kbytes of EProm with 0 wait state and 16 Kbytes of static RAM with 0 wait state. The EProm contains booting and initialization software, minimal diagnostics, and the debugger kernel. The RAM is used for the operating system software, the interrupt vector table, and for local buffering.

3.1 Hardware

To be provided.

3.2 Theory of Operations

To be provided.

3.3 Programmer Interface

To be provided.

3.1.1 EPROM

To be provided.

3.1.1.1 Booting Procedures

To be provided.

3.1.1.2 Initialization Software

To be provided.

3.1.1.3 Diagnostics

To be provided.

3.1.1.4 Debugger Kernal

To be provided.

3.3.2 RAM

To be provided.

3.3.2.1 Operating System Software

To be provided.

3.3.2.2 Interrupt Vector Table

To be provided.

3.3.2.3 Local Buffering

To be provided.

4 Bus Arbiter and Mode Control

The external arbiter determines use of the system address/data buses for the rigid disk controller or DMA controller after the IOP or PCE relinquishes the bus. A function called mode control is required to switch the bus between the IOP 80186 and the PCE 80186 execution.

Figure 4.1 illustrates the data flow of the arbiter and mode control.

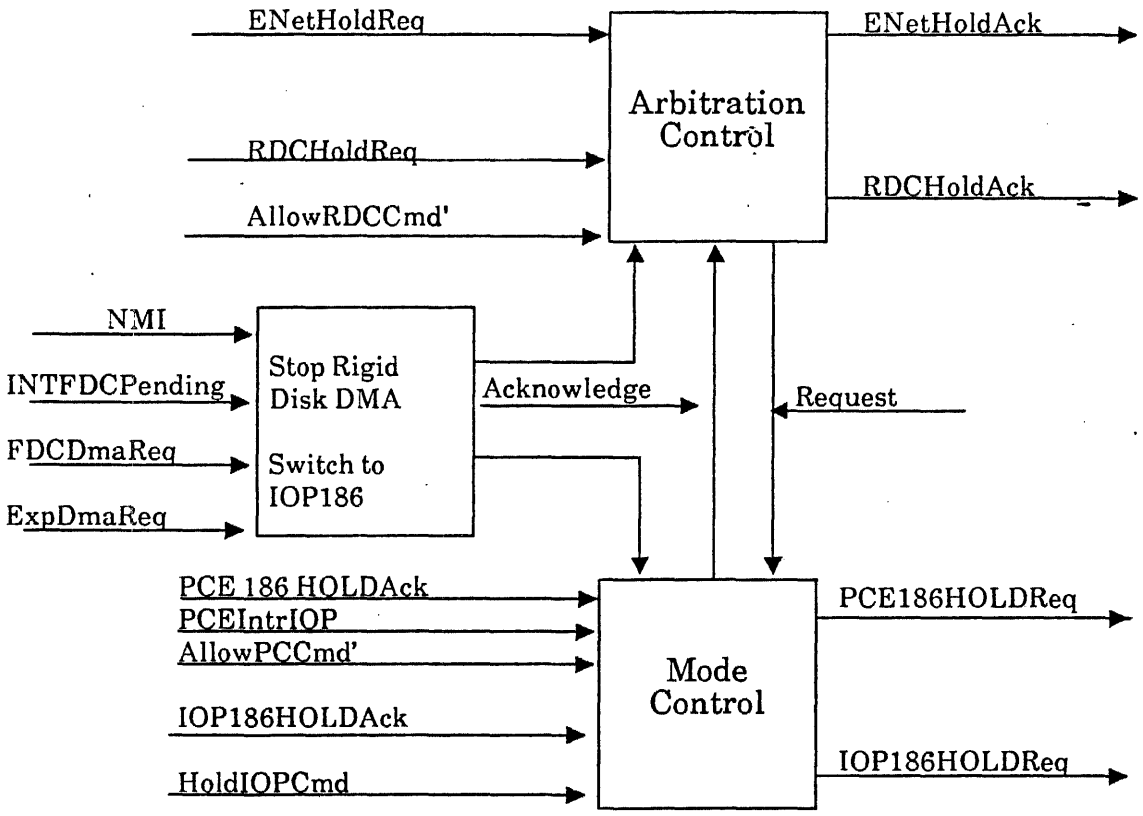


Figure 4.1. Arbitration and mode control block diagram

The bus arbiter functions are:

- accepts requests from the rigid disk DMA and Ethernet controllers.
- determines the highest priority device; Ethernet is given first priority, the rigid disk DMA is given second.

- passes the hold request to the mode control function.
- passes hold acknowledge to the controller being granted the bus.
- suspends the rigid disk DMA activity when the Ethernet requests service.

The mode control functions are:

- determines which of the two 80186s should be bus master.
- passes the hold request from the arbiter to the current 80186 bus master.
- passes hold from the bus master to the arbiter.
- controls the IOP to PCE bus switch.

4.1 Hardware

The IOP bus arbiter is 16 bits wide. An 8 MHz clock produces 125 ns T-states. A minimum of 4 T-states (T1, T2, T3, and T4) are required per bus cycle. A minimum bus cycle is 500 ns.

Table 4.1 lists (to be provided) the internal signals of the arbiter and mode controller and describes their function.

Table 4.1. Arbiter and Mode Control Signal Description

Signal Name	Connection to	Function
ENetHoldReq	(fill in please)	(fill in please)
RDCHoldReq		
AllowRDCCmd'		
ENetHoldAck		
RDCHoldAck		
NMI		
INTFDCPending		
ExpDmaReq		
PCE 186 HOLDAck		
PCEIntrIOP		
AllowPCCmd'		
IOP186HOLDAck		
HoldIOPCmd		
PCE186HOLDReq		
IOP186HOLDReq		
ENetHoldAck		
RDCHoldAck		

4.2 Theory of Operations/Programmer Interface

The following subsections describe the events that occur when service requests are made to the arbiter or mode control by the rigid disk DMA, Ethernet and PC emulator. *(Are these the only operations for the arbiter and mode control?)* Figures 4.2 – 4.5, at the end of this section, illustrate the flow charts for the service requests described in the following subsections.

Certain signals referred to in the text as "condition" signals affect service requests. These signals are NMI, IntrReqToArb, the FDCDmaReq, and ExpDmaReq. The IOP 80186 microprocessor is referred to as the IOP.

4.2.1 Service Request: Ethernet

When the Ethernet requests service, the following two conditions must be present.

- No condition signal may request service from the IOP.
- The PCE option is not operational.

The following sequence describes system operation after an Ethernet service request.

1) The Ethernet controller receives information from the IOP or transmits information over the Ethernet, and then issues a bus request (HLD) to the arbiter.

2) The arbiter passes this request to the mode controller. In turn, the mode controller issues a hold signal to the IOP.

3) When the IOP recognizes the hold request, it enters a machine cycle boundary, and issues a hold acknowledge to inform the arbiter that it is letting go of the system address and data buses.

4) The arbiter controller informs the Ethernet controller, via a hold request, that the requested buses are now available for use. The Ethernet then begins fetching or storing information in the system memory.

5) When the Ethernet no longer requires the buses, it deactivates its hold request in the arbiter. In turn, the arbiter issues a deactivate hold request to the mode controller. The mode controller then deactivates arbiter hold request in the IOP. At the same time, the arbiter issues a two clock delay before processing another Ethernet request.

4.2.2 Service Request: Rigid Disk DMA

When the rigid disk DMA requests service, the following four conditions must be present.

- The I/O command, AllowRDCCmd, was executed by the IOP before the rigid disk DMA activated a hold request.
- During rigid disk DMA information transfer, the Ethernet requests service.
- No condition signal may request service from the IOP.
- The PCE option is not operational.

The following sequence describes the system operation after a rigid disk DMA service request.

1) The IOP instructs the rigid disk DMA to sense data in the rigid disk DMA.

2) The rigid disk DMA fetches or stores the data to different parts of its subsystem. The RDC issues a hold request to the arbiter.

3) The arbiter passes the hold request to the mode controller. The mode controller issues a hold request to the IOP.

4) When the IOP receives the hold request, it goes to a machine cycle boundary and issues a hold acknowledge to the arbiter. The arbiter acknowledges the rigid disk DMA's request by activating RDC hold acknowledge.

5) If the Ethernet activates a hold request while the RDC is using the bus, the arbiter deactivates RDC hold acknowledge and waits for the RDC to deactivate its hold request. When this request is no longer active, the arbiter issues an Ethernet hold acknowledge. The Ethernet deactivates its hold acknowledge when the bus is no longer needed.

Note: If the rigid disk *DMA* reactivates a hold request, the arbiter reactivates a hold acknowledge. Otherwise, the arbiter instructs the mode controller to deactivate its hold request to the IOP so normal IOP processing can be resumed.

4.2.3 Service Request: Rigid Disk DMA Controller

For the rigid disk DMA controller to request service the following four conditions must be present.

- The I/O command, AllowRDCCmd, was executed by the IOP before the RDC? activated a hold request.
- During rigid disk DMA information transfers, the Ethernet requests service.
- Before the Ethernet transfers are complete, a condition signal requests service from the IOP.
- The PCE option is not operational.

The following sequence describes the system operation after a rigid disk DMA controller service request.

1) *(?)* fetches data from or stores data to the rigid disk DMA. The DMA controller senses data in the rigid disk DMA *(What do you mean by the word sense)*, and issues a hold request to the arbiter.

2) The arbiter then passes the hold request to the mode controller. The mode controller issues a hold request to the IOP.

3) When the IOP receives the hold request, it goes to a machine cycle boundary and activates a hold acknowledge. The arbiter in turn acknowledges the rigid disk DMA's request for service by issuing an hold acknowledge.

4) The Ethernet requests service while the RDC is using the bus. The arbiter deactivates hold acknowledge and waits for the rigid disk DMA to deactivate hold request.

5) Once the hold request is deactivated, the arbiter activates an Ethernet hold acknowledge. When the Ethernet controller no longer needs the bus, it deactivates its hold.

6) If the RDC requests service while the Ethernet using the bus, then the arbiter reactivates an RDC hold acknowledge for one transfer, provided the DMA reactivates the RDC hold acknowledge. After one DMA transfer occurs, the arbiter deactivates hold acknowledge, and waits for the DMA to deactivate its hold request.

When the hold request is deactivated, the arbiter in turn deactivates its hold. This signal allows processing of the requested service.

7) When the IOP completes service for the requested interrupt, it again executes a condition signal. The arbiter will not honor more rigid disk DMA hold requests until this task is complete. However, if the DMA does not reactivate a hold request during service to the Ethernet, then the arbiter instructs the mode controller to deactivate ArbHoldIOP. This signal allows the IOP to resume processing.

4.2.4 Service Request: PCE

For the PCE to request service the following conditions must be present.

- The IOP issues a PC condition command to enable the PCE option to start processing.
- While the PCE is processing, the rigid disk DMA requests service.
- During the rigid disk DMA servicing, a condition is activated to request service from the IOP.

Note: The mode controller controls the PCE option.

The following sequence describes the system operation after a PCE service request.

1) When the I/O issues a PCE condition command, the mode controller activates a hold to the IOP and waits for the IOP to issue a hold acknowledge to the arbiter.

2) When the IOP receives the hold request, the mode controller deactivates its PCE hold, thus allowing the PCE 80186 to begin processing. The mode controller directs any Ethernet or RDC bus requests toward the PCE until the PCE 80186 is instructed to stop processing.

3) While the PCE 80186 is processing, the rigid disk DMA issues a hold request. The arbiter passes this request to the mode controller which in turn activates a PCE hold. The arbiter' then waits for the PCE 80186 to activate a hold acknowledge. When the hold acknowledge is active, the arbiter also activates a RDC hold acknowledge.

4) During the rigid disk DMA transfers, the arbiter and mode controllers see that one or a combination of the condition signals are active *(how)*. After the next DMA transfer, the arbiter deactivates hold acknowledge and waits for the DMA to deactivate hold request.

5) After the DMA deactivates hold request, the arbiter instructs the mode controller to deactivate its hold request to the IOP. However, because the mode controller senses the same condition signals as the arbiter senses, it keeps the PCE hold activated and deactivates the hold to the IOP. This process restarts IOP processing, so other requests can be serviced.

6) After the IOP services the requesting device, the PCE processing and rigid disk DMA servicing do not resume until the IOP again executes an allow PCE command or allow RDC command.

7) When the IOP completes the required tasks, it instructs the mode controller via a hold command to activate a hold command via the arbiter. The IOP is then placed in a permanent hold state. This hold condition remains until the mode controller senses (or acknowledges?) that one or a combination of condition signals are activated. When one of these control signals is activated, the mode controller immediately deactivates the hold from the arbiter to allow the IOP 80186 to resume processing.

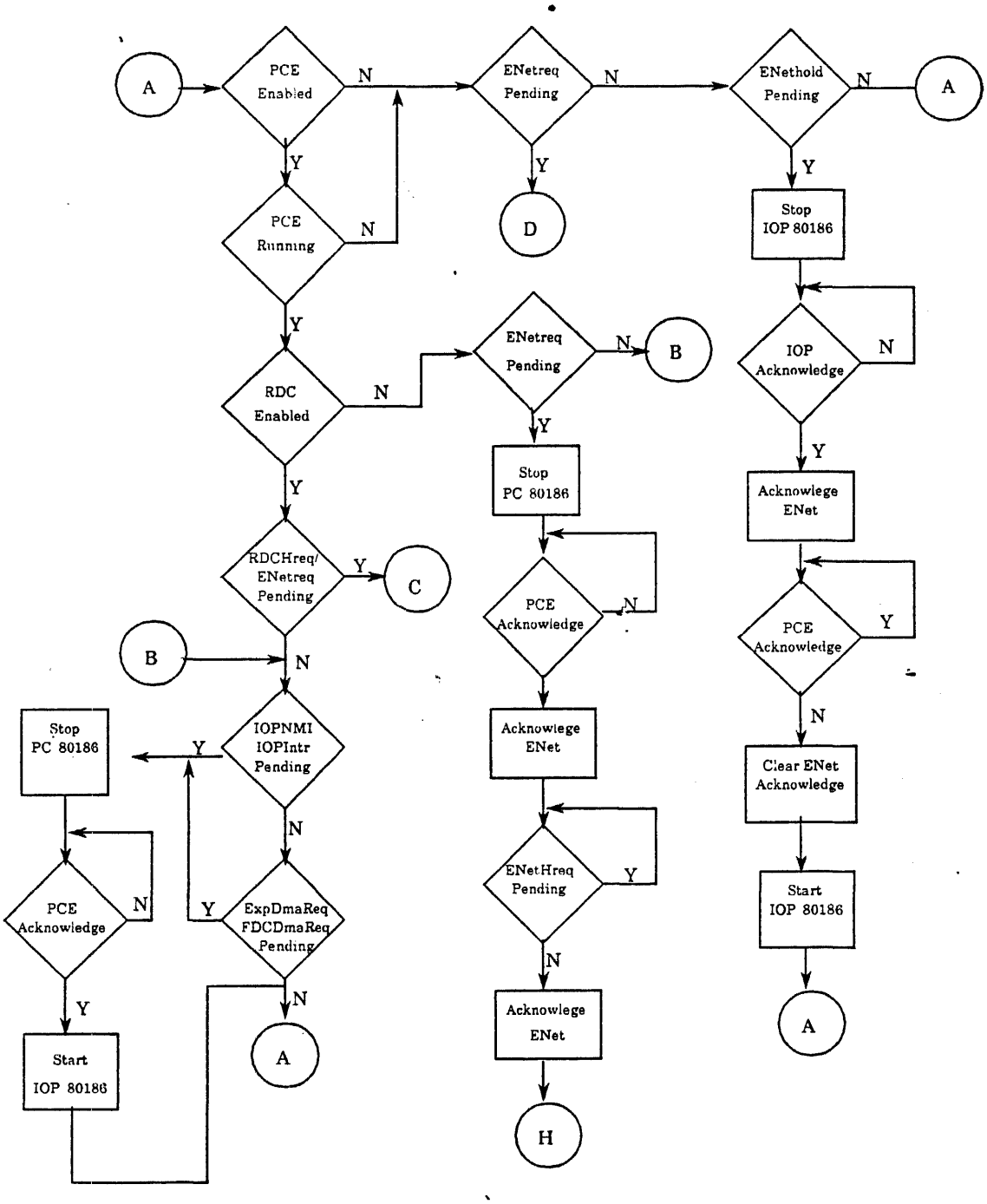


Figure 4.2. Service request: Ethernet

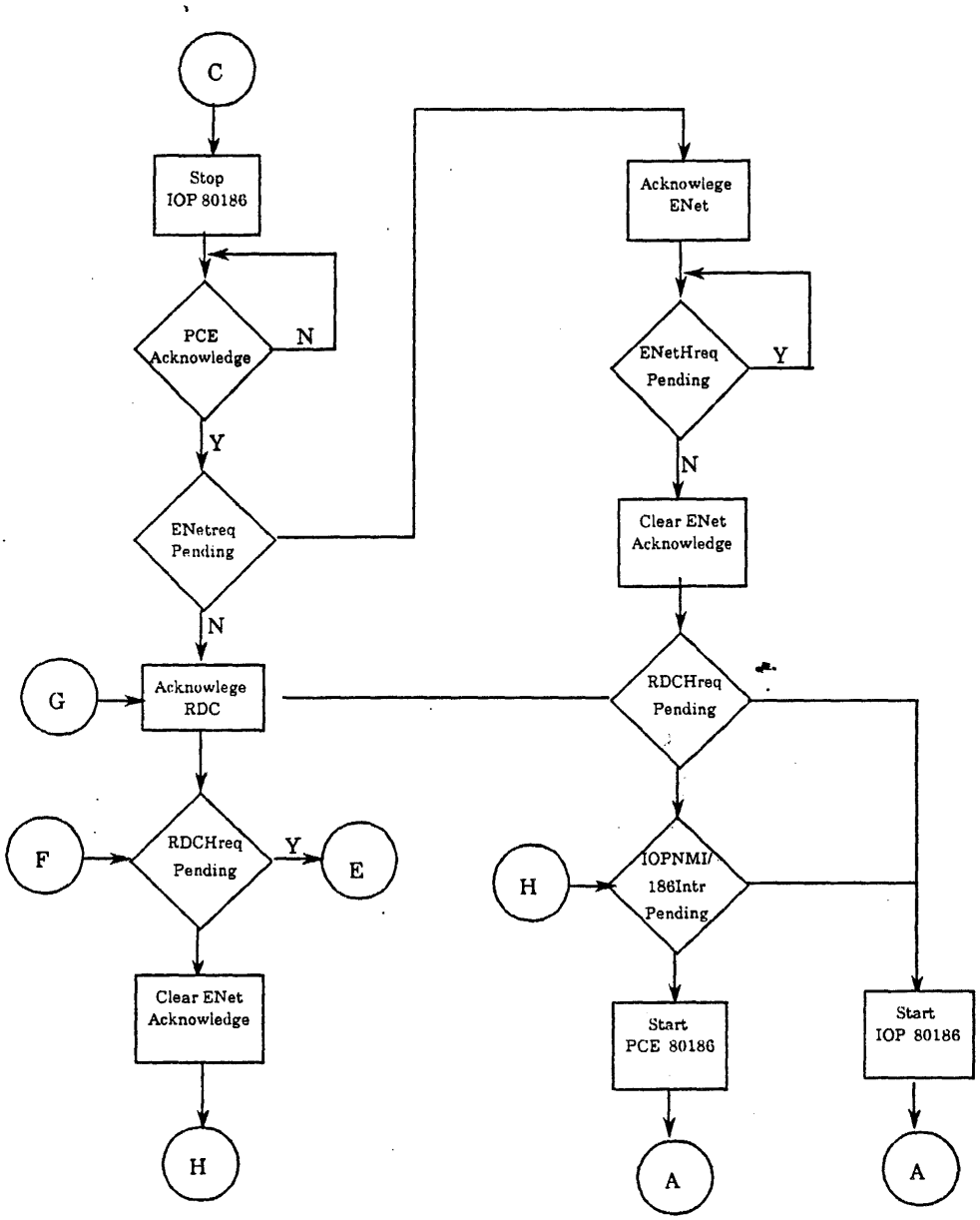


Figure 4.3. Service request: rigid disk DMA

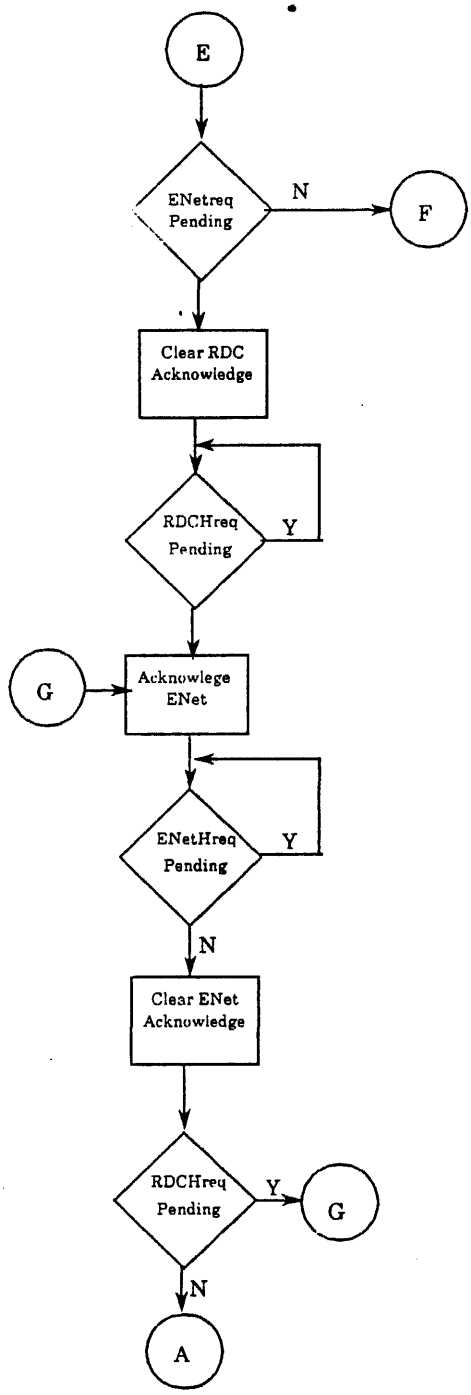


Figure 4.4. Service request: rigid disk DMA controller

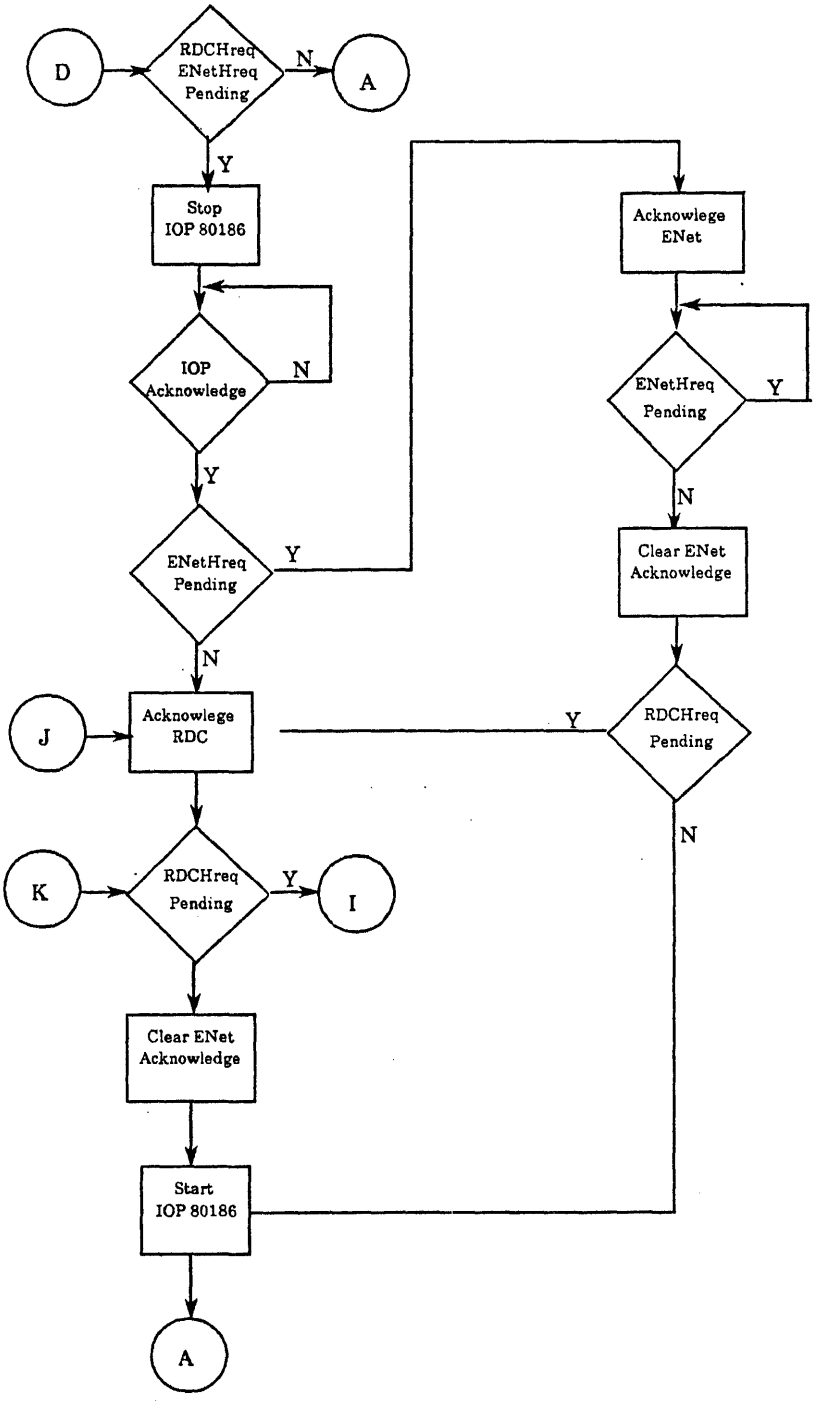


Figure 4.5. Service request: PCE

5 Rigid Disk Subsystem

The IOP rigid disk subsystem provides support for the Dove workstation rigid disk operation. The subsystem has four main components:

- the disk drive, which provides local permanent file (refer to Section xx)
- virtual memory swapping for Pilot operation system (refer to Section xx), and system booting (refer to Section xx)
- the rigid disk controller (RDC), which supports labels and the various disk operations required by the Pilot operating system.
- the DMA controller and the rigid disk FIFO, which transfer control, data, and status blocks between main memory and the rigid disk controller.

Figure 5.1 illustrates the rigid disk subsystem.

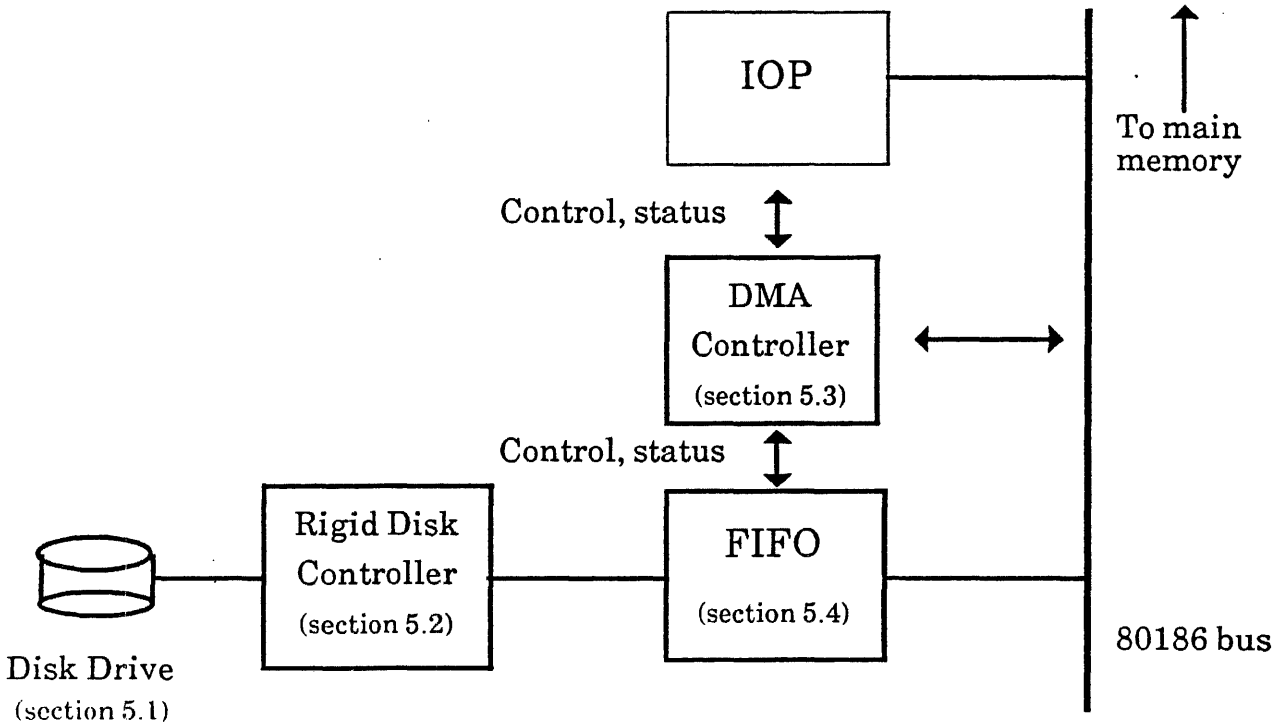


Figure 5.1. Rigid disk subsystem

5.1 Rigid Disk Drive

5.1.1 Hardware

A single rigid disk drive is supported, either half or full-height. Performance requirements of the drive meet or exceed the Shugart SA1004 performance standards. The drive is a $5\frac{1}{4}$ inch Winchester drive with 3 bits of head select (eight heads max), four drive select lines (four drives max), and 5 MBits/sec transfer rate with data encoding and decoding done by the controller. The drives use dc spindle and head motors. If the power supply can handle the drives, then no drive problems with varying line voltages and frequencies occur.

Spindle speed is required not to vary by more than one percent from nominal over the entire range of specified operating conditions.

One departure from the specifications is allowed: 4 bits of head select. This change gives 16 heads maximum as long as the drive does not require an external reduce write current signal. The bit normally used for reduce write current is reassigned to head select 3.

5.1.2 Theory of Operations/Programmer Interface

The format of a disk is the pattern written on the disk to define the sector numbers; that is, the format labels the sector, and provides a defined space for the data.

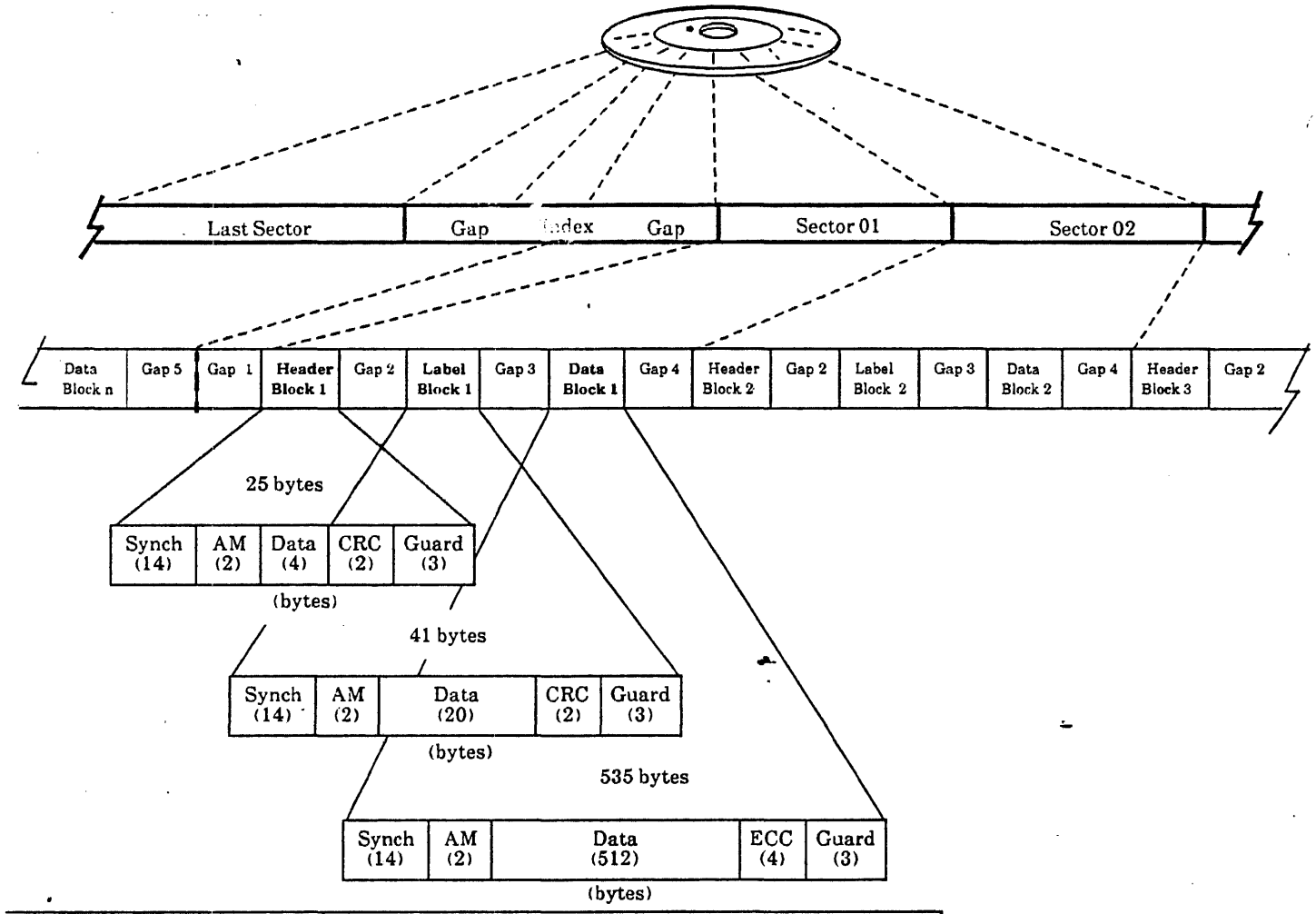
Figure 5.2 illustrates the format for the rigid disk drives.

The disks of a drive are divided into cylinders. A cylinder is the surface of the disks swept by all the heads as the disk revolves without moving the heads. A cylinder is further divided into tracks. A track is the area on one surface of a disk swept by one head as the disk revolves without moving the heads. A sector is a subdivision of a track. In a soft-sectored drive, each sector begins with a header block. The header marks the beginning of a sector and uniquely numbers the sector. Each sector has a unique combination of cylinder number, track number within the cylinder (head number), and sector number within that track.

Each sector is divided into three blocks: a header block, a label block, and a data block. Each block has the same parts: a synch field, an address mark byte, a block ID byte, a data field, a CRC or ECC field, and a Guard field. The address mark byte and the block ID byte are frequently called the address mark.

Each block may be written at a different time, but once a block is written, the remaining blocks in that sector must be written at the same time. The header block can only be written when all headers of a track are written. In this controller there is no "Write Header" command. Headers may only be written using the "Format" command, and the minimum length for a format command is one track.

A track extends from the leading edge of the index pulse to the leading edge of the next index pulse. The index gap of 50 bytes follows the index pulse. The gap is filled, as are all gaps,with zeros. This gap allows for the format operation to overrun the index pulse by a few bytes. It also provides room for the read amplifiers to recover after switching heads.



Notes:

Gap 4 includes allowance of 4 bytes for WriteLong
In WriteLong the data field in the data block = 516; ECC = 4; Guard = 3; Total = 539 bytes

Nominal Track length = 10416 bytes
1 minimum track = 10311 bytes (nominal - 1 %)

1 track = Gap 1 + [Header + Gap2 + Label = Gap3 + DataField] * 16 + Gap4 * 15 + Gap5
= 50 + [613] * 16 + 16 * 15 + Gap5
= 50 + 9808 + 240 + Gap5 = 10098 + Gap5
Since a nominal revolution = 10416 bytes, Gap5 (Nominal) = 318
Since a minimum revolution = 10311 bytes, Gap5 (Minimum) = 213
(Gap5 must be at least Gap4 + 1% of the track length excluding Gap5 + 117 bytes)

Gap 1 = 50 bytes
Gap 2 = 6 bytes
Gap 3 = 6 bytes
Gap 4 = 16 bytes
Gap 5 = 213 to 318 bytes

Figure 5.2. Rigid disk format

Notes:
- The sync field provides room for the phase locked loop to lock.
- The guard field eliminates problems caused by ending write current too close to the CRC field.
- The gap between blocks provides time to set up the next operation and allows for variations in disk speed.
- The gap between one data block and the next header is long enough to allow write current to be turned off without affecting the next header. This spacing takes into account all possible variations in disk speed.
- The gap at the end of the track provides space enough that a track formatted at the slowest allowable disk speed will fit completely between index pulses.

5.2 Rigid Disk Controller

The rigid disk controller directly controls the drive and read/write logic, and consists of a 1K x 24 PROM control store, a fast 8-bit microcontroller (8x305) with a 256 byte scratchpad memory, and read/write logic.

The rigid disk controller can function with any 5¼ inch drive that supports the ST412/ ST506 interface. The controller reads, writes, or verifies any number of contiguous sectors and will switch heads if necessary. The commands supported are: Restore (Recalibrate), Format, Write Data, Write Label and Data, Read Data, Read Label and Data, Read Label and Skip Data, and Verify Data. The diagnostic commands that are supported are: TBD.

5.2.1 Hardware

The rigid disk controller components are off-the-shelf components. Non-industry-standard format and operations are supported by the controller microcode.

Figure 5.3 illustrates the 50-pin 8x305 microcontroller chip. Table 5.1 lists the pins and signals.

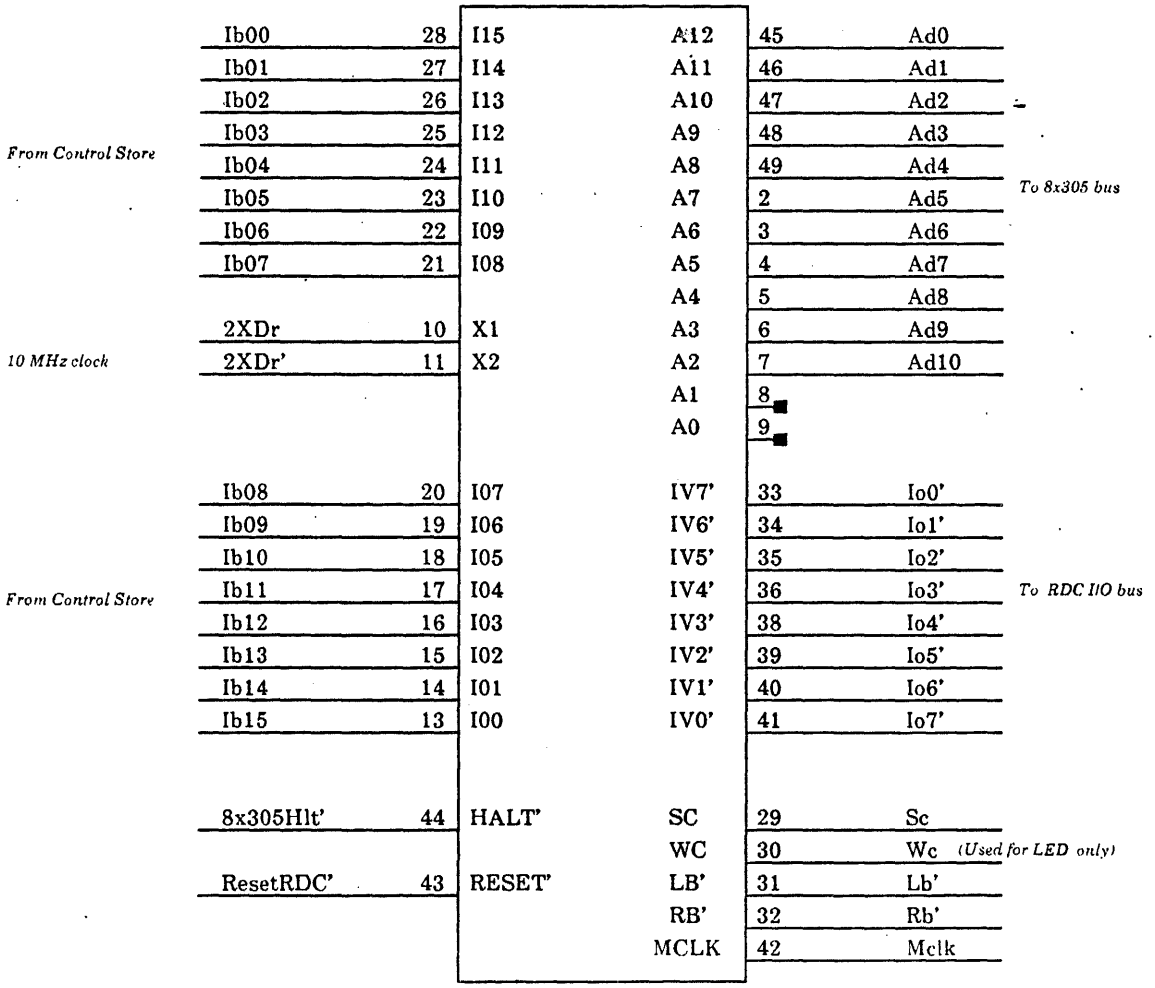


Figure 5.3. 8x305 microcontroller

Table 5.1. 8x305 Pin Description

Symbol	Pin #	Type	Function
A0-A12	2-9 45-49	Output	Program address lines to 8x305 bus. A0 and A1 (pins 8 and 9) are not used.
HALT'	44	Input	Not used in the RDC.
I0-I15	13-28	Input	Instruction lines for 16-bit instructions from program control store.
IV0'-IV7'	33-36 38-41	I/O	Bidirectional 3-state lines for data and/or addresses to RDC I/O bus. A low voltage level is binary 1.
LB'	31	Output	When low, allows access to devices connected to the left bank.
MCLK	42	Output	(Master Clock) clocks I/O devices and/or synchronizes external logic.
RB'	32	Output	When low, allows access to devices connected to the right bank.
RESET'	43	Input	When low, initializes the 8x305; that is, sets program counter/address register to zero and inhibits MCLK. WHile RESET' is low, LB' and RB'are forced high asynchronously.
SC	29	Output	(Select Command) Indicates that data is being written.
WC	30	Output	(Write Command) Writes to RDC diagnostic display.
X1, X2	10, 11	Input	10 MHz clock source with complementary output.

Pin # 1 = VCR, input from series-pass transistor.
Pin #12 = Ground
Pin # 37 = Vcc, +5V power supply
Pin #50 = VR, output reference voltage for external series-pass regulator transistor.

5.2.2 Theory of Operations

Figure 5.4 illustrates the rigid disk controller.

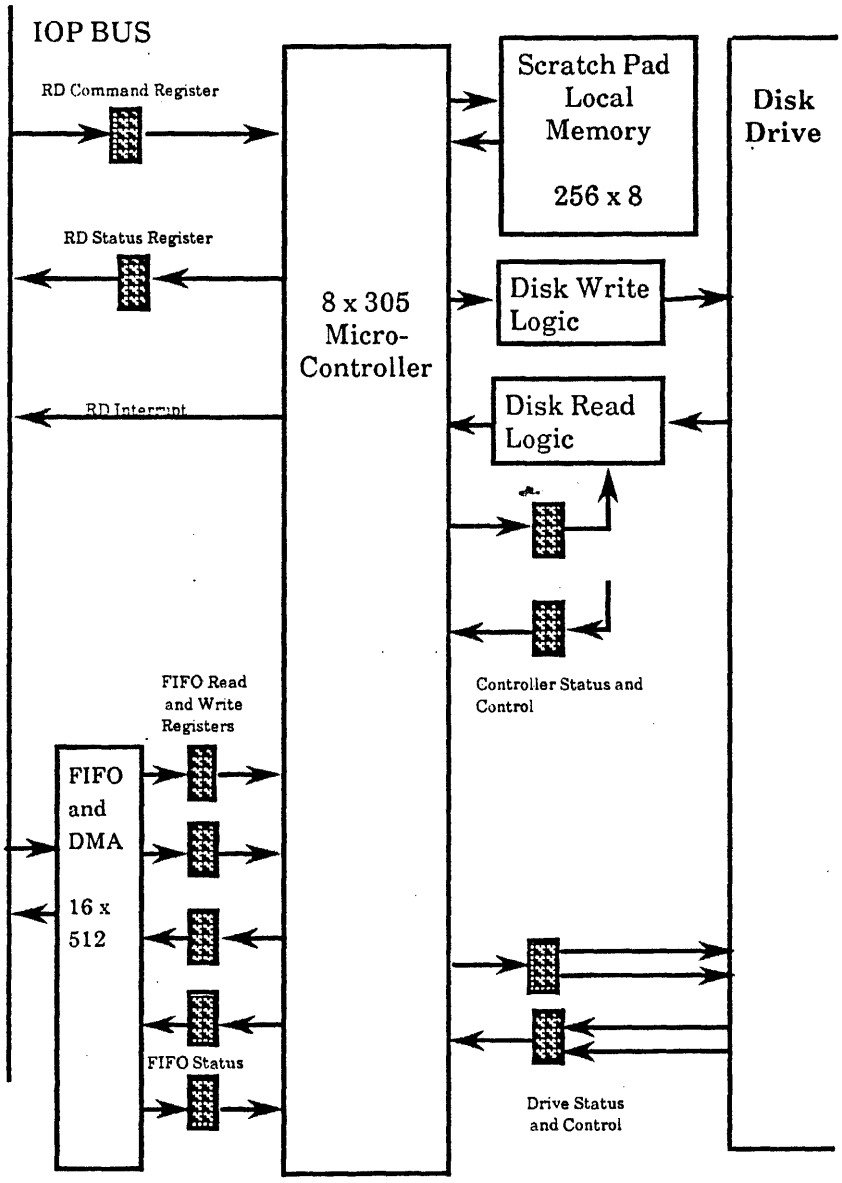


Figure 5.4. Rigid disk controller block diagram

The 8x305 microcontroller controls the rigid disk controller. The communication with the IOP and the disk control hardware is also shown.

Blocks in the diagram are discussed below.

5.2.2.1 Command and Status Registers

The 8-bit command and status registers permit the IOP and the 8x305 microcontroller to synchronize communication. The command register is loaded by the IOP and read by the 8x305. The status register is loaded by the 8x305 and read by the the IOP. The 8x305 interrupts the IOP, but is itself not interruptable. The disk command block and data are passed to and from the controller via the FIFO and the DMA.

5.2.2.2 Scratch Pad and Local Memory

The scratch pad stores the image of the header and label and other information needed by the controller.

The scratch pad memory is a high speed 256 x 8 RAM. The scratch pad address register consists of two 4-bit parallel-loadable counters. The WriteMA instruction parallel-loads this address counter. Subsequent reads and writes to and from the scratch pad use this address. Any read command with the appropriate bit set increments the address. This increment MA feature is used with a read into 8x305 command.

Note: The address counter connection appears to decrement the address instead of incrementing it. The counters are defined for positive-true logic while the 8x305 I/O Bus is negative-true. With negative-true logic, the positive-true decrement becomes a negative-true increment.

5.2.2.3 Drive Status and Drive Control Registers

The drive status and drive control registers are used by the 8x305 microcontroller to monitor and index the current state of the drive. Control signals are: Head Select, Drive Select, Step Direction In, Step, and Write Enable. Status signals are: Track00, Ready, Seek Complete, Write Fault, and Index.

5.2.2.4 Microcontroller Data Paths

Figure 5.5 illustrates data paths of the 8x305 microcontroller.

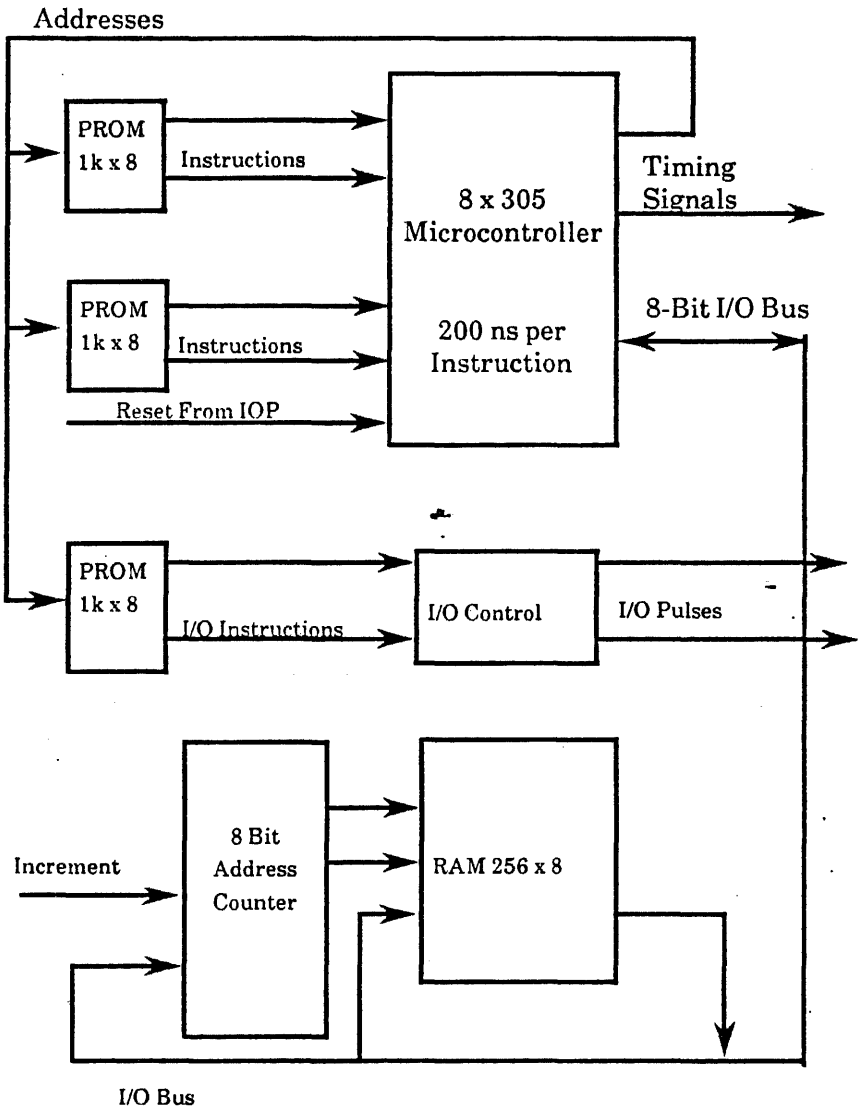


Figure 5.5. Rigid disk controller microcontroller detail

The 8x305 sends address signals to three PROMs. Two of the PROMs send instructions back to the 8x305. The third PROM stores input/output portions of the commands. The I/O portion of each instruction and 8x305 portion are sent to I/O control and to the 8x305 at the same time.

The 8x305 communicates with the rest of the system via an 8-bit, bi-directional I/O bus. The connections to the 8x305 illustrated in Figure 5.4 are connections to this I/O bus.

The 8x305 sends out timing signals to synchronize the system with its own internal timing. The 8x305's clock is the 10MHz 2x writeclock used by the disk write circuits. The 8x305 executes each instruction in 200 ns; each bit to or from the disk also takes 200 ns. The IOP 80186 controls the reset line to the 8x305.

5.2.2.5 Write Logic Data Path

Figure 5.6 illustrates the write logic data path.

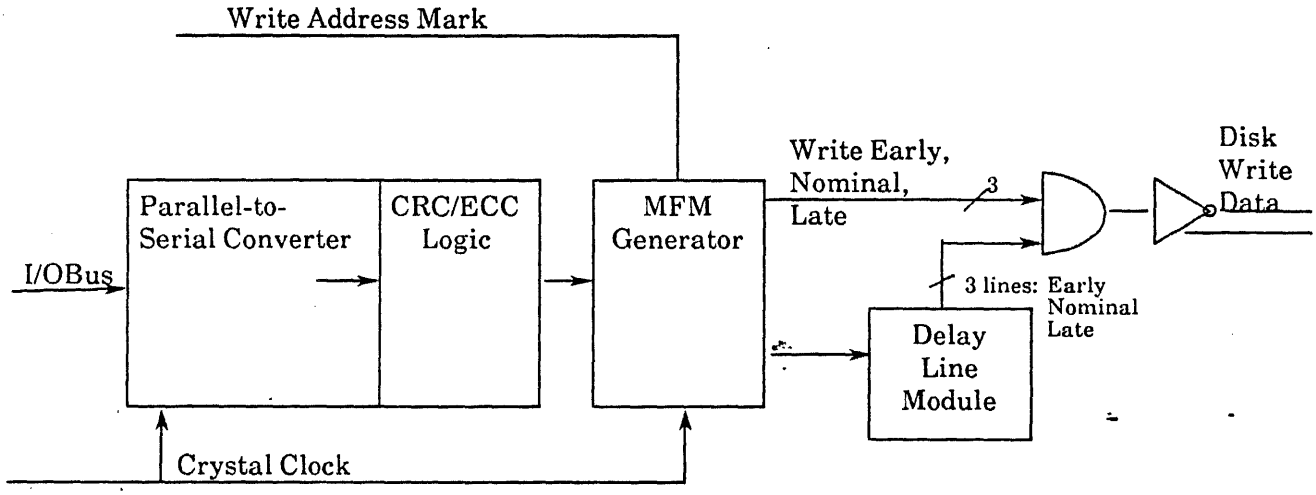


Figure 5.6. Rigid disk controller write logic data path

The 8-bit parallel data from the I/O bus is loaded into the parallel-to-serial converter. The serial output from this chip, in NRZ form, is sent to the CRC/ECC chip, where the CRC or ECC is accumulated. From the CRC/ECC chip, the serial data goes to the MFM generator. The data is then converted to MFM format.

The MFM generator also provides the early,nominal, and late signals necessary for precompensation. If precomp is not enabled, then only nominalis used. The MFM signal goes through the delay line which produces three outputs, each 12 ns apart. These early, nominal, or late MFM signals from the MFM generator are gated with the early, nominal, or late signals from the delay line to produce the correct output. The write data signal is converted to a differential signal for noise immunity, and is sent to the drive.

When the data has completely passed through the CRC/ECC generator, a control signal from the 8x305 changes the internal configuration of the CRC/ECC chip. This ensures that the check bytes are not accumulated, but are instead sent out following the data. The MFM generator also has a provision to delete a clock, producing the missing clock needed to generate the address mark.

Note: Two control signals have names that include "write": write and write gate. "Write" sets chips that have both a read and a write function to the write function. Write gate enables write current in the drive head, and is turned on when all components of write logic are ready to write.

5.2.2.6 Read Logic Data Path

Figure 5.7 illustrates the read logic data path. Refer to section 5.2.3.3 for timing of this data path.

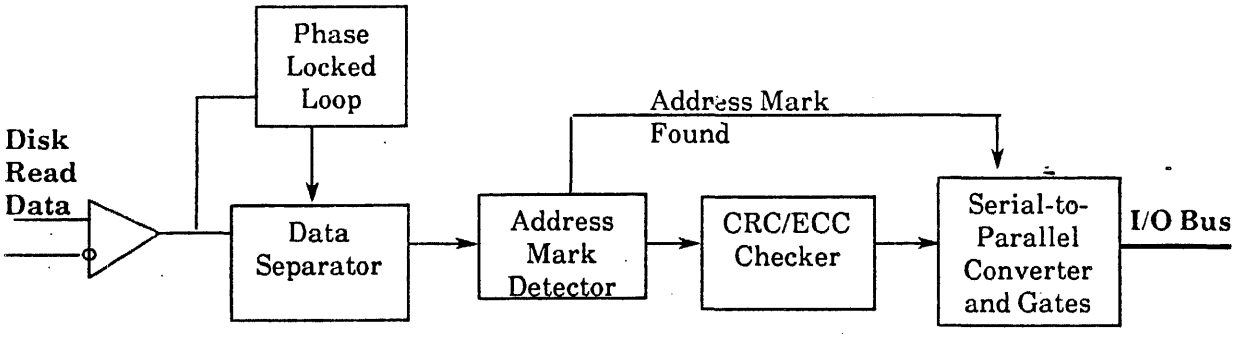


Figure 5.7. Rigid disk controller read logic

The disk is normally in a read condition with read gate off. A chip with both read and write functions is normally in read mode. When write is enabled, the chip switches to write mode; read is enabled with a read gate.

Differential data from the disk enters as disk read data, is converted to single-ended data, and goes both to the phase locked loop and to the data separator. When write gate is off, the PLL locks on the 2x write clock, and thus is near the frequency required to read data. When write gate is turned on by the 8x305, the PLL switches to disk read data. When the PLL locks, it generates the signal LockDet.

The data separator separates the data pulses from the clock pulses and sends the separated pulses to the address mark generator. The data separator produces the signal RdDataFnd when the first "1" bit is detected.

The address mark detector looks for a certain pattern in the data and clock output of the data separator. This unique pattern is the address mark. The address mark found signal (AmDet) enables the serial-to-parallel converter.

Data passes through the CRC/ECC checker, which is the same chip that generates the CRC/ECC check bytes for write operation. The 8x305 sends a signal to the checker after

the entire data stream, including the check bytes, has passed through the checker. When the checker is finished the internal shift register should contain all zeros. Anything else is an error. The contents of the internal shift register are shifted out, following the data, to the serial-to-parallel converter and then to the 8x305.

The serial-to-parallel converter accumulates the data, including the status of the CRC/ECC register, into 8-bit words, and makes these words available to the 8x305. The converter has an internal counter that controls the transfer of data between the internal shift register and the internal buffer register. This counter also signals the 8x305 that another byte is available. The 8x305 now has seven instruction times to read the data before the next byte is loaded on top of buffer.

5.2.2.7 Reading and Writing the DMA and FIFO

The 8x305 reads from and writes to the FIFO, but has no control over FIFO direction. The IOP initializes the FIFO to give the FIFO the proper direction and to clear the FIFO to empty, when appropriate.

Disk operations are set up by high level programs in the Disk Command Block . Table 5.2 lists disk commands and rigid disk controller operations.

Table 5.2. Disk Commands and RDC Operations

Disk Commands		RDC Operations
NOP-IDLE		Restore-Recalibrate
GetCommandBlock		Format Tracks
ExecuteCommandBlock		Read Data
LoadCommandBlock		Write Data
ExecuteDiagnostic		Write Label and Data
		Read Label and Skip Data
		Read Label and Data
		Verify Data

The read and write sequence is as follows:

1) TheIOP send the Disk Command Block via the FIFO to the 8x305 microcontroller.

2) The IOP puts a command into the command register.

3) When the 8x305 recognizes the new command, it sets the status register to indicate that it has received the command.

4) The 8x305 executes the command.

5) When execution is completed, the 8x305 sets the status register. At this time, the 8x305 also sets the RDC Interrupt. The interrupt is reset when the next command is received.
Note: The command NOP-IDLE does not set the RDC interrupt.

6) If the received command is ExecuteCommandBlock or ExecuteDiagnostic, then the 8x305 signals the rest of the rigid disk controller to execute the next operation contained in the control block.

7) An operation is complete when the number of sectors to transfer or tracks to format is complete or when an error occurs. After an operation is complete, the 8x305 notifies the IOP through the status register.

8) Via the command register, the IOP instructs the 8x305 to put the ending block in the FIFO. The ending block now contains the current status, including errors (by type), of the operation.

9) The 8x305 signals the IOP that the ending block is in the FIFO; the IOP in turn signals the 8x305 to wait for the next command.

5.2.3 Programmer Interface

The following subsections describe DMA transfers, the rigid disk controller registers, normal operation sequence and error recovery sequence, and timing.

5.2.3.1 DMA Transfers

Mode control on the AM2942 word counter must first be initialized by writing 0101 to address 6H of the CR2-CR0 register. (?) (Same as setting mode control to 3?)

When a DMA transfer occurs, the AM2942 word count register is set to indicate a word count between 1 and 256. The data is expressed in bits 8-1. Because the address is moved over by one bit, it becomes two's complement in number of words. Refer also to Table 5.11 in subsection 5.3.3.2.

Next, the AM2942 address register A8-A1 is set up. The page point registers, the direction bit, and the DMA communcation register are also set up to notify the IOP(?) that the data is going either from memory to disk or from FIFO to memory.

When the preceding steps are complete, AllowDMA and AllowRDC commands are issued. AllowDMA starts the DMA but does not let it run. AllowRDC flags the Arbiter for control of the bus. A wait for a DMA interrupt and a verification of the status of the register occur.

The DMA controller directs data transfers among the rigid disk controller, FIFO, and main memory, completing a data transfer in a single bus cycle.

5.2.3.2 Registers

Two 8-bit hardware registers communicate between the rigid disk controller running on the IOP and the microcode running on the 8X305.

The disk controller command register is hardware port 0214H in the rigid disk controller. The controller treats the command register as a write-only port, and issues commands through this register. (The opposite is the case on the 8X305).

The disk controller status register is also hardware port 0214H. The controller treats the status register as a read-only port, and obtains status information from this register. (The opposite is the case on the 8X305).

Disk Controller Command Register

Figure 5.8 illustrates the disk controller command register. Bits 08 through 15 are not used; bits 00 through 07 are programmable.

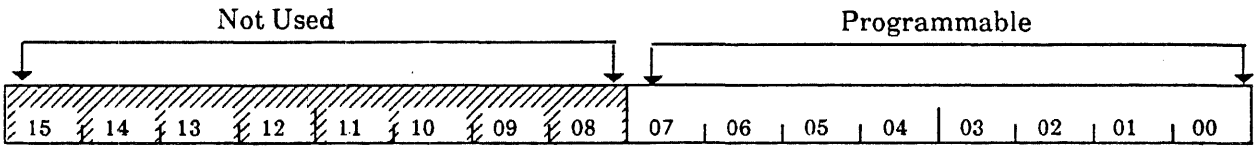


Figure 5.8. Disk controller command register (write-only I/O Addr = 0214 hex)

The bit-level definition of the command register is: **sdxx xxcc,** where

- **s,** when s = 1, is a request to stop at the end of the present sector on the commands read/write/verify. When s = 0, this bit is in normal running mode.
- **d,** when d = 1, indicates that the given command is a diagnostic function.
- **x** bits are unassigned bits, currently set to zero; that is, 0000.
- **cc** defines the command from the IOP's handler to the 8X305 microcode, as follows:

  00 = > Go to and stay in idle mode
  01 = > Fetch the disk command block (DCB)
  10 = > Execute the DCB
  11 = > Store the DCB

Disk Controller Status Register

Figure 5.9 illustrates the RDC status register. Bits 08 through 15 are not used; bits 00 through 07 are programmable.

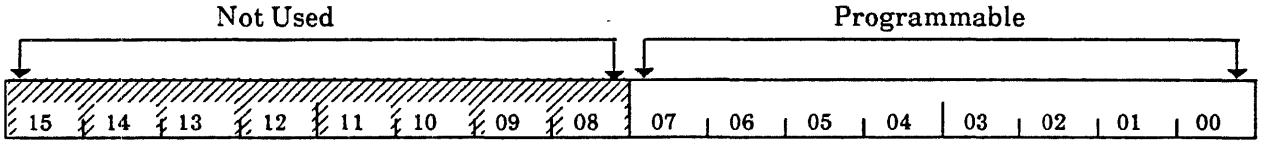


Figure 5.9. RDC status register (read-only I/O Addr = 0214 hex)

The bit-level definition of the status register is: **edss xxrr,** where

- **e,** when e = 1, indicates that an error has occurred. When e = 0, assume that no error condition has occurred.
- **d,** when d = 1, indicates that the operation has been completed. Otherwise, d = 0.

- ss bits indicate that a special operation has occurred. Normally ss = 00.
  ss = 01 indicates a FIFO error has occurred: the FIFO is not empty at the start of the control block loading operation. The Disk Control Block contents will not start at the normal location in the FIFO.
  Note: Errors are always shown in the DCB. However, if DCB cannot be read, the FIFO error is pointed out in the ss bits.
  Note: ss = 10 and ss = 11 are unassigned.

- xx are currently unassigned bits. The controller does not assume that the bits are 00; however, the 8x305 places 0s in these bits.

- rr repeats (echoes) the command bits (cc) in the disk controller command register, and correspond one-to-one to the cc definitions given above.

Register Sequences

Table 5.3 lists the contents of both registers as the controller goes through a normal command sequence. Table 5.4 lists the registers' contents during an error recovery sequence.

Table 5.3. Normal Sequence

Disk Controller Command Register	Disk Controller Status Register	Remarks
0000 0000	0000 0000	In idle state.
0000 0001	0000 0000	IOP issues fetch DCB.
0000 0001	0000 0001	8X305 receives fetch command and begins fetch.
0000 0001	0100 0001 *	8X305 has fetched DCB and has issued an interrupt.
0000 0010	0100 0001	IOP issues execute DCB.
0000 0010	0000 0010	8X305 receives execute command and begins execution.
0000 0010	0100 0010 *	8X305 finishes the command and interrupts the IOP.
0000 0011	0100 0010	IOP issues command "Store DCB".
0000 0011	0000 0011	8X305 receives command and begins storing DCB.
0000 0011	0100 0011 *	8X305 has completed store of DCB and interrupts the IOP.
0000 0000	0100 0011	IOP issues command "GoIdle".
0000 0000	0000 0000	8X305 goes to idle mode (no interrupt is given).

* 8x305 interrupts IOP

Table 5.4. Error Recovery Sequence

Disk Controller Command Register	Disk Controller Status Register	Notes
0000 0000	1100 00xx	IOP issues command "GoToIdleMode".
0000 0000	0000 0000	8X305 goes to idle mode (no interrupt is given).
0000 0011	0000 0000	IOP issues command "Store DCB ".
0000 0011	0000 0011	8X305 sees command and begins storing DCB.
0000 0011	0100 0011 *	8X305 has completed store of DCB and interrupts the IOP.
0000 0000	0100 0011	IOP issues command "GoToIdleMode".
0000 0000	0000 0000	8X305 goes to idle mode (no interrupt is given).

* 8x305 interrupts IOP

5.2.3.3 Timing

Figure 5.10 illustrates timing for reading data, data separation, and address mark detection, described in section 5.2.2.6. In the figure, C indicates a clock pulse and D indicates a data pulse. The number below D is the content of the cell; the character x below C in column 21 is a missing clock, which is interpreted as an address mark.

Bits occur at 5 MHz or 200 ns/bit with no variation. A write operation is fixed at 200 ns; a read operation is plus/minus 1% of 200 ns.

Signals in the figure are:

DOUT – Disk data out
DtaInDlyd – Data in delayed
VFOClk – VFO clock (from Phase Locked Loop)
PLLLocked' – Phase Locked Loop locked
Search – Search for address mark
DHld – Separator hold locked in 0
RDi – Read data within data separator
RDta – Read data out of separator
RCi – Read clock within data separator
RCkS – Read clock out of separator
AmDet – Address mark detected

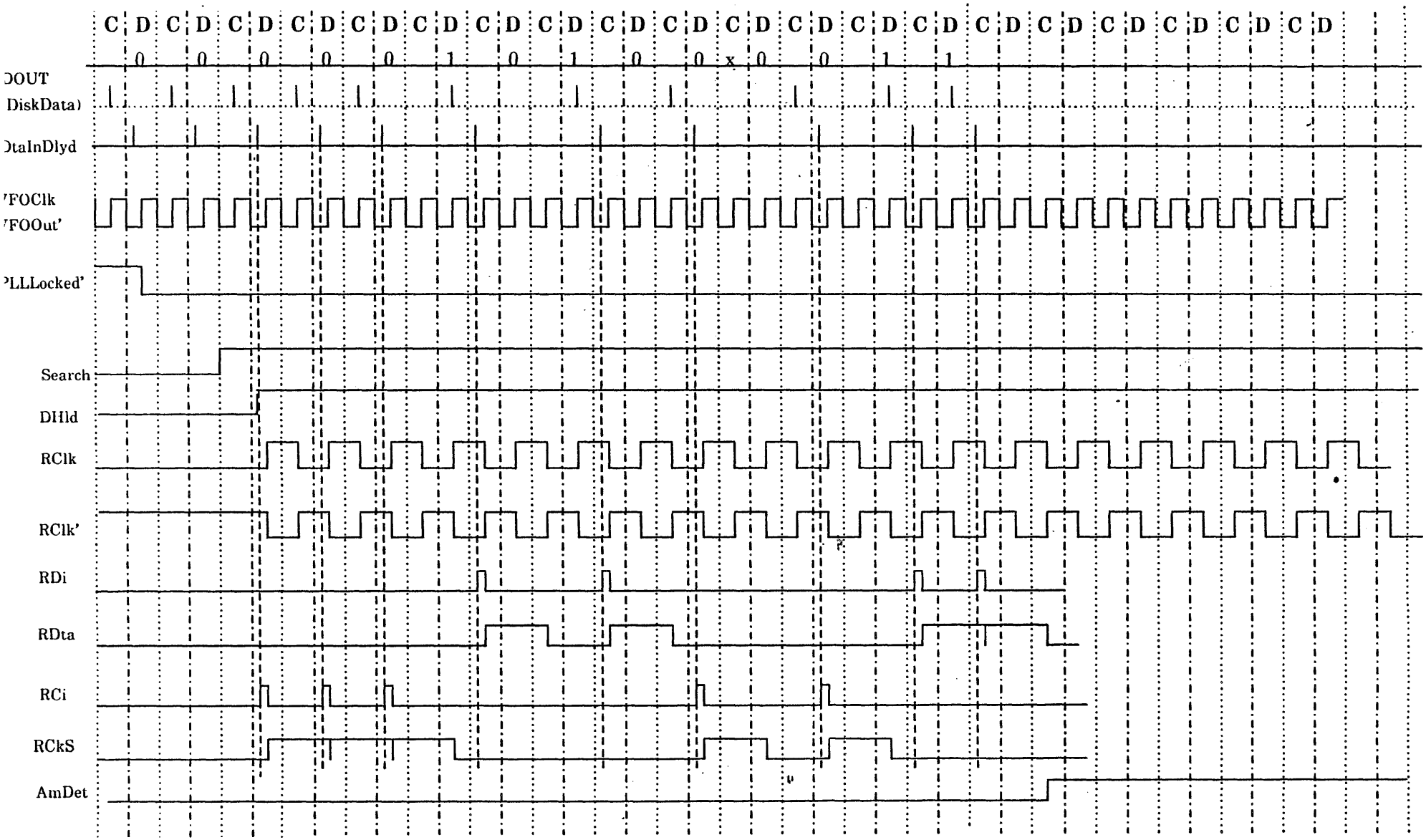


Figure 5.10. Timing for Read data, data separator, and address mark detection

5.3 DMA Controller

The DMA controller controls high speed data transfers between the FIFO and main memory, and consists of three sections: a control section, a data path section and a third section which is the data and control route by which the IOP programs the DMA and FIFO.

The control section includes a control register, a status register, a state machine, and other control logic required to make the system fully functional. The data path section includes the buffers, registers, and latches necessary to direct the data back and forth between the FIFO and the 80186 bus. The program section includes a buffer and other logic.

Figure 5.11 illustrates the DMA controller.

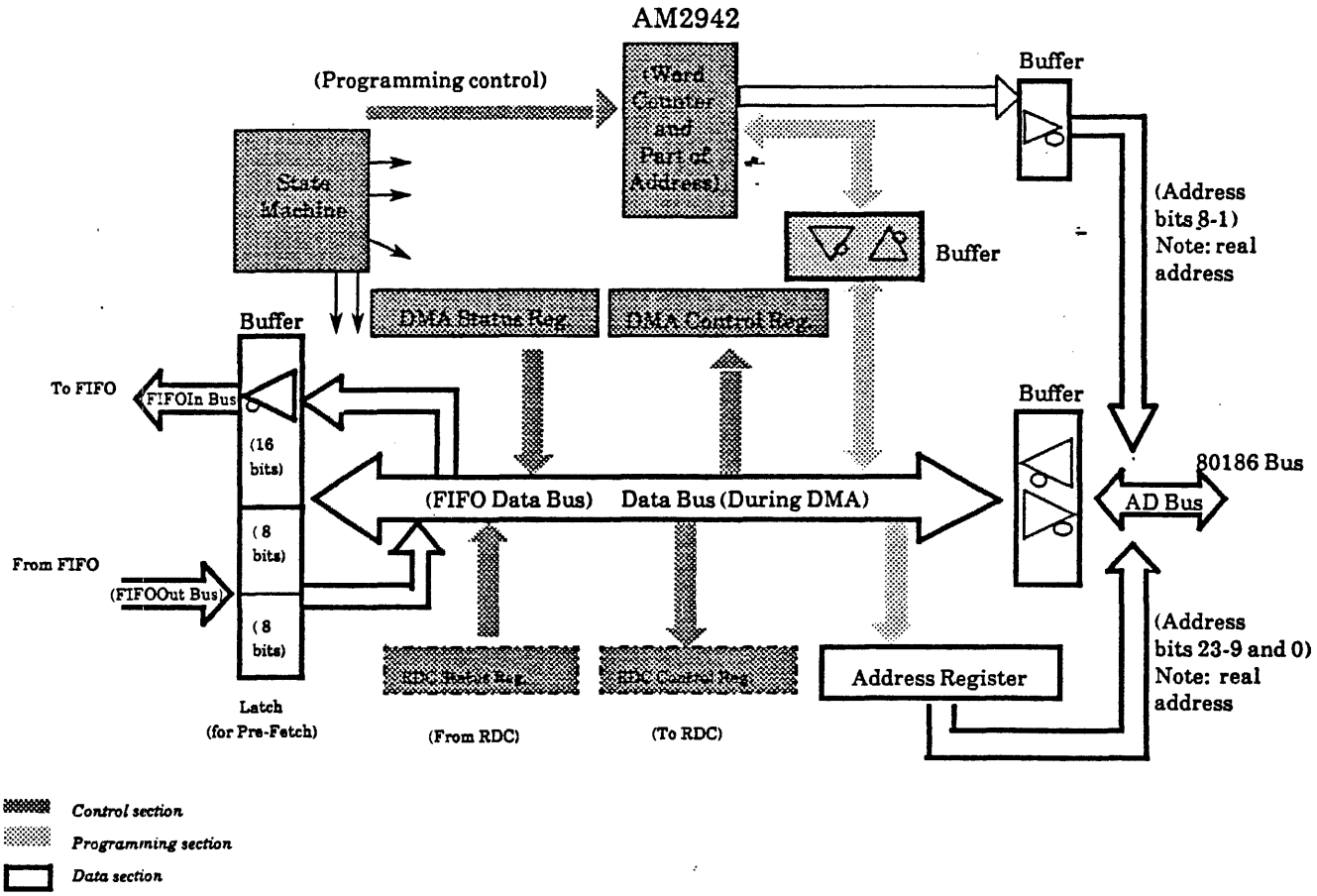


Figure 5.11. DMA block diagram

The main features of the DMA controller are:

- programmable by the IOP (Starting Address, Word Count, and Transfer Direction).

- transfers up to 256 16-bit words at a time.
- provides 24 bits of real address to the main memory controller.
- transfers data at the full 80186 bus rate (one word per four T-state memory cycle); can also tolerate slower memories by inserting wait states in the memory cycle.
- provides an end-of-transfer interrupt to the IOP.
- dynamically uses the available 80186 bus cycles.

5.3.1 Hardware

The controller for the DMA function is a custom implementation and provides the controlling task as programmed. The controller interfaces to main memory through the 80186 bus and is one of the 80186 bus masters.

5.3.1.1 DMA Signals

Tables 5.5 – 5.8 list and explain significant signals of the DMA and bus interfaces.

Table 5.5. DMA Signal Description

Pin Name	Type	Function
StartDMA'	Output	Active low pulse generated when the IOP issues a StartDMA command.
DMAActive'	Output	Negative true signal that indicates whether or not DMA is active; that is, DMA operation in progress. The signal is cleared (DMA not Active) when DMA operation is temporarily suspended by outside factors, such as Ethernet, a qualified interrupt to IOP, or the IOP itself.
Run SM (Run State Machine)	Output	Goes high in response to IOP's StartDMA request, and allows the DMA to run. The signal remains high until the completion of DMA operation and automatically goes low when the DMA operation is complete. During the time that DMA operation is temporarily suspended by outside factors, the signal maintains its active (high) level, indicating that DMA transfer is not yet finished. This signal is available as one bit of the status register.
EndOfXfer'	Output	Negative true signal indicates that the total number of words transferred between main memory and the FIFO is equal to the number of words originally requested for DMA transfer.
ADDr'/Data	Output	Identifies the type of information on the bus during DMA transfer. Low only during the T1 states of the bus cycle while DMA operation is in progress. Remains high when DMA is not active.

Table 5.6. 80186 Bus Control Interface Pin Description

Pin Name	Type	Function
RDCHoldReq	Output	DMA Hold Request (equivalent to HLD in 80186) sent by the RDC DMA to the bus arbiter.
RDCHoldAck	Input	DMA Hold Acknowledge (equivalent to HLDA of 80186) sent by the bus arbiter to the RDC DMA.
S.2'-S.0'	Output	Provides status lines for the 80186 bus, as follows: S2' - S0' — IOP Function 000 — Interrupt Acknowledge 001 — Read I/O 010 — Write I/O 011 — Halt 100 — Instruction Fetch 101 — Read Memory * 110 — Write Memory * 111 — Passive (no action)
186BHE'	Output	Indicates high byte.

* Note: DMA controller can only generate these two combinations.

Figure 5.7. 80186 Data Bus Interface Pin Description

Pin Name	Type	Function
AD.15-AD.00	I/O	Address /Data bus bits 15-00.
A.23-AA.16	Output	Address Bus Bits 23-16.
IOPARDY	Input	Indicates whether the main memory subsystem requires that additional wait states be introduced in the memory cycle.
ALE	Used indirectly	Latches addresses for each memory cycle.

5.3.1.2 DMA as a Peripheral for IOP 80186

The IOP also treats the DMA controller as a peripheral device. Under this condition, dedicated buffers and logic are used to facilitate the DMA programming. Table 5.8 lists the signals used by the IOP when it programs the DMA controller or queries DMA controller status.

Table 5.8. 80186 DMA Program Signals Description

Pin Name	Type	Function
Rd'	Input	Read pulse from 80186 indicating that status information is being requested by 80186.
WrL'	Input	Pulse used for writing the low order byte of a control register.
186DEn'	Input	Pulse requesting the data to be made available. (Refer to 80186 specifications.)
RDMASel	Input	One of the Peripheral Chip Select lines that is dedicated to the RDC/FIFO/DMA generated by the IOP.
RDiskDmaIntr	Output	Causes an interrupt signal from the DMA controller to the 80186, and informs the IOP that it has finished the DMA transfer.

5.3.2 Theory of Operations

Included in the control section of the DMA controller is a PROM-based state machine which provides appropriate control signals to the data path section and the FIFO, communicates with main memory and the bus arbiter, and sends interrupts to the IOP.

The DMA state machine also receives external status information, such as FIFO status or end-of-transfer indication.

Sequencing through consecutive addresses as well as maintaining the word count and parts of the address bits are made possible using an AM2942 chip.

Operation details are described in the following subsections after a brief explanation of the state machine structure.

5.3.2.1 DMA State Machine

Figure 5.12 illustrates the state machine hardware. The DMA state machine is PROM-based; the control information is stored in the form of a bit pattern within the PROM. The figure corresponds to the state machine block of the control section in Figure 5.11.

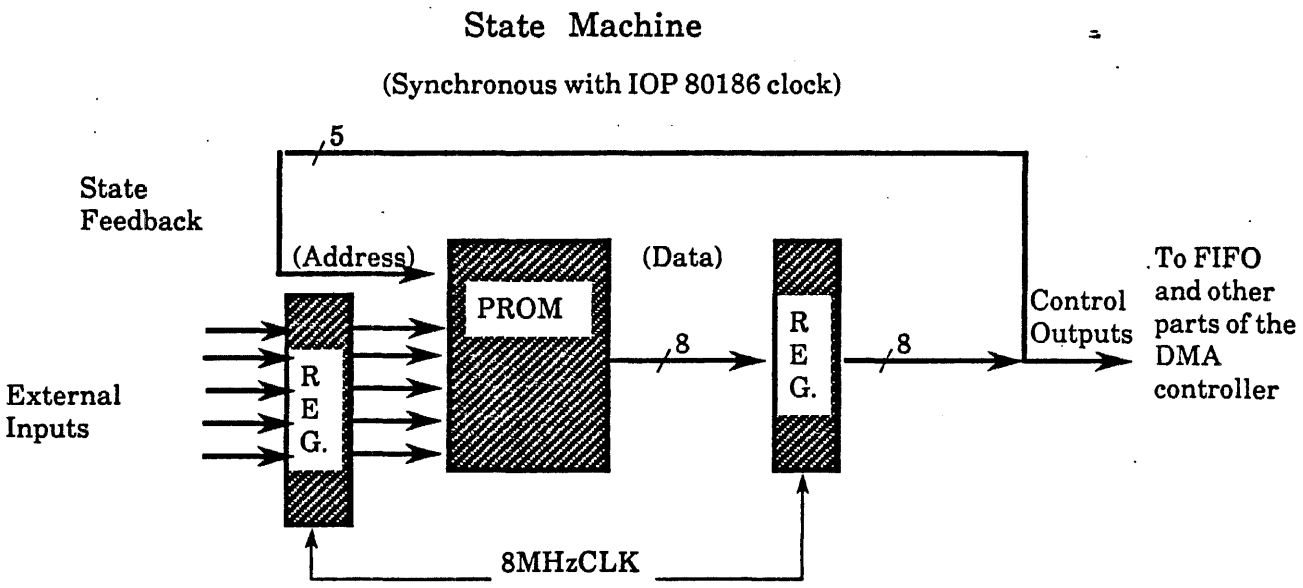


Figure 5.12. Block diagram of DMA state machine

Address inputs to the PROM are signals from different parts of the DMA controller and FIFO that are clocked into a register; they provide a stable address to the PROM for a period of 125 ns. The data vector saved in this location of the PROM is clocked into another register to provide a stable pattern for 125 ns for use as control signals by other parts of the rigid disk subsystem. Some of the data outputs are also fed back as address bits to the same PROM. The state machine marches through the state sequences, as determined by its previous state and by external information.

If an illegal condition is presented to the inputs of the state machine, the machine creates an error condition. The machine does this by: 1) setting the error bit in the DMA status register; 2) freezing the clock for the status register to preserve the latest status just before the error condition; and 3) sending an interrupt to IOP.

5.3.2.2 States of the State Machine

Figures 5.13 – 5.15 illustrate the three phases of the states of the state machine. The circled numbers in the figures refer to the steps as described in the text. Timing for the states is given in section 5.3.2.4.

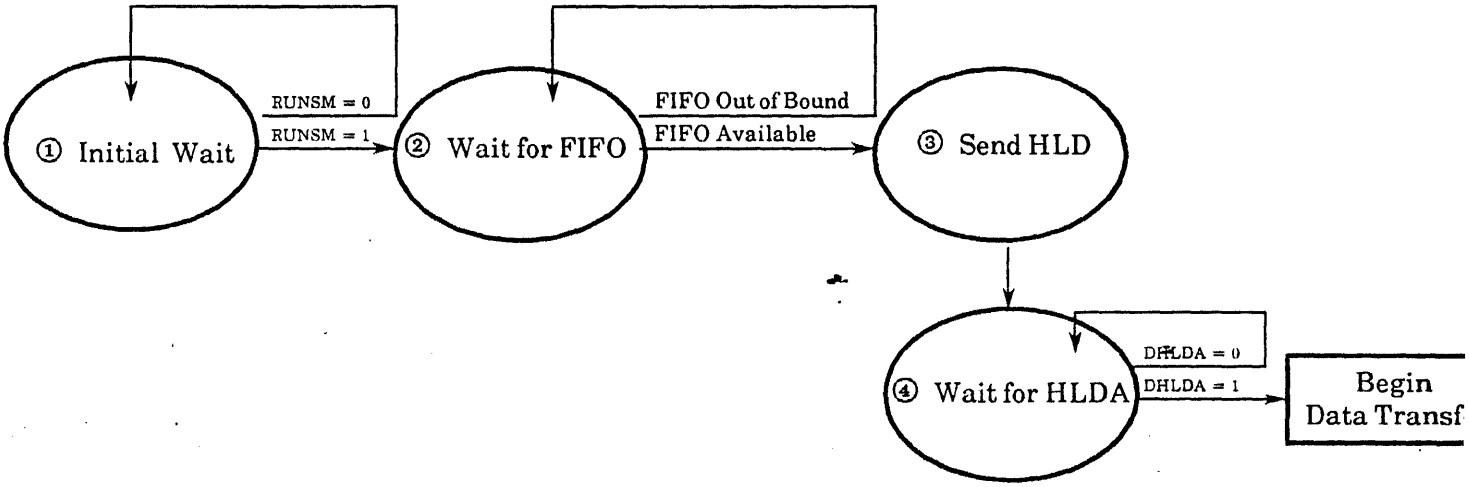


Figure 5.13. DMA states: I - Transfer initiation

1. The state machine normally is in the "initial wait" state. This state occurs at power-up time when the DMA controller is reset, and also occurs after successful completion of a DMA transfer.

2. The StartDMA command from IOP causes the state machine to go to the wait-for-FIFO state, during which FIFO availablity for DMA operation is verified.

   If the FIFO is not available, then the DMA waits in the same state until the FIFO is available. The FIFO may be unavailable for one of two reasons:

   FIFO is full AND the transfer direction is memory-to-disk.
   FIFO is empty AND the transfer direction is disk-to-memory.

   If the FIFO is available, then the DMA goes to the next state.

3. A HoldRequest (HLD) is sent to the bus arbiter.

4. The DMA waits for HoldAcknowledge (HLDA). When the acknowledgement is received, the 80186 bus is clear and available for DMA transfer. The DMA then begins the bus cycles for the transfer.

Figure 5.13 illustrates the bus cycles.

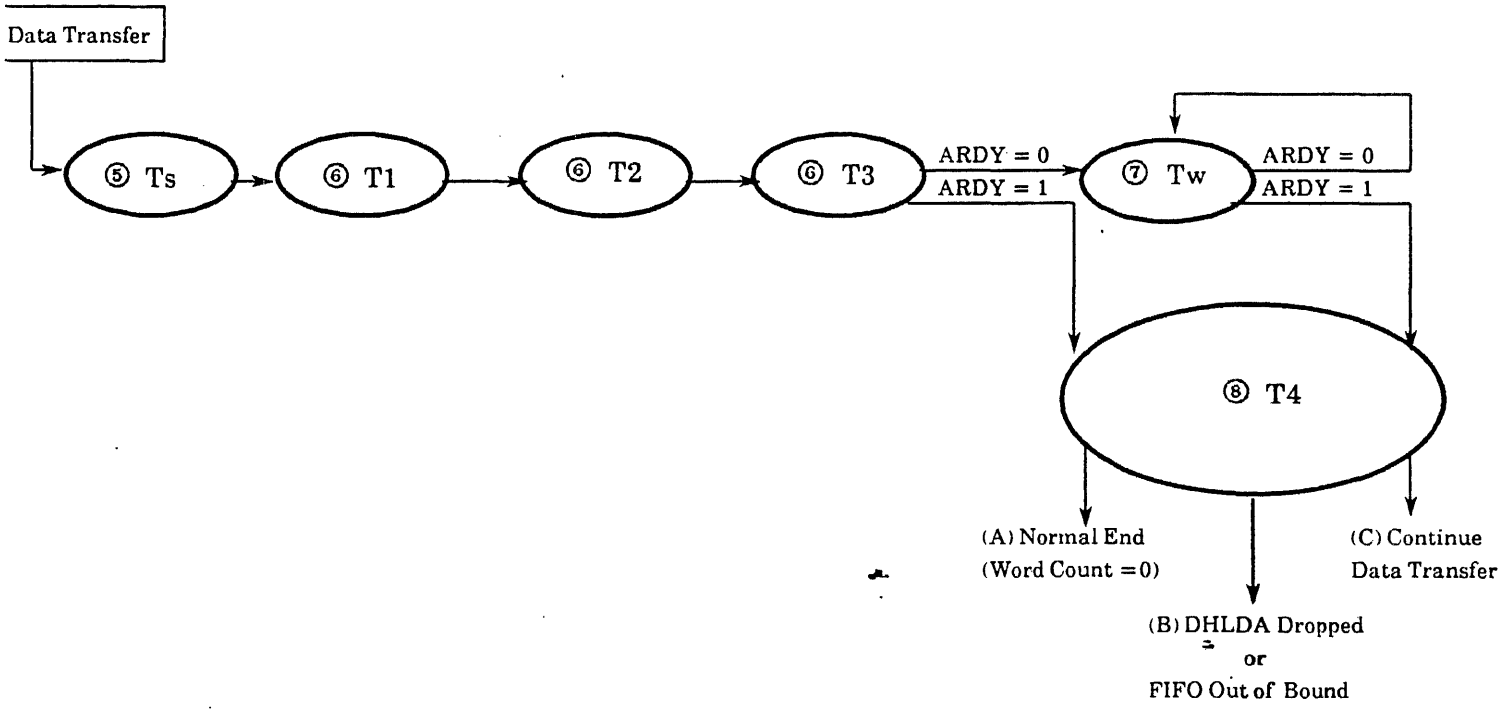


Figure 5.14. DMA states: II. Bus cycles

5. An initial T state, Ts, is first introduced. Ts is equivalent to the last T state of an ongoing memory access.

6. The DMA goes consecutively to T1, T2 and T3 states, while providing signals according to the 80186 bus protocol for memory access and providing appropriate control signals to the FIFO and address/word count logic. If main memory can complete memory access in four T cycles, then the DMA goes to T4 state (step 8).

7. If main memory needs more time to complete the access, then memory informs the DMA controller by pulling ARDY signal low. In response, the state machine, after T3, goes to a Tw state and introduces Tw cycles until the main memory announces completion of access by pulling the ARDY signal back high. The DMA controller then moves from Tw state to T4 state and completes a one word transfer.

8. At T4 state, the state machine branches into three possible operating states: A) normal end; and B) and C) more data to be transferred. Figure 5.14 illustrates these states, which are described below. The error state, labeled D on the figure, is also described. The circled number refers to the step number previously described.

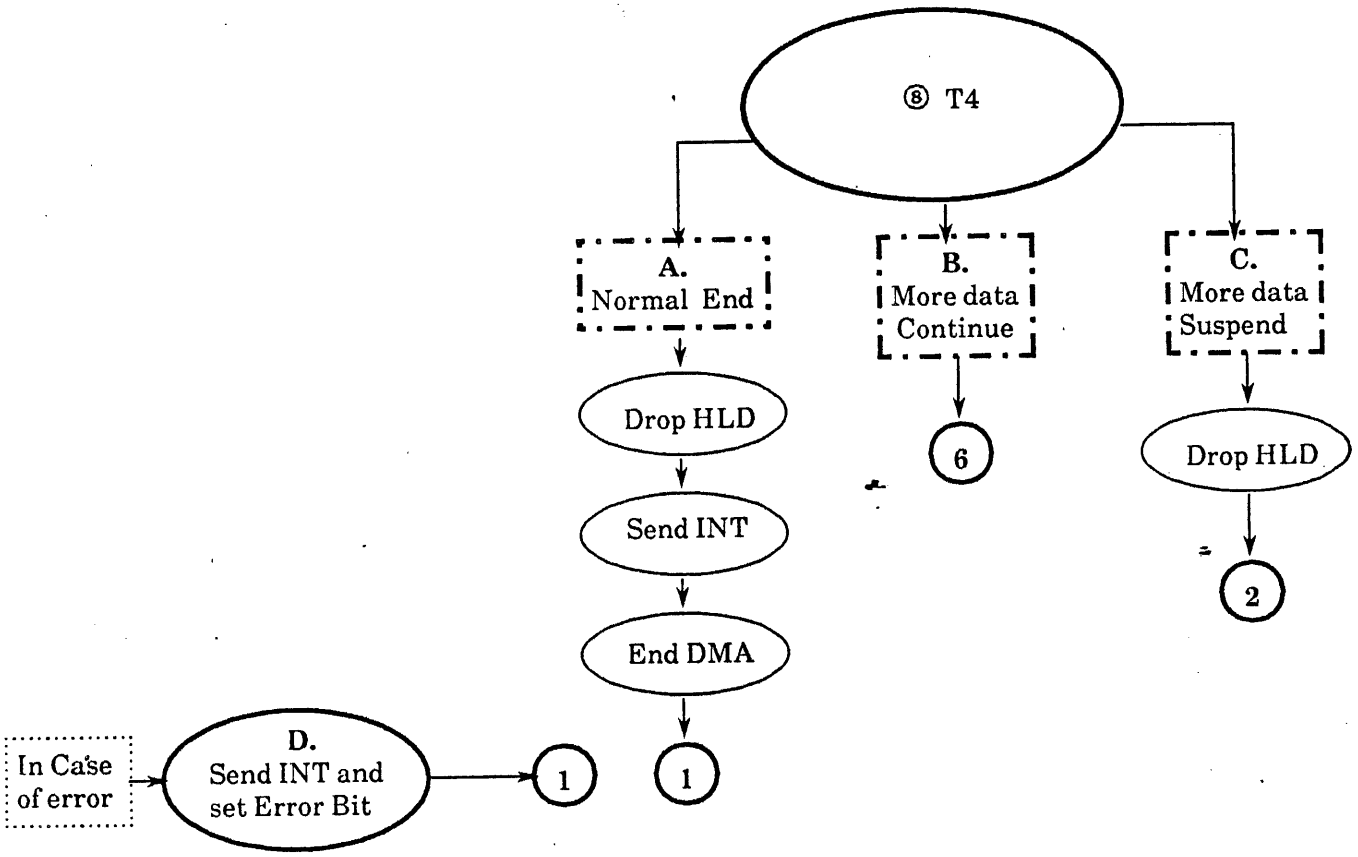


Figure 5.15. DMA states: III. Transfer conclusion

A) If the total number of words requested for DMA transfer has been reached (normal end), then the state machine drops the HLD line, and branches to send an interrupt request to IOP. The branch occurs when an end-of-transfer condition is reached, regardless of other factors, such as unavailable FIFO or other bus master requesting the bus.

B) If more data is to be transferred, and the FIFO is available, and no other bus master is requesting the 80186 bus, then the state machine branches to T1. (Refer to step 6.)

C) If the FIFO is not available for the next word to be transferred or if another 80186 bus master is requesting the bus, then the DMA drops the hold request line (HLD) and relinquishes the 80186 bus. After dropping the HLD line, the DMA goes to the state of waiting for FIFO to become available (refer to step 2).

If the FIFO is available, then, after one more T state (delay through the state described in step 2) DMA moves to SendHLD state to send the hold request again.

D) The DMA state machine is designed around a PROM, but not all of the addressable locations within the PROM are used in implementing the state machine. An error state occurs when the address lines point to an unused portion of the PROM.

When an error state occurs, an error bit is set in the DMA status register, and an interrupt is given to the IOP. The error bit does not disable the state machine. Instead, the state machine goes to the Initial Wait state (step 1) until it receives further directions from the IOP. The status register is frozen until an explicit reset clears it. No further DMA transfers occur until the IOP intervenes, takes recovery measures, and re-issues StartDMA.

For the DMA controller, presence and absence of the HLD and HLDA lines, in addition to the normal meanings under 80186 bus protocol, is a means of communication with the bus arbiter regarding the bus availability. The arbiter communicates the need for the bus evacuation to the DMA controller by dropping HLDA for the DMA controller while DMA is in operation. The controller then leaves the 80186 bus and informs the arbiter of bus availability by dropping the HLD line.

5.3.2.3 Operating Sequence

In a rigid disk DMA operation, one of two possible sequences of operation is exercised. Table 5.9 lists the operating sequence for transferring data from FIFO to memory and for transferring data from memory to FIFO. In the table, actions that are the same in both directions are centered. Actions that are specific for the direction are placed in the appropriate column.

Table 5.9. Data Transfer Operating Sequence

Device	From FIFO to Memory	Same	From Memory to FIFO
IOP		1. Programs the DMA for CR register, word count (two's complement), and starting address (24 bits), not necessarily in the order given.	
	2. Sets direction bit to 0.		2. Sets direction bit to 1.
		3. Issues StartDMA.	
		4. Issues AllowRDC to bus arbiter. Note: IOP will have communicated with RDC regarding the I/O control block and the data transfer between rigid disk and FIFO.	
DMA		5. Checks if FIFO is available for transfer.	
		6. Sends hold request (HLD) and waits for HLDA.	
		7. When HLDA is received, transfers one word per memory cycle, as follows:	
	7a. Moves S'lines active (memory write)		7a. Moves S' lines active (memory read)
		7b. Puts 24-bit address on the bus (AD15-00 and AA23-16) during T1 state.	
	7c. Pre-fetches data from FIFO (during last T4 and present T1)		7c. (no equivalent step)
	7d. Puts data on the 80186 bus during T2 and T3 (plus Tw caused by IOPARDY)		7d. Waits until data from memory is stable on the bus (delays for IOPARDY).
	7e. Data written into main memory at T4 state.		7e. Data written into FIFO at T4 state.
		8. Decrements the word count (increments the two's complement).	
		9. Increments the address (24 bits).	
		10. Transfers another word (during the next T1 through T4).	
		11. If HLDA drops, then DMA drops HLD, delays for two cycles, and sends Hold request again.	
		12. If the FIFO is unavailable, then DMA drops HLD, waits for FIFO available, then sends HLD.	
		13. DMA continues to transfer until End-of-Transfer.	
		14. At End-of-Transfer, sends Interrupt to IOP and returns to ready (initial wait) state.	
IOP		15. Once interrupted, will check for End-of-Transfer or error.	
		16. If IOP is interrupted by other I/O during the DMA, then it issues AllowRDC again.	
		17. After DMA interrupt, IOP has control for the next DMA transfer.	

5.3.2.4 Timing

DMA timing behavior is explained in Figures 5.16 – 5.24. A description of each diagram precedes the figure.

The timing diagrams explain:

- typical transfers with zero, one and two wait states
- starting DMA operation
- suspending DMA transfer (and releasing the bus) in response to an Ethernet request or other interrupt to the IOP
- resuming DMA transfer after such a suspension
- delaying and postponing DMA operation in response to unavailability of FIFO
- continuing the operation after FIFO becomes available
- completing DMA operation after all the requested words are transferred

No Wait State

Figure 5.16 illustrates timing for No Wait state. Use this diagram as a reference for the other diagrams as well as for an independent timing chart. The diagram enumerates possible delays in the subsystem when the component variations are taken into account. To better illustrate a starting operation, a Ts state just prior to the first T1 state is also shown.

Delay values are expressed in ns and are measured from the falling (and in a few instances the rising) edge of the ClkOut signal (same as the 8MHzClk on the schematics). S'lines illustrate the S1' or S0' that is low during a memory cycle. If the lines are high, the signal remains in its default level of high without going to low level.

S'lines and ADBus specify the timing tolerances experienced on the IOP Bus while the DMA controller is the bus master.

ALE is not generated or used by this subsystem, yet is essential for the proper function of the DMA transfer. Its timing in any given system provides enough setup and hold time so that it can be used for latching the address that is given by the DMA controller during the T1 state.

IOPARDY and SARDY (synchronized version of IOPARDY with DMA state machine) are assumed to be high at all times. This naturally yields the no-wait condition.

ADDr'/Data is low during those clock cycles in which the AD bus contains an address.

DRead/DWrite' and FIFOPreFetch' are activated by the state machine as specified in the diagram. The signals are generated regardless of the transfer direction, and are only used

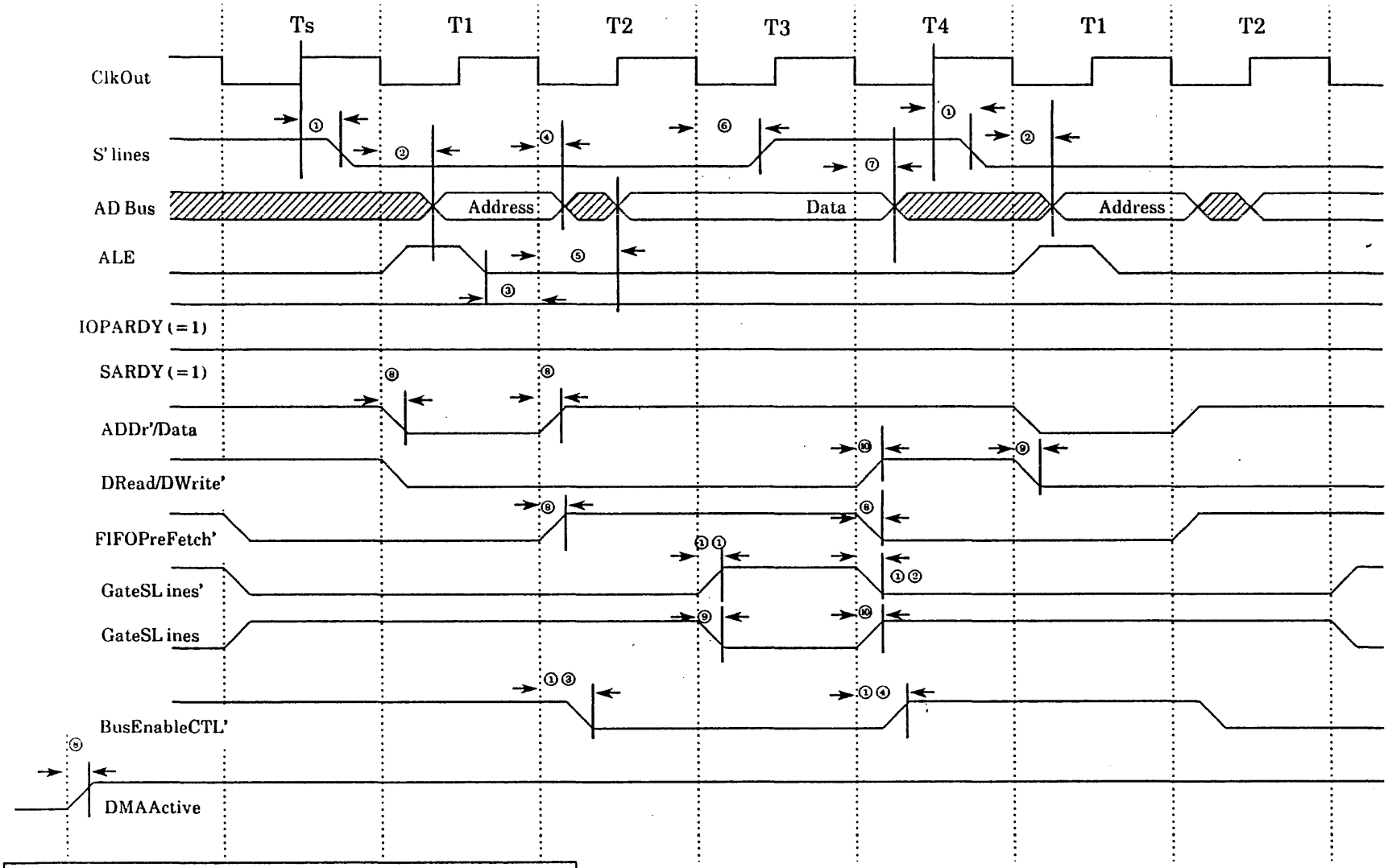
by logic external to the state machine, as determined by the programmed transfer direction.

GateSLines and its inverted counterpart provide appropriate IOP bus status on the S2', S1' and S0' lines, and are also used by other parts of the controller.

BusEnableCTL' enables the AD buffer and delivers the data to or accepts data from the IOP bus, depending on the direction. The signal is generated by the DMA controller while a DMA operation is in progress, and is generated by the IOP 80186 when the IOP attempts to program or query the DMA controller.

Note: The timing specifications on the diagram associated with this signal are valid only during DMA operation and are no longer true when the signal is controlled by the IOP.

DMAActive is a state-machine-generated signal used in various areas of the controller to indicate that a DMA cycle is running.



Key: Minimum/Typical/Maximum in ns. N = not defined.

① N/16/24	⑤ 14/N/34.5	⑨ N/11/20	①③ N/9.5/14.5
② N/N/27	⑥ N/11/36	⑩ N/14/22	③④ N/23.5/36
③ 2/N/N	⑦ 10/N/43.5	①① N/8/15	
④ 8/N/N	⑧ N/6/9	①② N/11/17	

Notes:
1) The signal "S'Lines" represents only the signals among S1', S2', and S0' that go low during a memory cycle. (For the case like S2' = 1, obviously the signal remains high)
2) In all timing values, neither the effect of clock skew nor the delay between ClkOut and ClkOut' is included

Figure 5.16. DMA timing: No wait states

Figure 5.17 illustrates timing for one wait state. One wait state is similar to no-wait state, except that a narrow pulse on the IOPARDY signal has produced a ShavedARDY signal. Later, ShavedARDY creates a suitable pulse on SARDY. This refined signal is input to the state machine and forces it to introduce one Tw (wait) state between T3 and T4.

Note that in order to create the Tw state, the IOPARDY signal must meet the minimum requirements given on this diagram. IOPARDY can go low earlier than is required (e.g., early in T2 or sometime during T1), without causing a problem and will simply be ignored by the state machine. To achieve one Tw state, the signal must be low at the end of T2 and must be high prior to the second half of the T3 state (see Figure 5.18).

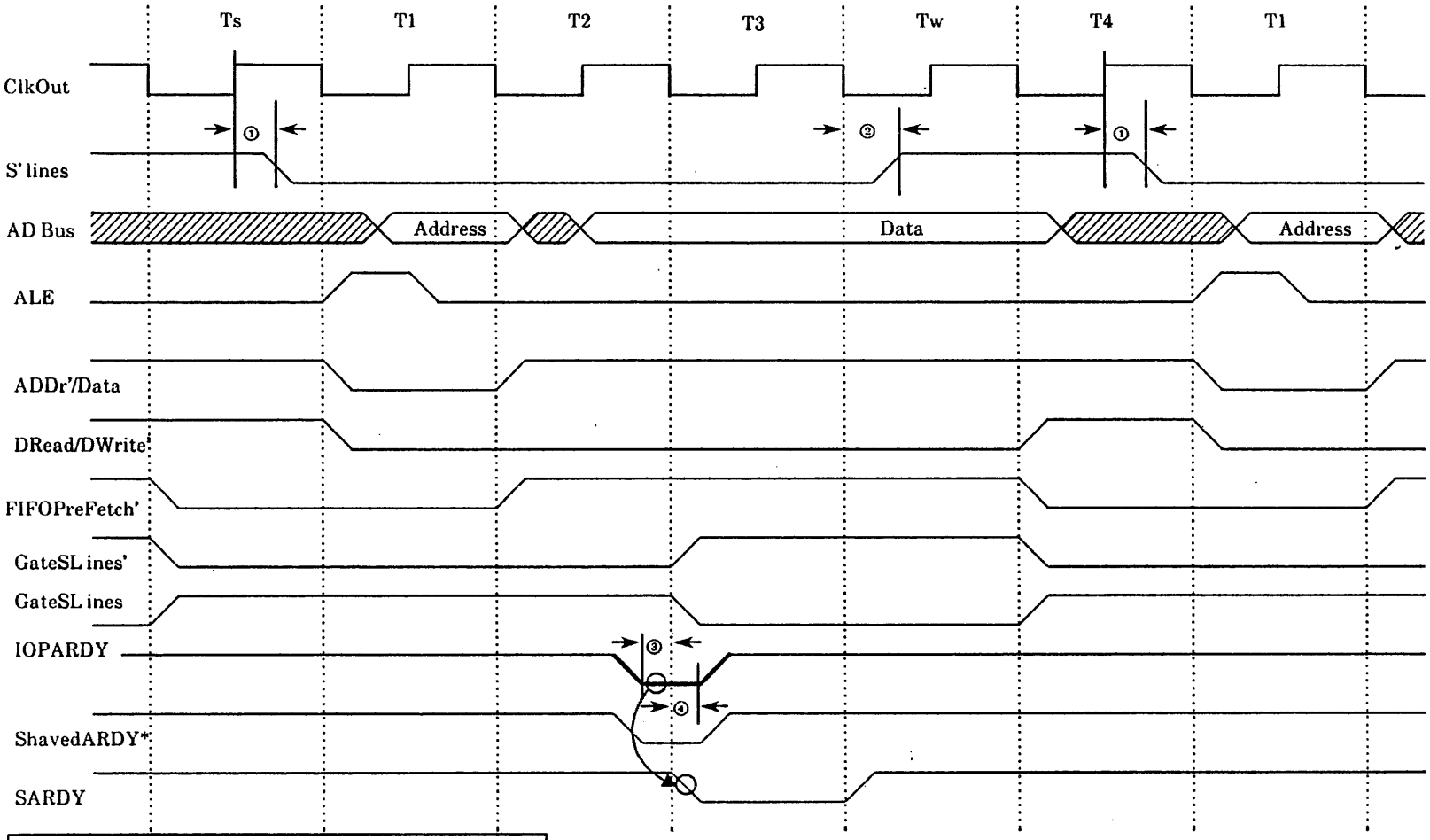


Figure 5.17. DMA timing: One wait state

Figure 5.18 illustrates timing for two wait states. The timing is similar to the two previous figures. Two Tw states are introduced by the length of the negative pulse on IOPARDY signal (which has maintained its low level during the first half of the T3 state).

This pulse in turn has been modified to become a synchronized pulse on the SARDY signal for a period of two T cycles. The state machine responds by introducing two Tw cycles. A pulse longer than these cycles on the IOPARDY signal yields more Tw states accordingly. There is virtually no limit to the number of Tw states that can be introduced. The memory cycle can be extended for a long period of time. Only when the low level signal is removed from the IOPARDY signal will the system resume the normal operation as before without loss of data.

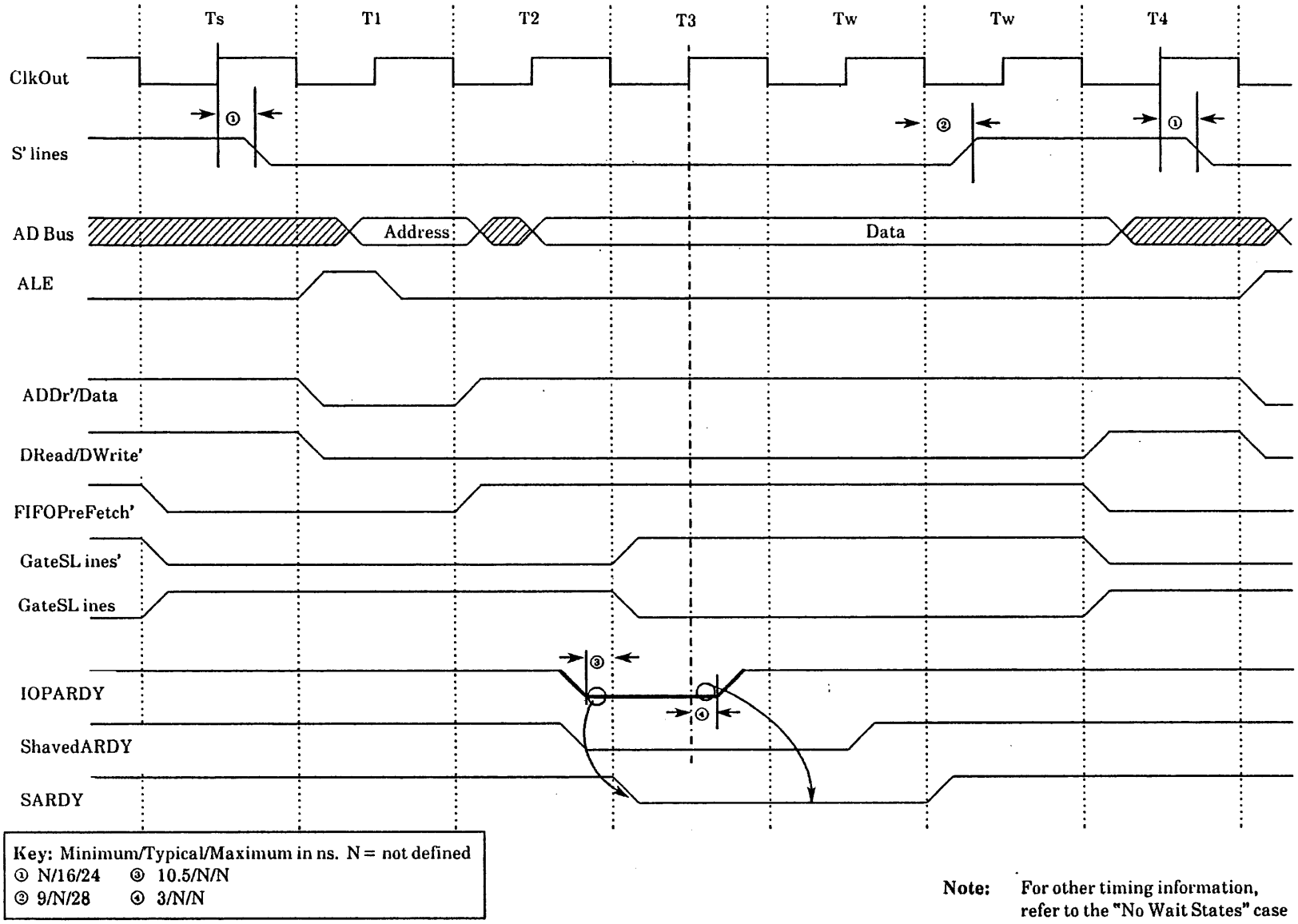
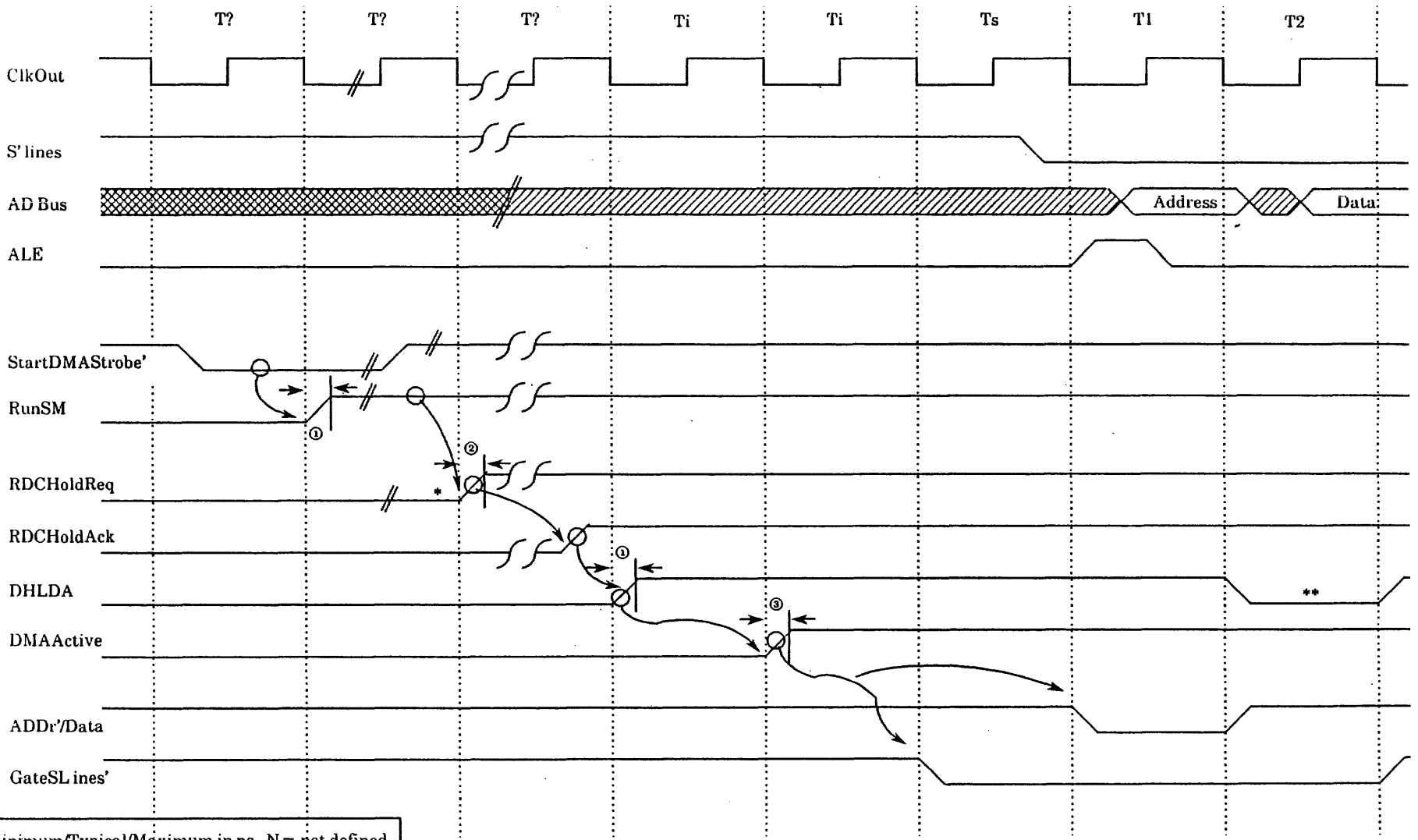


Figure 5.18. DMA timing : Two wait states

Figure 5.19 illustrates starting DMA operation. Issue of a StartDMA command by the IOP causes a negative pulse on StartDMAStrobe' that in turn moves RunSM to a high level.

If the FIFO is available, then the state machine moves RDCHoldReq high to request the bus mastership. After delay for a number of cycles (depending on bus traffic and on the status of the bus masters), the bus arbiter responds by moving RDCHoldAck high.

A refined version of this signal, DHLDA, informs the state machine of IOP bus availability. The state machine initiates the DMA operation activity by moving DMAActive high. The next cycle is a Ts cycle; the following cycles will then proceed as shown in the previous figures.



Key: Minimum/Typical/Maximum in ns. N = not defined
① N/13/25
② N/8/15
③ N/6/9
* Delay if FIFOOB' = 0
** Ignored by State Machine
(Assuming Zero Wait state case)

Note: For other timing information, refer to the "No Wait States" cas

Figure 5.19. DMA timing: Starting DMA operation

Figure 5.20 illustrates Ethernet- or interrupt-originated DMA suspension, which results in dropping the RDCHoldReq line.

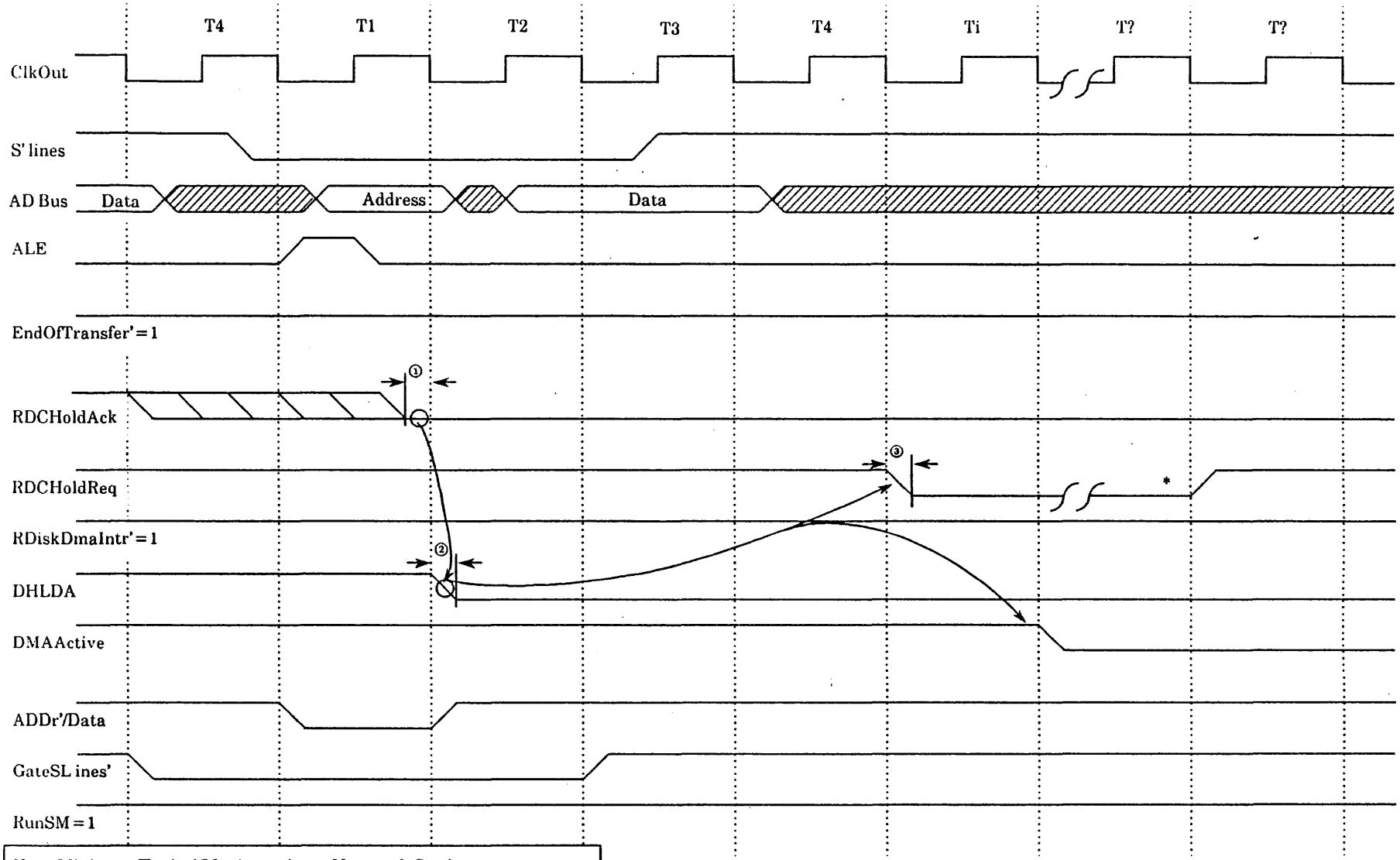
If Ethernet requests the bus mastership while DMA is in operation or if an interrupt requires the IOP's attention to the interrupt source, then the bus arbiter informs the DMA controller by moving RDCHoldAck low, to signal the need to relinquish the bus.

This change is relayed to the state machine as DHLDA. The state machine then proceeds with the current transfer until reaching T4 state. After that, without inserting another bus cycle (note the S'Lines), the state machine drops the RDCHoldReq line. One cycle later it clears the DMAActive signal.

The state machine then waits one more T cycle. If the FIFO is available for additional transfers, the DMA raises the RDCHoldReq and waits for a response from the bus arbiter (without exercising any bus activities).

This delay allows the bus arbiter to determine if the bus is available for another bus master. Reinsertion of RDCHoldReq has no immediate impact on the bus mastership, but instead confirms that the DMA controller can resume its operation as soon as the bus is available again.

Note that while DMA controller leaves the bus, RunSM remains high to inform the system (in particular the IOP) of the incomplete DMA operation.



Key: Minimum/Typical/Maximum in ns. N = not defined
① 25/N/N
② N/25/40
③ N/11/17

* A minimum of two clock clockcycles for "RDCHoldReq = 0" is guaranteed. Delay beyond that is introduced only if the FIFO is Out of Bound (Assuming Zero Wait state case)

Note: For other timing information, refer to the "No Wait States" case

Figure 5.20. DMA timing: Ethernet- (or any interrupt-) originated Drop Disk Hold Request

Figure 5.21 illustrates the continuation of DMA operation after the Ethernet- or interrupt-originated DMA suspension.

At an Ethernet-originated DMA suspension, the DMA controller re-inserts its RDCHoldReq shortly after it leaves the bus (if the FIFO is available) to wait for the acknowledge. As soon as the arbiter responds with RDCHoldAck, and after the other bus master has completed its task and relinquished the bus, the state machine activates the DMAActive signal. A normal DMA transfer is then resumed at the beginning of the next cycle.

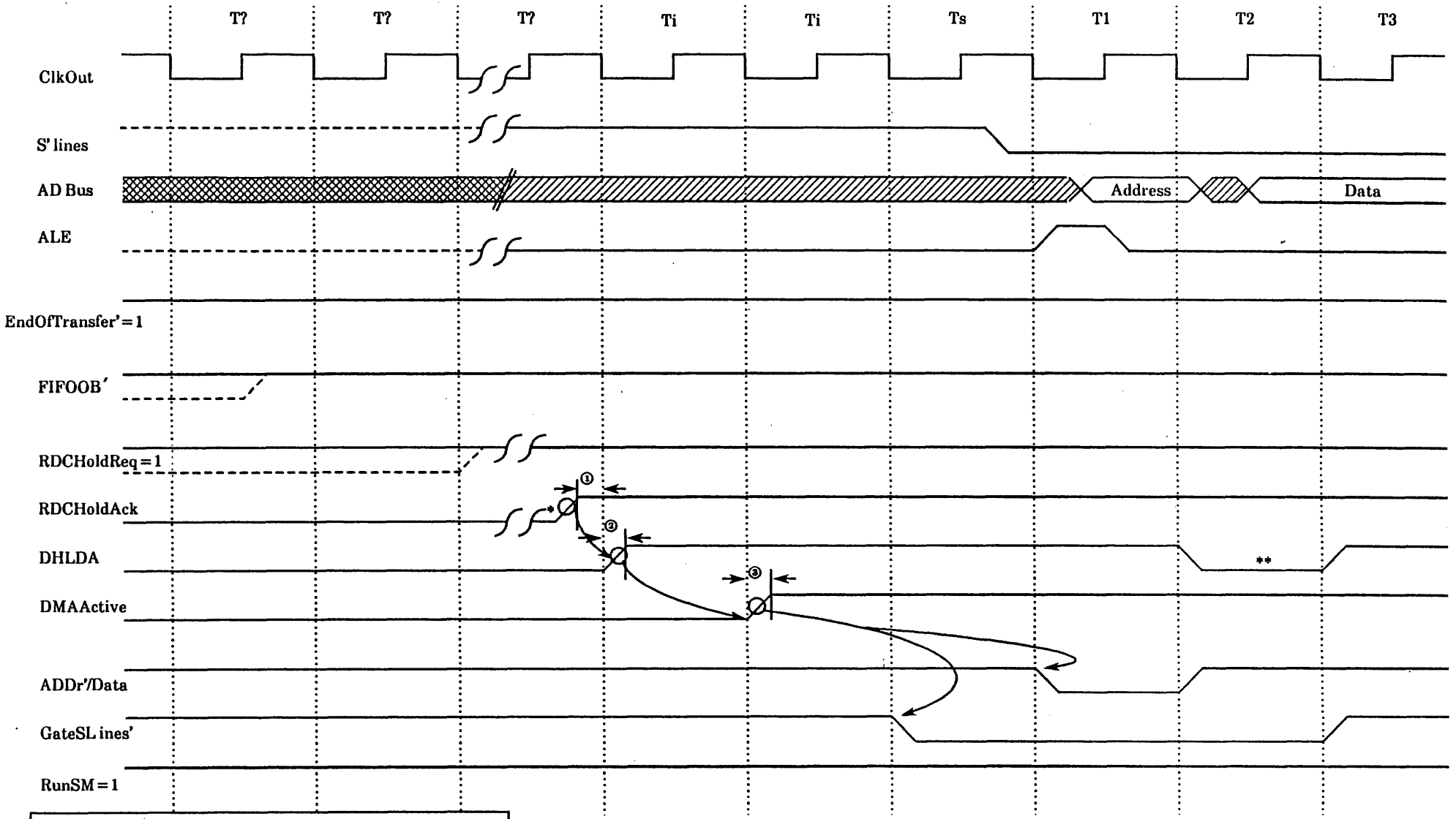
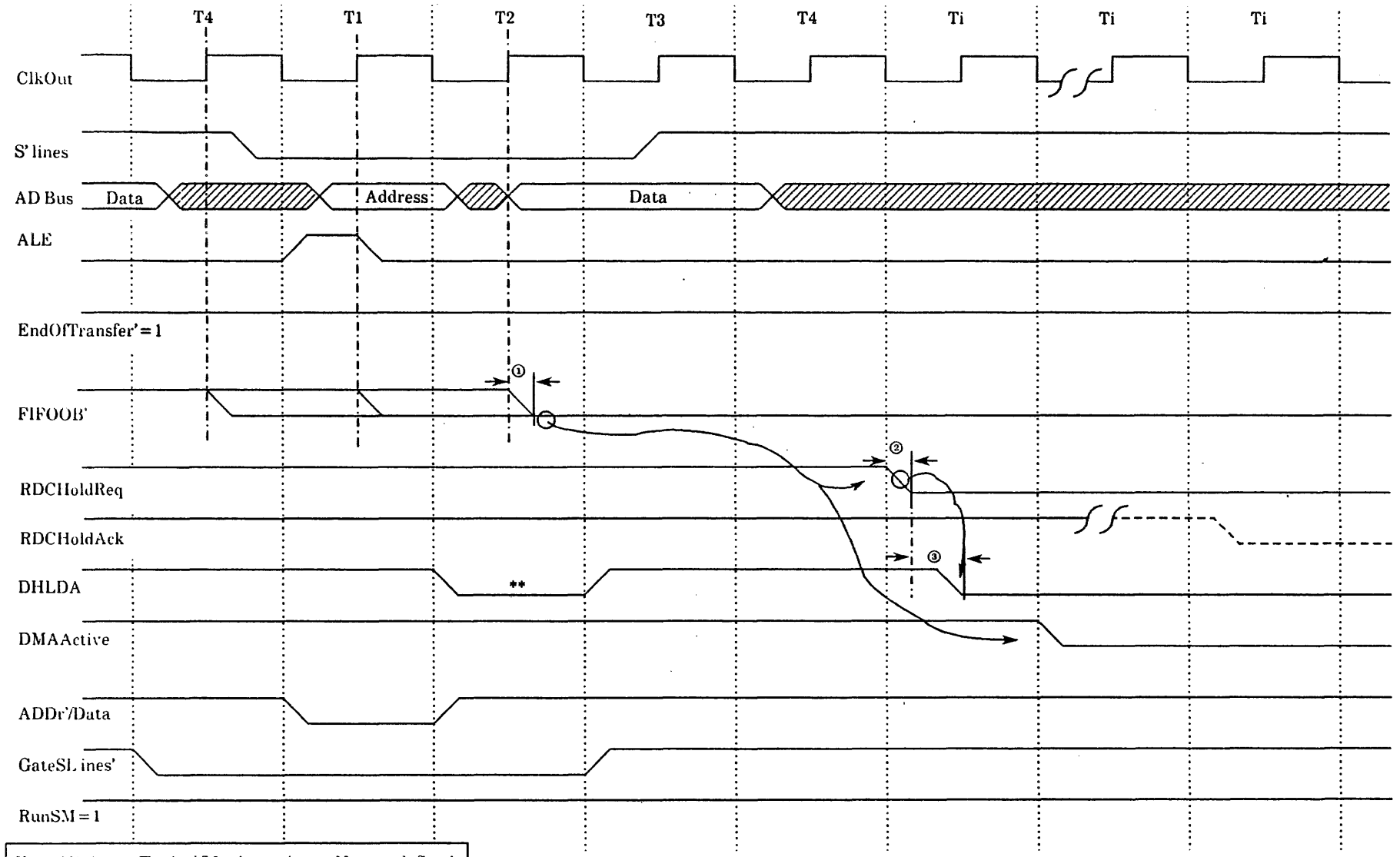


Figure 5.21. DMA timing: Continuing DMA operation after Ethernet- (or interrupt-) suspension

Figure 5.22 illustrates a FIFO-originated (FIFOOB) DMA suspension, which results in dropping the RDCHoldReq line. In the event the FIFO is not available (FIFOOB'=0) due to possible delays from the rigid disk, the DMA controller does not keep the IOP bus occupied. Instead, the DMA controller relinquishes the bus for use by other bus masters. The sequence is similar to the previous sequence (Figure 5.20), except that the next RDCHoldReq does not occur until after the FIFO is indeed available and a successful transfer involving the FIFO can be achieved.



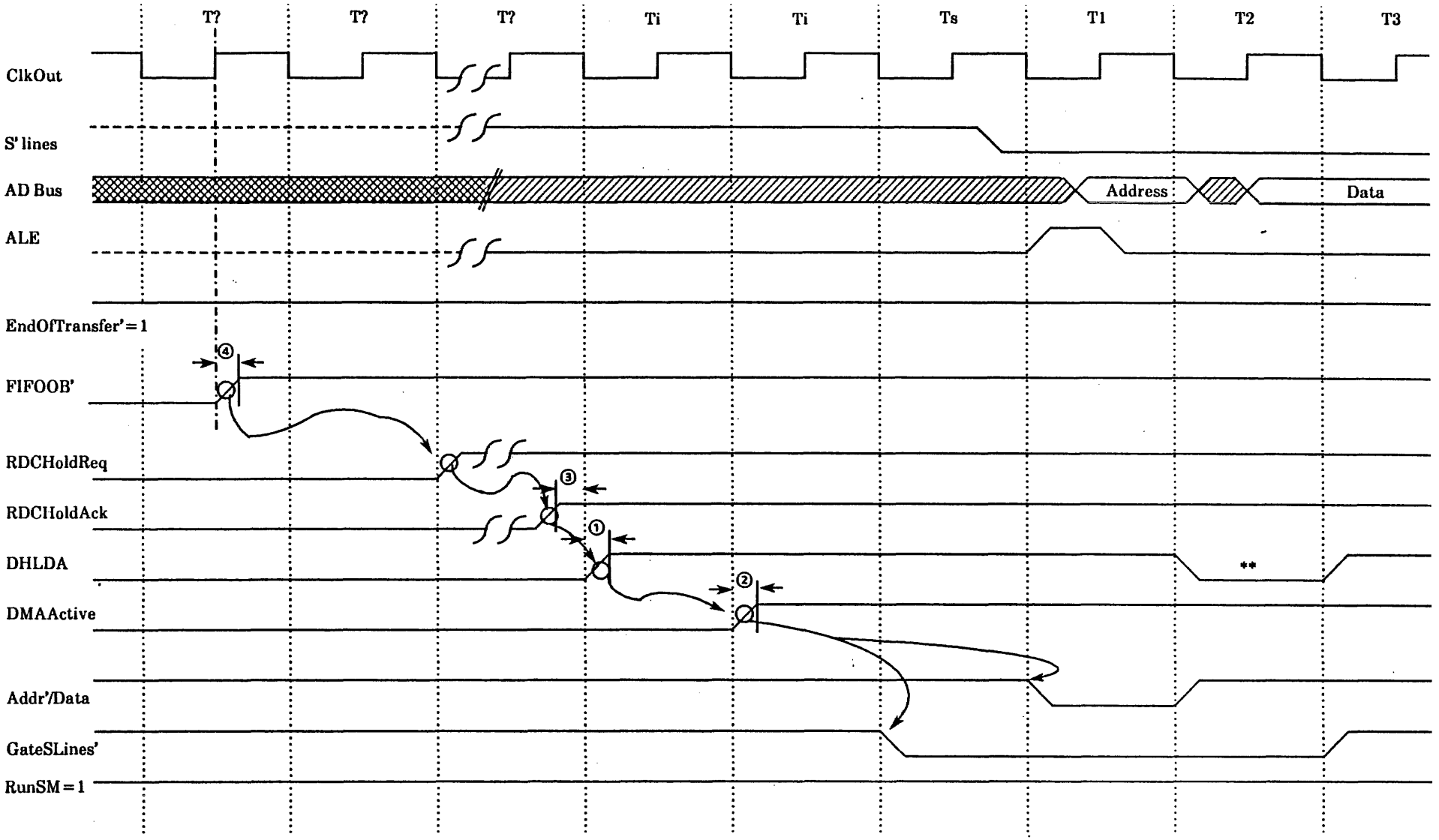
Key: Minimum/Typical/Maximum in ns. N = not defined
① N/16/24
② N/N/27
③ N/25/40

** Ignored by State Machine
(Assuming Zero Wait state case)

Figure 5.22. DMA timing: FIFOOB Drop Disk Hold request

Figure 5.23 illustrates the continuation of DMA transfer after a FIFO-originated (FIFOOB) DMA suspension. The sequence is much like that shown in Figure 5.21.

In response to the FIFO available (FIFOOB' = 1), the state machine moves RDCHoldReq high and, as before, waits for RDCHoldAck.



Key: Minimum/Typical/Maximum in ns. N = not defined
① N/13/25 ③ 35/N/N
② N/8/15 ④ N/8/12

** Ignored by State Machine
(Assuming Zero Wait state case)

Note: For other timing information, refer to the "No Wait States'

Figure 5.23. DMA timing: FIFOOB Continuing DMA operation

Figure 5.24. illustrates timing for ending DMA operation.

The end of DMA operation is signalled to the state machine as well as to the system by the signal EndOfTransfer'. Whenever the signal goes low, it indicates that the number of words transferred is equal to the number of words programmed (Word Count).

The transition from high to low on this signal causes the state machine to complete the DMA transfer. The combination of interrupt to the IOP (RDiskDmaIntr'=0) and RunSM=0 is the unique feature of this time interval.

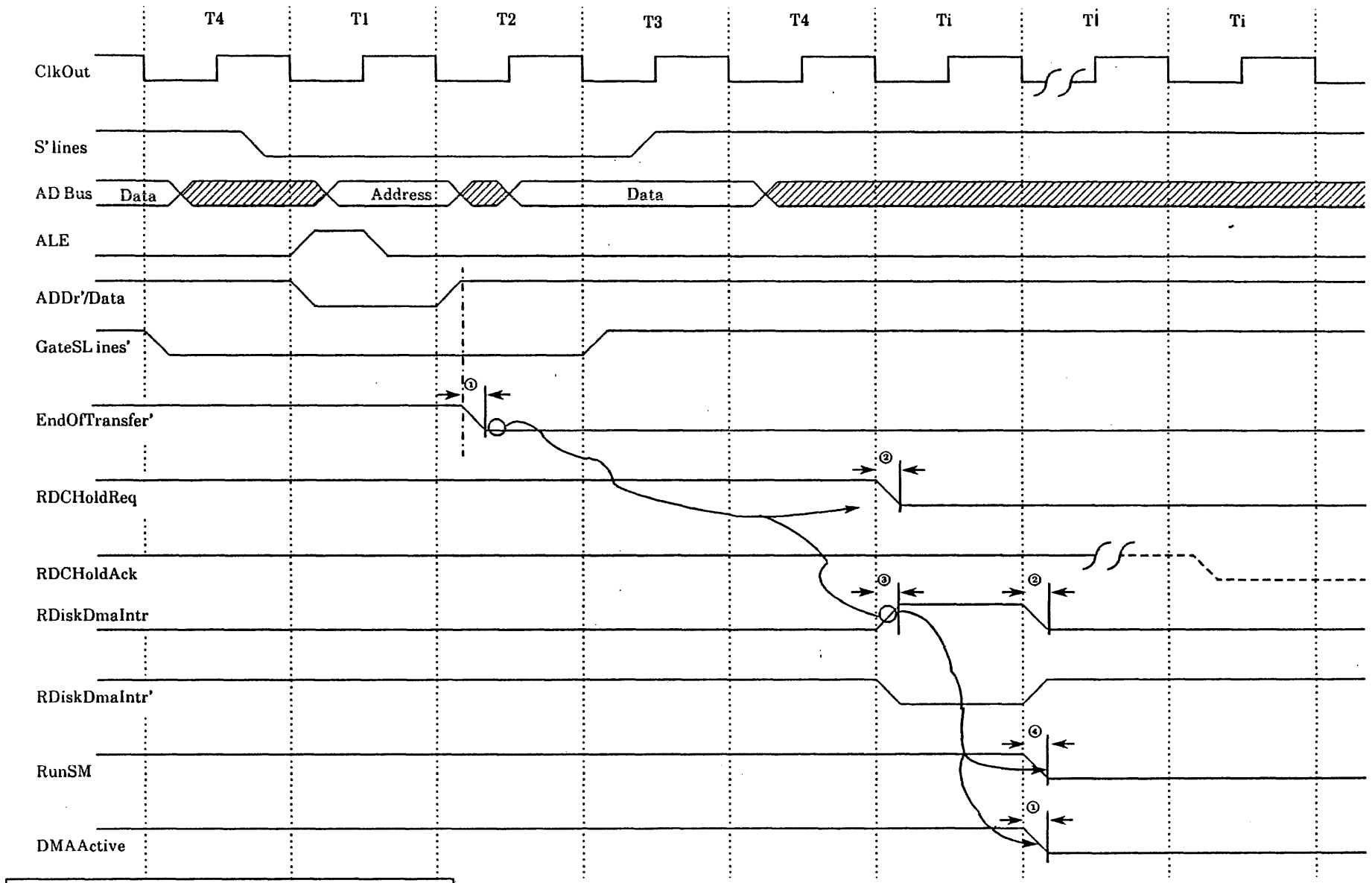


Figure 5.24. DMA timing: Ending DMA Operation

5.3.3 Programmer Interface

The DMA is programmable by the IOP. Registers are accessed as I/O ports of the 80186 and reside within its I/O address space. Some I/O operations perform more than one function. Since not all of the registers have read and write access, the IOP uses the "in" and "out" instructions in accessing them.

The registers that relate to the DMA function are:

- Starting Address Register – DMA address bits 8-1.
  Starting Address Register – DMA address bits 23-9 and bit 0.

- Current Address Register – DMA Address bits 8-1. Changes during DMA transfer to provide a current address.

Note: The higher significant bits of address (bits 23-9) remain unchanged during a maximum of 256 word transfers. Transfers are designed in software to be at page boundaries. If fewer than 256 words are transferred, then the transfers are within the same page.

Note: The DMA address bit 00 for all DMA transfers must be 0. Rigid disk and DMA related address/data/control/status are at word boundaries.

- Word Count Register – Keeps count of the number of words remaining to be transferred.

- Word Count Shadow Register – Maintains the original value given as word count; that is, number of words to be transferred in one DMA operation.

- Control Register within AM2942 (CR2-CR0) – Used by the IOP to program the functioning mode of the AM2942.

The IOP uses four registers within the RDC/FIFO/DMA to communicate with the rigid disk subsystem. They are:

- DMA command register

- DMA status register

- Rigid disk controller command register (described in subsection 5.2.3.2)

- Rigis disk controller status register (described in subsection 5.2.3.2)

Table 5.10 summarizes the I/O addresses.

Table 5.10. I/O Addresses

I/O Address	Description
0200H	Write CR2-CR0 (inside AM2942)
0202H	Read CR2-CR0 (inside AM2942)
0204H	Read word count (in two's complement form) (inside AM2942)
0206H	Read address counter bits 8-1 (inside AM2942)
0208H	Load starting address bits 23-9 (& 0) and re-initialize counters
020AH	Load starting address bits 8-1 (inside AM2942)
020CH	Load word count (two's complement) (inside AM2942)
020EH	Enable counters (inside 2942) -- Not used
0210H	DMA command/status registers (Write = command reg, Read = status reg)
0212H	Preset FIFOIn17(Write) (for Diagnostics)
0214H	Rigid disk controller's control/status registers (Write = control reg, Read = status reg)
0216H	Start DMA (Write)
0218H	Not Used
021AH	Not Used
021CH	Not Used
021EH	Not Used

5.3.3.1 Address Register

Figure 5.25 illustrates the 24-bit starting address register.

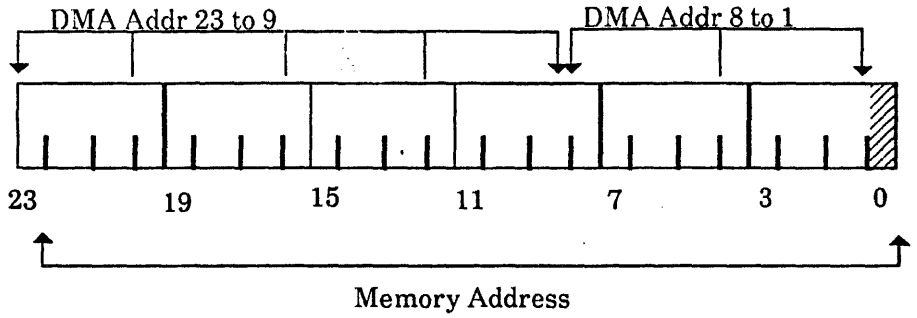


Figure 5.25. Starting address register

The 24-bit address is given by two I/O write commands:

Bits 8-1 => Write to address 020AH while data is given on bits 8-1 of the data bus. Bit 00 of data bus is ignored and can be 0; that is, a word address can be given without a shift.

Bits 23-9 => Write to address 0208H while data is given on bits 15-1 of the data bus. Bit 15 of the data is forwarded as the most significant bit (bit 23), and bit 1 will appear as bit 9 of the 24-bit starting address register.

Note: The instruction for writing bits 23-9 of starting address register will automatically re-initialize both the word counter and the starting address register bits 8-1. Their current value, if different from the original, is changed and becomes equal to the original.

Whenever DMA is not the bus master, IOP can read the current word count and part of the current address.

Only bits 8-1 of the latest address are accessible by the IOP. Reading is done by an I/O read "In"function from 0206H. Bits 8-1 of the current address will be delivered on bits 8-1 of the AD bus as data. The user should ignore Bit 00 of the AD bus and bits 15-9. Address register bits 23-9 are not accessible.

5.3.3.2 Word Count

The word count is 8 bits wide. A maximum of 256 words can be transferred in one DMA operation. The count should be given over the AD bus bits 8-1, shifted left by one, in the form of two's complement of the actual number of words to be transferred. Table 5.11 lists word count examples.

Table 5.11 Word Count Examples

To transfer...	Load the word count register with...	This means: write...
256 words	00H	0000H
1 word	FFH	01FEH
2 words	FEH	01FDH

An I/O write command "Out" to address 020CH with the word count on the data bus as specified above implements the write function.

An I/O write command "In" to address 0204H results in the word count on the data bus as specified above.

5.3.3.3 Control Register on AM2942 Chip

The AM2942 chip within DMA is used in function mode 3; mode 3 requires the word count register and word counter to be loaded with the two's complement of the number of data words to be transferred.

An I/O write to address 0200H loads bits 3-1 of the data bus into bits 2-0 of the AM2942 control register. Because the chip is used in mode 3, the contents of CR register must be CR2-0=011. This translates to writing 0006H to the I/O port; note the shift to the left by one bit.

A read from this register may be done by an I/O read from address 0202H. Contents of the register are provided on bits 3-1 of the data bus; other bits should be ignored.

Note: For details on programming the starting address and word count, as well as for mode 3 specifics, please refer to the AM2942 specifications.

5.3.3.4 DMA Command Register

The DMA command register is a write-only register at I/O address 0210H. Bits 01 through 15 are not used; bit 00 indicates FIFO direction.

Figure 5.26 illustrates the DMA command register.

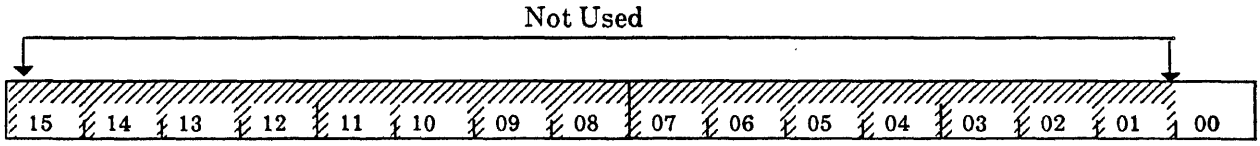


Figure 5.26. DMA command register (Read only I/O Addr = 0210 hex)

where:

Value = 1 is the direction from main memory to disk.
Value = 0 is from disk to main memory.

Note: After each reset exercised on the DMA controller, the transfer direction defaults to disk-to-memory. If the alternate direction is desired, it must be programmed prior to activating DMA operation.

5.3.3.5 DMA Status Register

The DMA status register is a read-only register at I/O address 0210H. The register has the characteristic of reset-on-read. Bits 08 through 15 are not used.

Figure 5.27 illustrates the DMA status register.

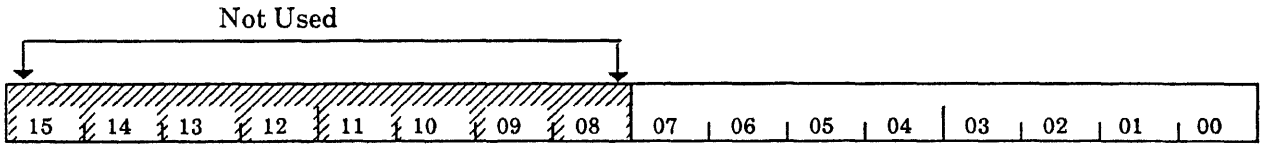


Figure 5.27. DMA status register (Read only I/O Addr = 0210 hex)

where:

Bit 00 – Error
The bit normally is 0, but becomes 1 when the DMA state machine encounters an error condition. In such a case, all other bits of this register maintain the information held just prior to occurrence of the error. The bit remains 1 until a

reset occurs. Note that a read from the status register will not clear this bit or cause the register to be updated.

Bit 01 - RunSM
The bit is 1 when DMA has not completed a DMA transfer; that is, when a DMA operation is in progress. After a RDC Reset, or an end-of-transfer, this bit is 0. Only a StartDMA command will change the bit to 1.

Bit 02 - EndOfXfer'
The bit is 0 when a DMA transfer is completed and DMA logic has no other request pending. After RDC Reset, this signal has no meaning until the first StartDMA command is issued.

Bit 03 - FIFOOutOfBound'
The bit is 0 if FIFO is found empty and the FIFO is in a disk-to-memory direction, or if FIFO is full and is in memory-to-disk direction.

Bit 04 - FIFOEmpty'
The bit is 0 if FIFO is empty.

Bit 05 - FIFOFull'
The bit is 0 if FIFO is full.

Bit 06 - FIFOOut17
This bit is the output of the FIFO for a status bit carried along with data through the FIFO. This output has meaning only when compared with FIFOIn17, the input to FIFO for the same bit. In addition to the internal use of FIFOIn17and FIFOOut17 by the disk controller during DMA transfer, these two bits can be used to monitor transport of data words through the FIFO.

Bit 07 - FIFODIR
The loopback of the direction bit from DMA command register; the bit can be used to verify the direction of transfer later in the programming cycle or during diagnostics.

5.4 Rigid Disk FIFO

To further isolate the processor/main memory subsystem from the inherent latency characteristics of the rigid disk, a temporary storage is provided in the form of a First-In First-Out (FIFO) buffer between main memory and the rigid disk. The temporary storage transfers data or disk control blocks (DCB) between the rigid disk/rigid disk controller at one end and main memory at the other end.

The main features of the FIFO are:

- Sizable data buffering capability (512 words long = 2 Mesa pages = 2 disk sectors), which minimizes the impact of disk latencies on Ethernet and DMA operations
- Programmable bi-directional access (memory-to-disk and disk-to-memory)
- Simultaneous and asynchronous access by DMA controller and rigid disk controller
- Full, half-full and empty indications
- Reset capability at power-up and under program control

Figure 5.28 illustrates the rigid disk FIFO.

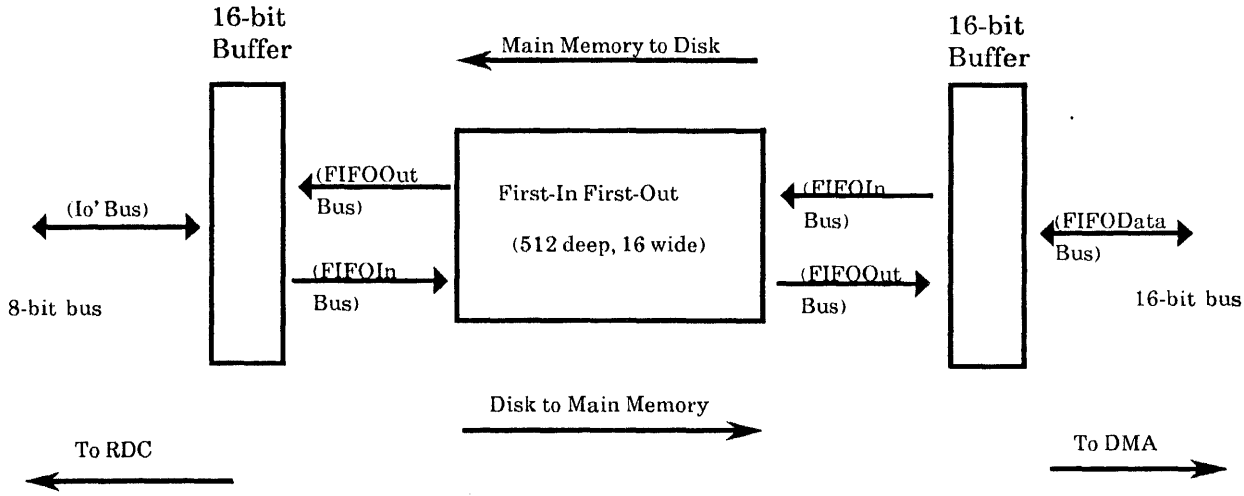


Figure 5.28. FIFO block diagram

5.4.1 Hardware

The FIFO consists of two 512 x 9 components that arrange a 512 x 18 structure, yielding a 512 x 16 logical FIFO block plus one bit for FIFOIn17/FIFOOut17 and one unused bit. Additional logic provides the half-full capability and appropriate buffering at both ends such that data is received by the rigid disk controller in byte quantities and received by DMA and by the 80186 bus in word quantities.

Two signals that affect the FIFO are described below:

FIFODIR	Programmable by IOP, indicates the direction of DMA transfer: 1 = memory to disk 0 = disk to memory
FIFOOutOfBound	Depending on DMA transfer direction (FIFODIR), sets a flag when the FIFO is not capable of servicing the DMA operation; that is, when FIFO is full and direction is from memory to disk or when FIFO is empty and direction is from disk to memory.

5.4.2 Theory of Operations

Depending on FIFODIR (direction bit), either the DMA controller or rigid disk controller may write into the FIFO; the alternate source is able to read from it. The contents are readable in the same order as written. FIFO status is available to both sources. IOP software ensures that the direction bit is correct and that it does not change until a data transfer is successfully completed from the origin to destination.

After the IOP programs the DMA controller, the controller responds to a StartDMA and initiates a DMA transfer in the programmed direction. The DMA begins with the word at the location given by the starting address, and transfers words with consecutive addresses up to the total number in the word count. Transfer occurs at the rate of one word per memory cycle at the memory speed. The DMA transfer may be interrupted according to the bus arbiter's programmed priority scheme. The transfer resumes from the same point where interrupted and proceeds either to the end of DMA or until the next interruption.

For the DMA controller, transfer in the disk-to-memory direction is subject to availability of data in the FIFO. Transfer in the memory-to-disk direction is subject to availability of storage space in the FIFO. If storage space is not available during the transfer, then the DMA controller automatically relinquishes the 80186 bus and waits until FIFO is available again. The DMA controller then requests the control of the bus. After control is granted, DMA transfer resumes from the point at which it was suspended. The process is described in detail in section 5.3.2.

The rigid disk controller and DMA circuitries operate asynchronously and independently. No direct communication facilities exist between them, and therefore they do not exchange information about the contents of the FIFO; that is, whether the contents are I/O control block or actual data. System software ensures data integrity, using the process informatiion and the hardware facilities (rigid disk controller and DMA controller) at both ends of the FIFO.

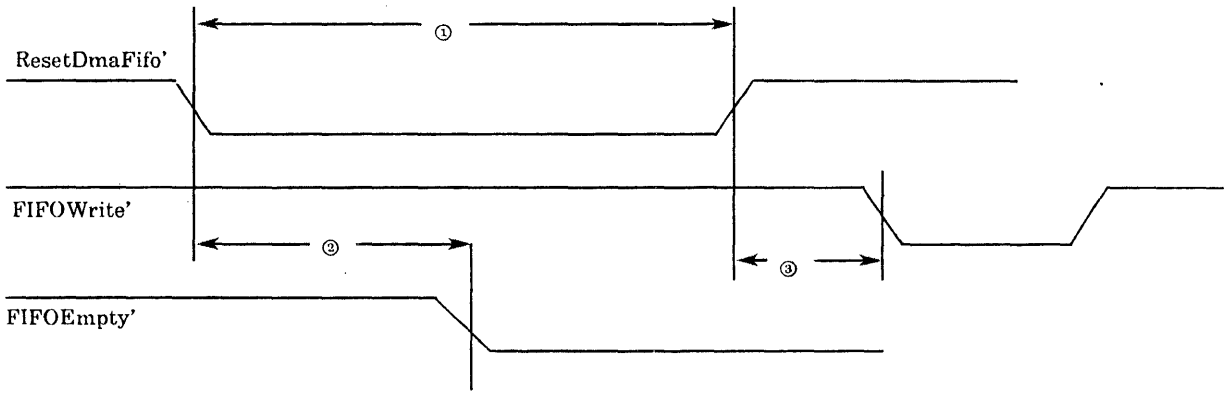
Since FIFO contents are treated in a first-in, first-out manner, particular attention should be paid to the case of reading from the disk. For example, when a DCB is sent to the rigid disk controller, the direction will be memory to disk. Then, in order for data to be transferred from the disk via FIFO to the main memory, the direction must be changed: disk to memory. Next, in order for memory to receive the updated DCB from the rigid disk controller, the direction bit must remain disk to memory.

5.4.3 Programmer Interface: Timing

The components used for the FIFO implementation of this system allow asynchronous access to both ports of the FIFO (input and output). If the timing requirements reflected on the following diagrams are met, then accesses may also occur simultaneously.

Figures 5.29 – 5.32 illustrate timing for the FIFO.

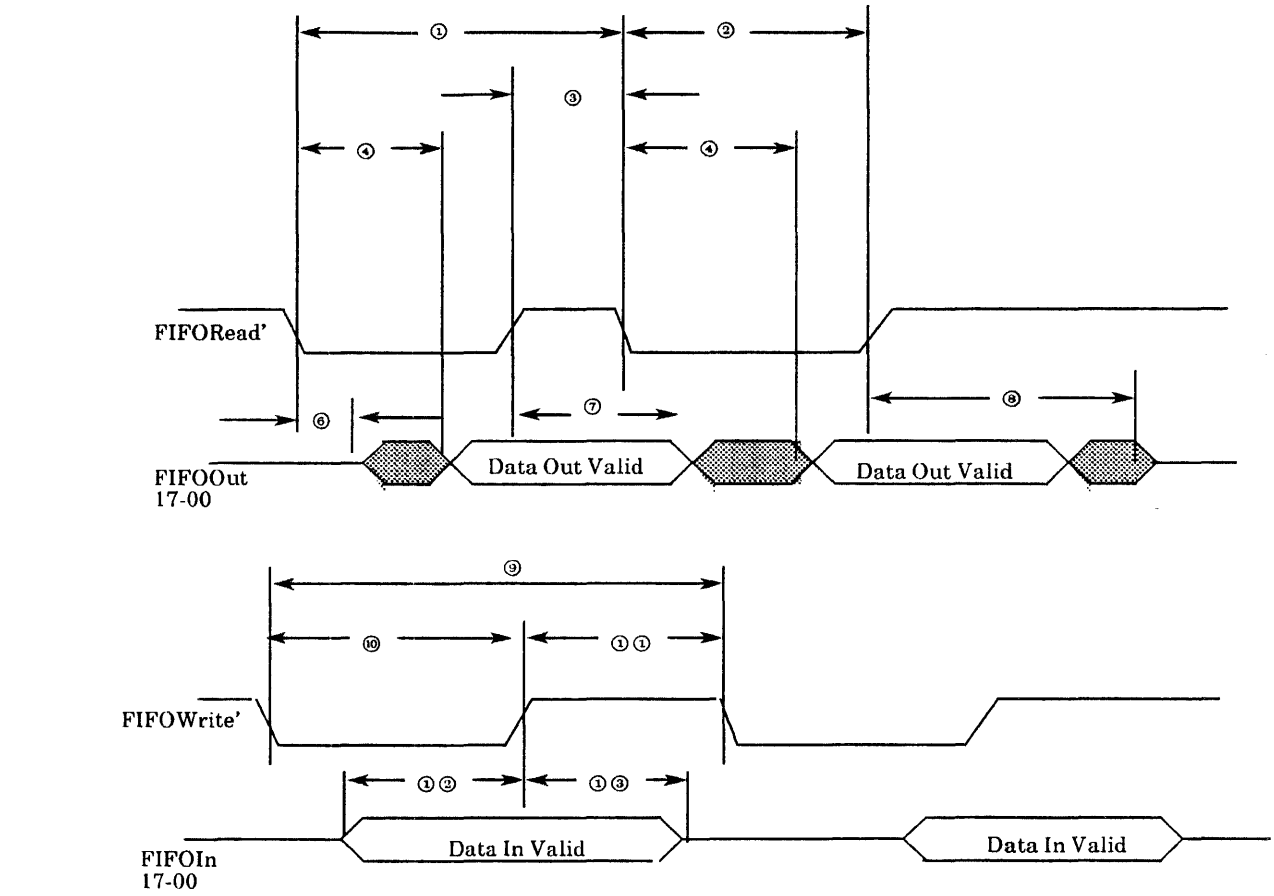
Figure 5.29 illustrates Empty Flag timing with respect to the reset condition. In order for an empty or full flag to be activated by the FIFO, certain minimum delays between the last read and the first write (for empty) and the last write and first read (for full) must occur, as shown in figures 5.31 and 5.32. Two consecutive accesses on one end allow enough time for such a flag to register. However, the system will continue to function even with no flag signal generated, because the rigid disk controller and the DMA controller operate asynchronously at each end of the FIFO. No interruption in the transfer process occurs until one side delays long enough to allow the other side to make more than one access during the delay period. To guarantee the correct operation, two consecutive read or write operations should have a minimum distance from each other and minimum pulse width. (See Figures 5.30 and 5.31.)



Key: Minimum/Typical/Maximum in ns. N = not defined

① 120/N/N
② N/N/140
③ 20/N/N

Figure 5.29. Reset timing



Key: Minimum/Typical/Maximum in ns. N = not defined		
① 140/N/N	⑤ 20/N/N	⑨ 120/N/N
② 120/N/N	⑥ 5/N/N	⑩ 20/N/N
③ 20/N/N	⑦ N/N/35	①① 40/N/N
④ N/N/120	⑧ 140/N/N	①② 10/N/N

Figure 5.30. Asynchronous write/read timing

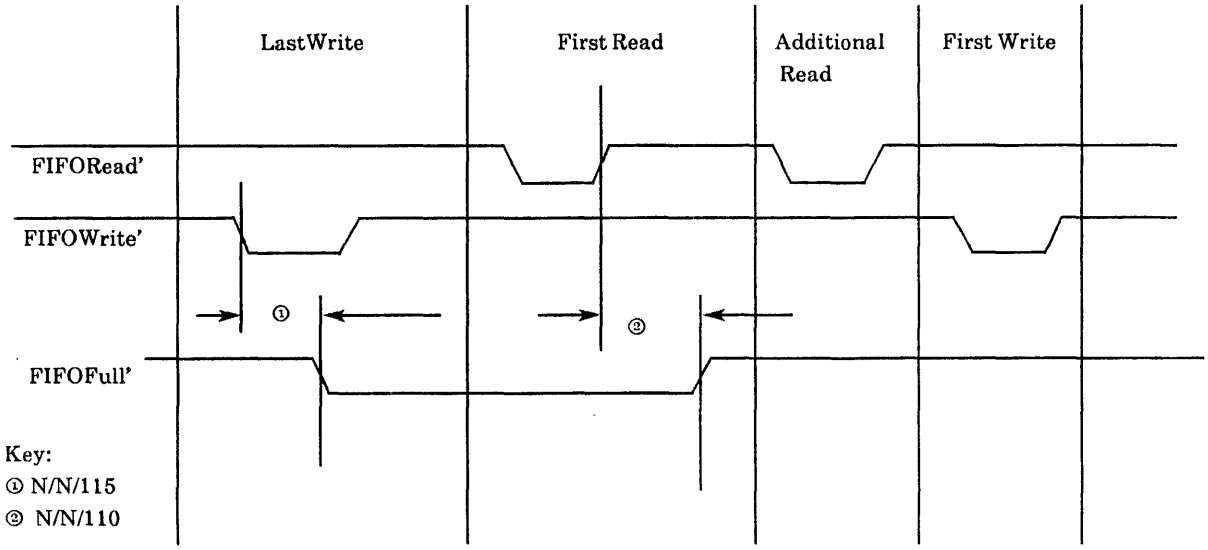


Figure 5. 31. Full flag from last write to first read

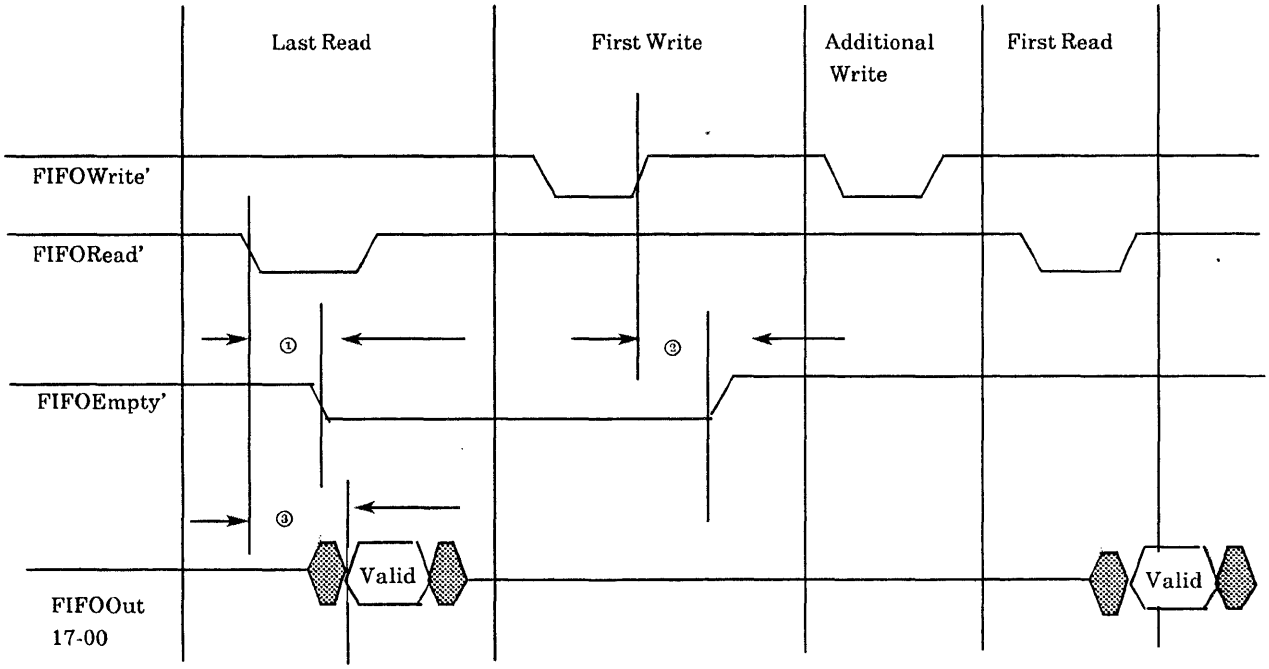


Figure 5.32. Empty flag from last read to first write

6 Ethernet Controller

The Ethernet controller provides a connection between the Dove workstation and the Ethernet.

The controller uses an integrated data link controller (Intel 82586), and an integrated serial interface (SEEQ 8002). Two off-the-shelf chips essentially implement the controller. The Ethernet controller is compatible with the IEEE 802.3 standard. Figure 6.1 illustrates the components of the ethernet controller and their relationship to the IOP board.

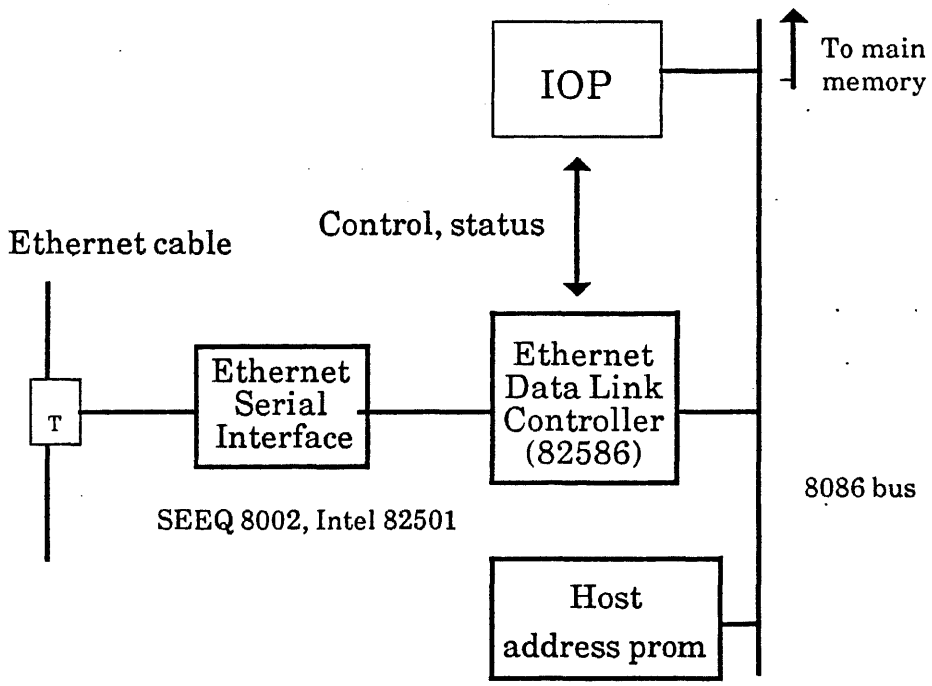


Figure 6.1. Ethernet controller block diagram

6.1 Hardware

The following figures and tables illustrate and list the pin descriptions for the necessary ethernet controller hardware. They are as follows:

- Table 6.1, Figure 6.2 Ethernet Data Link Controller (INTEL 82586)
- Table 6.2, Figure 6.3 Ethernet serial interface (SEEQ 8002)
- Table 6.2, Figure 6.4 Ethernet Serial Interface (Intel 82501)

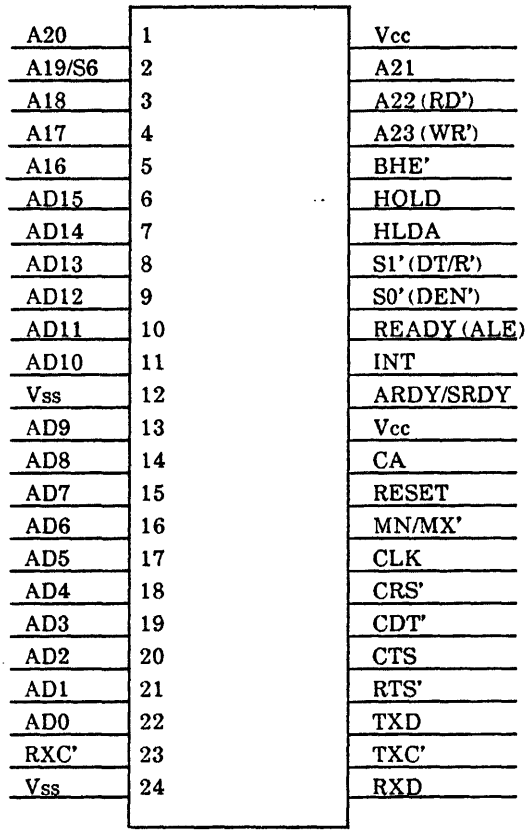


Figure 6.2 82586 pins and signals
(Reprinted by permission of Intel)

Table 6.1 82586 Pin Assignments
(Reprinted by permission of Intel)

Symbol	Pin No.	Type	Function
VCC,VCC	48,36		System Power: =5 volt power supply
VSS,VSS	12.24		System Ground
RESET	34	I	Reset is an active HIGH internally synchronized signal that causes the 82586 to terminate present activity immediately. The signal must be HIGH for at least four clock cycles. The 82586 will execute RESET within ten system clock cycles starting from RESET HIGH. When RESET returns low, the 82586 waits for the first CA to begin the initialization sequence.
TxD	27	O	Transmitted serial data output signal. This signal is HIGH when not transmitting.
TxC'	26	I	Transmit Data Clock. This signal provides timing information to the intrnal serial logic, depending upon the mode of data transfer. For NRZ mode of operation, data is transferred to the TxD pin on the HIGH to LOW clock transition.
RxD	25	I	Received Data input signal.
RxC'	23	I	Received Data Clock. This signal provides timing information to the internal shifting logic depending upon the mode of data transfer. For NRZ data, the state of the RxD pin is sampled on the High to Low clock transition.
RTS'	28	O	Request To Send signal. When LOW, RTS notifies an external interface that the 82586 has data to transmit. It is forced HIGH after a Reset and while the transmit Serial Unit is not sending data.
CTS'	29	I	Active LOW Clear To Send input enables the 82586 transmitter to actually send data. It is normally used as an interface handshake to RTS' This signal going inactive stops tranmission. It is internally synchronized. If CTS' goes inactive, meeting the setup time to TxC negative edge transmission is stopped and RTS' goes inactive within, at most two TxC cycles.
CRS'	31	I	Active LOW Carrier Sense input used to notfy the 82586 that there is traffic on the serial link. It is used only if the the 82586 is configured to external Carrier Sense. When so configured, external circuitry is required for detecting serial link traffic. It is internally synchronized. To be accepted, the signal must stay active for at least two seral clock cycles.
CDT'	30	I	Active LOW Collision Detect input is used to notify the 82586 that a collision has occurred. It is used only if the 82586 is configured tor external Collision Detect. Eternal circuitry is required for detectin the collision. It is internally synchronized. To be accepted, the signal must stay active for at least two sereial clock cycles. During transmission, the 82586 is able to recongnize a collision one bit time after proamble transmission has begun.
Int	38	O	Active HIGH interrupt request signal.
CLK	32	I	The system clock input from the 80186 or another symmetric clock generator.

Table 6.1 82586 Pin Assignments
(Reprinted by permission of Intel)

Symbol	Pin No.	Type	Function
MN/MX'	33	I	When High MN/MX' selects RD', WR', ALE, DEN', DT/R' (Minimum Mode). When LOW, MN/MX' selects A22, A23, READY, S0', S1' (Maximum Mode). Note: This pin should be static during 82586 operation.
AD0-AD15	6-11 13-22	I/O	These lines for the time multiplexed memory address (t1) and data (t2, t3, tW, t4) bus. When operating with an 8-bit bus, the high byte will output the address during the entire cycle. AD0-Ad15 are floated after a RESET or when the bus is not acquired.
A19/S6	2	O	During t1, it forms line 19 of memory address. During t2 through t4 it is used as a status indicating that this is a Master peripheralcycle, and is HIGH. Its timing is identical to that of AD0-AD15 during write operation.
HOLD	43	O	HOLD is an active HIGH signal used by the 82586 to request local bus mastership at the end of the current CPU bus transfer cycle, or at the end of the current DMA burst transfer cycle. In normal operation, HOLD goes inactive before HLDA. The 83586 can be forced off the busy HLDA going inactive, at most, three bus cycles after HLDA goes inactive.
HLDA	42	I	HLDA is an active HIGH Hold Acknowledge signal indicating that the CPU has received the HOLD request and that bus control has been relinquished to the 82586. It is internally synchronized. After HOLD is detected as LOW, the processor drives HLDA LOW. Note, Connecting VCC to HLDA IS NOT ALLOWD because it will cause a deadlock. Users wanting to give permanent bus access to the 82586 should connect HLDA with HOLD. IF HLDA goes inactive before HOLD, the 82586 will release the bus (By Hold going inactive, within three bus cycles at most.)
CA	35	I	The CA pin is a Channel Attention input used by the CPU to intitiate the 82586 execution of memory resident Command Blocks. The CA signal is synchronized internally. The signal must be HIGH for at least one system clock period. It is latched internally on HIGH to LOW edge and then detected by the 82586.
BHE	44	O	The Bus High Enable signal (BHE') is used to enable data onto the most significant half of the data bus. Its timing is identical to that of A16-A23. With a 16-bit bus it is LOW and with an 8-bit bus it is HIGH. Note: after RESET, the 82586 is configured to 8-bit bus.
READY	39	I	This active High signal is the acknowlegement from the addressed memory that the transfer cycle can be completed. While LOW, it causes wait states to be inserted. This signal must be externally synchronized with the system clock. The READY and SRDY/ARDY'.
SRDY/ARDY	37	I	This active High signal performs the same funtion as READY. If it is programmed at configure time to SRDY, it is identical to READY. If it is programmed to ARDY', the positive edge of the READY signal is internally synchronized. Note, the negative edge must still meet setup and hold time specifications, when in ARDY' mode. The ARDY signal must be active for at least one system clock HIGH period for proper strobing. The Ready signal internal to the 82586 is a logical OR between READY (In Maximum Mode only) and SRDY/ARDY'. Note that following RESET, this pin assumes ARDY' mode.

Table 6.1 82586 Pin Assignments

(Reprinted by permission of Intel)

Symbol	Pin No.	Type	Function
S0', S1'	40,41,	O	Maximum mode only. These status pins define the type of DMA transfer during the current memory cycle. They are encoded as follows S1' S0' 0 0 Not used 0 1 Read Memory 1 0 Write Memory 1 1 Passive Status is active from the middle of t4 to the end of t2. They return to the passive state during t3 or during tW when READY or ARDY is HIGH. These signals can be used by the 8288 Bus Controller to generate all memory control and timing signals. Any change from the passive state, signals the 8288 to start the next t1 to t4 bus cycle. These pins are pulled HIGH and floated after a system RESET and when the bus is not acquired.
RD'	46	O	Used in minimum mode only. The read strobe indicates that the 82586 is performing a memory read cycle. RD' is active LOW during t2, t3, and tW of any read cycle. This signal is pulled HIGH and floated after a RESET and when the bus is not acquired.
WR'	4	O	Used in minimum mode only. Address Latch Enable is provided by the 92586 is performing a write memory cycle. WR' is axctive LOW during t2, t3, and tW of any write cycle. It is pulled HIGH and floats after RESET and when the bus is not acquired.
ALE	39	O	Used in minimum mode only. Address Latch Enable is provided by the 82586 to latch the address into the 8282. 8283 address latch. It is a HIGH pulse, during t1 ('clock low') of any bus cycle. Note that ALE is never floated.
DEN'	40	O	Used in minimum mode only. Data ENable is provided as output enavbe for the 8286/8287 transceivers in a stand-alone (no 8288) system. DEN' is active LOW during each memory access. For a read cycle, it is active from the middle of t2 until the beginning of t4. For a write cycle, it is active from the beginning of t2 until the middle of t4. It is pulled HIGH and floats after a system RESET or when the bus is not acquired.
DT/R'	41	O	Used in minimum mode only. DT/R' is used in non-8288 systems using an 8286.8287 data bus tranceiver. Logically, DT/R' is equivalent to S1'. It becomes valid in the t4 preceding a bus cycle and remains valid until the final t4 of the cycle. This signal is pulled HIGH and floated after a RESET or when the bus is not acquired.

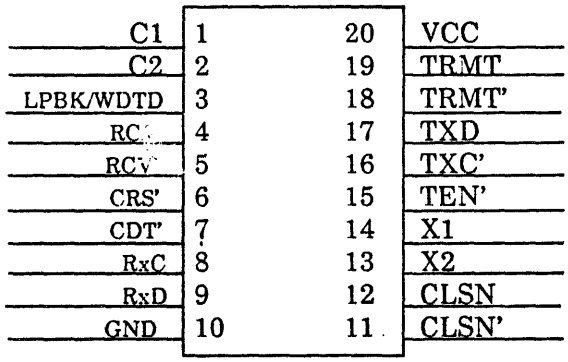


Figure 6.3 82501 Ethernet serial interface pins and signals
(Reprinted by permission of Intel Corporation)

Table 6.2 82501 Pin Assignments
(Reprinted by permission of Intel Corporation)

Symbol	Pin No.	Type	Function
TXC'	16	O	Transmit Clock: A 10 MHZ clock output woth 5 nsec rise and fall times and MOS driving levels. This clock is provided to the 82586 for serial transmission.
TEN'	15	I	Transmit Enable: An active low. TTL-Level signal synchronouts to TXC' that enables data transmission to the tranceiver cable. TEN' can be driven by RTS from the 82586.
TXD'	17	I	Transmit Data: A TTL-Level input signal that is directly connected to the serial data output, TXD, of the 82586.
RXC'	8	O	Receive Clock: An MOS-level clock output with 5 nsec rise and fall times and 50% duty cycle. This output is connected to the 82586 receive clock input RXC' There is a maximum 1.2 nsec discontinuity at the beginning of a frame reception when the phase-locked loop switches from the on-chip oscillator to the incoming data. During idle (no incoming frames) the clock frequencey will be half that of the 20 MHz crystal frequency.
CRS'	6	0	Carrier Sense: A TTL-Level, active low output to notify the 82586 that there is activity on the coaxial cable. This signal is asserted when valid data or a collision signal from the transceiver is present. It is deasserted at the end of a frame synchronous with RXC', or when the end of the collision-presence signal (CLSN and CLSN') is detected, whichever occurs later. Once deaserted CRS' wil not be reasserted again for a period of 5 nsec minimum, 7 nsec maximum, reguardless of any activity on the receive or collision-presence pairs.
RXD	9	O	Receive Data: An MOS-level output directly tied to the RXD input of the 82586 at the negative edge of RXC'. The bit stream received from the transceiver cable is Manchester decoded prior to being transferred to the controller. This output remains high during idle.
CDT'	7	O	Collision Detect: A TTL, active low signal which drives the CDT' input of the 82586 controller. It is asserted as long as there is activity on the collision-presence pair (CLSN and CLSN').
LPBK'/WDTD	3	I	Loopback: A TTL-Level control signal to enable the loopback mode. In this mode, serial data on the TXD input is routed through the 82501 internal circuits and back to the RXD output without driving the TRMT/TRMT' output pair to the tranceive cable. When LPBK is asserted, the collision circuit will also be turned on at the end of each transmission to simulate the collision test. The on-chip watchdog timer can be disabled by applying a 12V level through a 4 K ohm resistor to this pin.
TRMT /TRMT'	19-18	O	Transmit Pair: An output drive pair which generates the differential signal for the transmit pair of the Ethernet transceiver cable. Following the last transition, which is always positive at TRMT, the differential voltage is slowly reduce to zero volts. The output stream is Manchester encoded.

Table 6.2 82501 Pin Assignments
(Reprinted by permission of Intel Corporation)

Symbol	Pin No.	Type	Function
RCV/ RCV'	4-5	I	Receive Pair: A differentially driven input pair which is tied to the receive pair of the Ethernet transceiver cable. The first transition on RCV will be negative-going to indicate the beginning of a frame. The last transition should be positive going, indicating the end of a frame. The received bit stream is assumed to be Manchester encoded.
CLSN/ CLSN'	11-12	I	Collision Pair: A differentially driven input pair tied to the collision-presence pair of the Ethernet tranceiver cable. The collision-presence signal is a 10 MHz + 15 % square wave. The first transition at CLSN is negative-going to indicate the beginning of the signal; the last transition is positive-going to indicate the end of the signal.
C1-C2	1-2	I	PLL Capacitor: Phase-locked-loop capacitor inputs.
X1-X2	13-14	I	Clock Crystal: 20 MHz crystal inputs.
Vcc	20		Power: 5 + 10% volts
GND	10		Ground:Reference

6.2 Theory of Operations

The controller acts as one of the I/O subsystem bus masters. All data and control transfers are performed under DMA control. The DMA controller is implemented within the data link controller chip.

All comunication between the controller and the IOP occurs through common memory, either local IOP memory or system memory. All commands, data, and status information are communicated through memory data structures. The IOP can interrupt the ethernet controller and the ethernet controller can interrupt the IOP.

The controller thus requests the I/O bus whenever it needs information from or has information for the IOP. In addition, several diagnostics features such as error checking and loopback modes are available.

The Ethernet host address is stored in a prom in the I/O subsystem. Figures 6.4 – 6.7 are a collection of block diagrams. Text must still be provided.

6.3 Programmer Interface (TO BE PROVIDED)

7 Floppy Disk Subsystem-Preliminary

The floppy disk subsystem provides low cost, high density, removable storage media for the Dove workstation. Figure 7.1 illustrates the floppy disk subsystem as it applies to the overall IOP board.

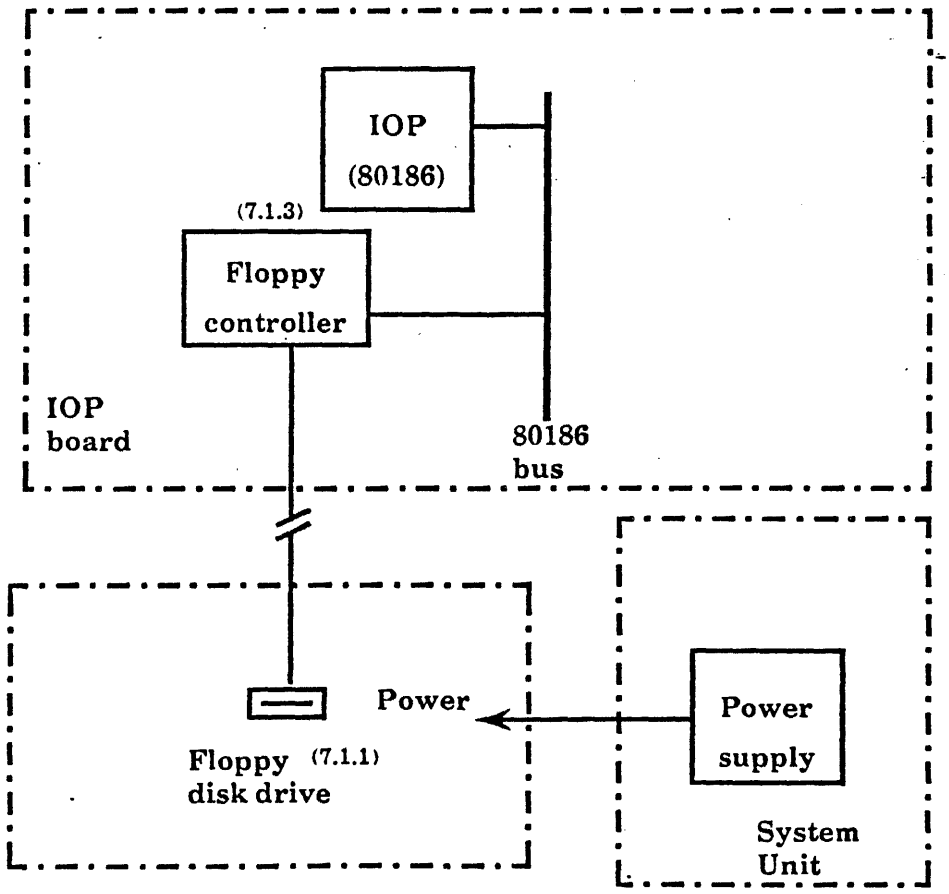


Figure 7.1. Floppy disk subsystem block diagram

7.1 Hardware

The floppy disk subsystem consists of the following components:

- Floppy disk drive (Section 7.1.1)
- Floppy diskette (Section 7.1.2)
- Floppy disk controller (Section 7.1.3)

Together, these components constitute the floppy disk subsystem. The hardware is discussed in the following subsections.

7.1.1 Floppy Disk Drives

The floppy disk drive is a half-height device. The interface to the drive is compatible with the Shugart SA450 Standard Interface.

The floppy disk subsystem supports up to four 5¼" double-sided or single/double density floppy disk drives. The drives (?) have a 250 K bit/second or 500K bit/second transfer rate.

7.1.2 Diskettes

Figure 7.2 illustrates the diskette used with the disk drive. The diskette consists of a flexible magnetic disk enclosed in a protective jacket. The jacket cleans the diskette during normal operation or rotation (300/360 rpm). An opening in the jacket provides read/write/erase head access.

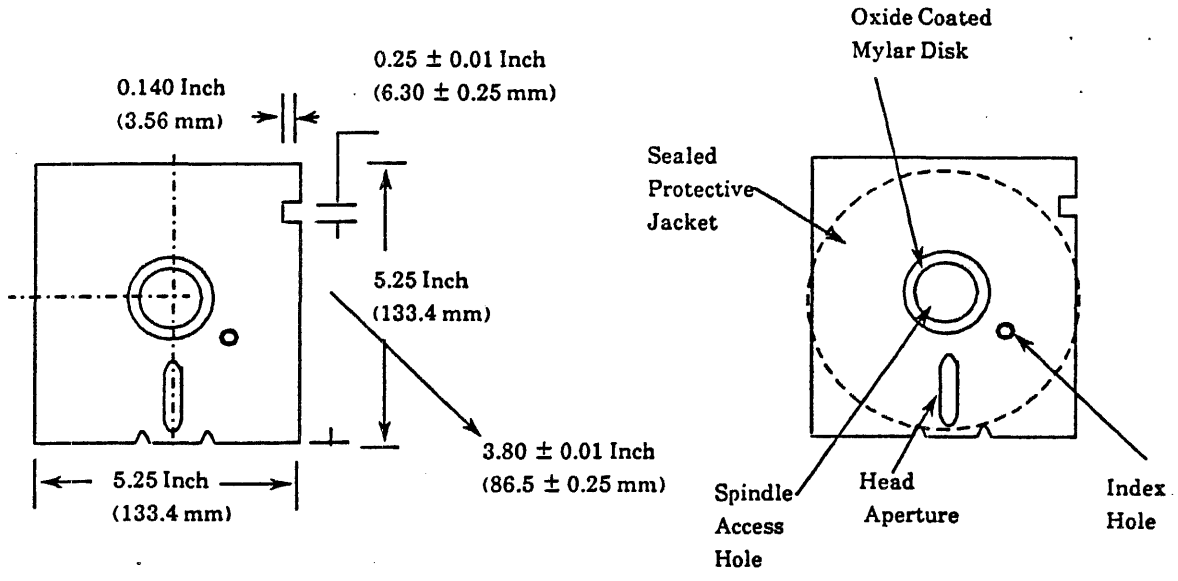


Figure 7.2. Diskettes

Two types of diskettes are supported: 48 TPI drives (IBM PC compatible) and 96 TPI drives (not IBM PC compatible). Table 7.1 summarizes the characteristics of both diskette types. Diskette format is discussed in section 7.3.4.

Table 7.1. Characteristics of Formatted Diskettes

Characteristic	48 TPI SA455	96 TPI SA465	96 TPI SA475
Diskette Size	5¼ Inches	5¼ Inches	5¼ Inches
Sides	2	2	2
Tracks/Side	40	80	80
Sectors/Track	9	9	15
Bytes/Sector	512	512	512
Bytes/Track	4608	4608	7680
Bytes/Side	184.32K	368.64K	614.4K
Bytes/Diskette	368.64K	737.28K	1.2288M
Rotation Speed	300 RPM	300 RPM	360 RPM
Transfer Rate	250K bit/sec	250K bit/sec	500K bit/sec
Recording Method	MFM	MFM	MFM

7.1.3 Floppy Disk Controller

The Floppy Disk Controller (FDC) consists of an Intel 8272 chip and a Standard Microsystems 9229 chip. The 8272 chip serves as the IOP interface to the floppy disk; the 9229 chip acts as a data separator.

Floppy disk schematics are given in Appendix G1.

7.1.3.1 Floppy Disk Controller

The Intel 8272 floppy disk controller consists of two chips: the controller itself and the controller interface to the data separator and floppy disk drive.

The floppy disk controller interface to the IOP is a standard microprocessor interface; that is, it has a data bus, chip select, write strobe, and read strobe.

Figure 7.3 illustrates the pins and signals of the 8272 floppy disk controller. Table 7.2 lists the pins by name and number and explains their function in the controller system.

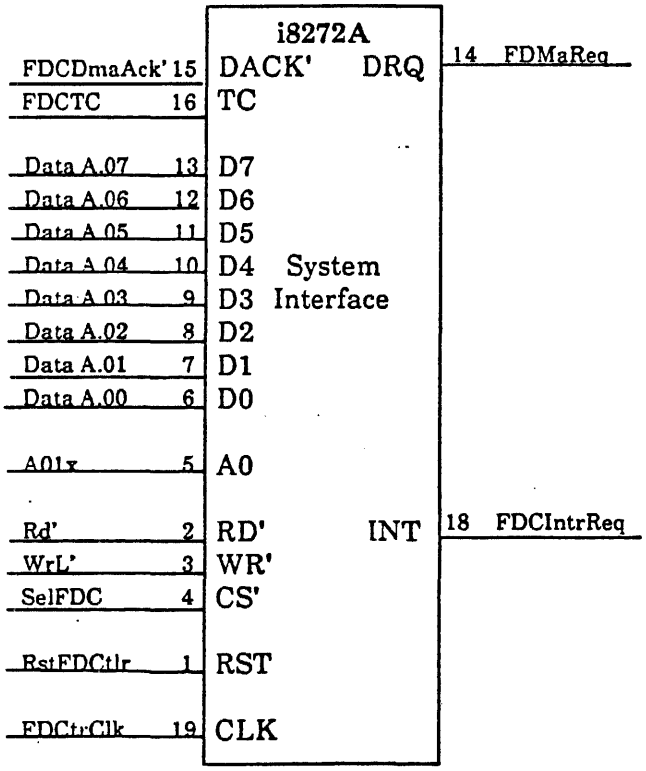


Figure 7.3. 8272 floppy disk controller pins and signals

Note: The signal FDMaReq goes through logic that delays the DMA request by approximately 1 µs.

Table 7.2. 8272 Pin Assignments
(reprinted by permission of Intel Corporation)

Symbol	Pin No.	Type	Connection To	Name and Function
A0	5	Input	µP	**Data Status Register Select** Selects Data Reg (A0 = 1) or Status Reg (A0 = 0) contents to be sent to Data Bus.
CLK	19	Input		**Clock** Single phase 8 MHz Squarewave clock.
CS	4	Input	µP	**Chip Select** IC selected when "0" (low) allowing RD' and WR' to be enabled.
D0-D7	6-13	I/O	µP	**Data Bus** Bidirectional 8-Bit Data Bus.
DACK'	15	Input	DMA	**DMA Acknowledge** The DMA cycle is active when "0" (low) and controller is performing DMA transfer.
DRQ	14	Output	DMA	**Data DMA Request** DMA request is being made by FDC when $\overline{DRQ}$ "1."
INT	18	Output	µP	**Interrupt** Interrupt request generated by FDC.
RD'	2	Input	µP	**Read Control** Signal for transfer of data from FDC to Data Bus when "0" (low).
RST	1	Input	µP	**Reset** Places FDC in idle state. Resets output lines to FDC to "0"(low). This does not clear the last specify command.
TC	16	Input	DMA	**Terminal Count** This pin indicates the termination of a DMA transfer when "1" (high).
WR'	3	Input	µP	**Write Control signal** for transfer of data to FDC via Data Bus when "0" (low).

The 8272 also has a floppy disk drive interface connecting the 9229 and the floppy disk drive. The drive interface is limited to a 10-foot maximum cable length. Figure 7.4 illustrates the pins and signals of the interface. Table 7.3 lists the pins by name and number and explains their function.

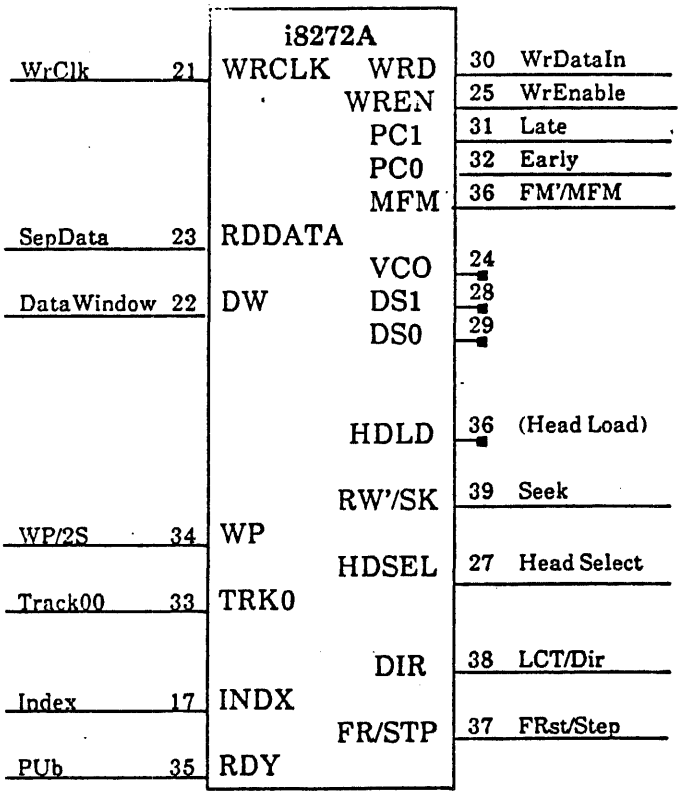


Figure 7.4. 8272 floppy disk controller interface pins and signals

Table 7.3. 8272 Interface Pin Assignments
(reprinted by permission of Intel Corporation)

Symbol	Pin No.	Type	Connection To	Name and Function
DW	22	Input	PLL	**Data Window** Data sample signal from the phase-locked loop indicating that the FDC should sample input data from the disk drive.
DS_1, DS_0	28, 29	Output	Drive	**Drive Select** Selects the disk drives.
FLT/TRK0	33	Input	Drive	**Fault/Track 0** Senses the disk drive fault condition in the Read/Write mode and the Track 0 condition in the seek mode.
FR/STP	37	Output	Drive	**Fault Reset/Step** Resets the fault flip-flop in the disk drive when operating in the Read/Write mode. Provides head step pulses (to move the head from one cylinder to another cylinder) in the Seek mode.
GND	20			**Ground** DC power return.
HDLD	36	Output	Drive	**Head Load** Loads the disk drive read/write head. (The head is placed in contact with the disk.)
HDSEL	27	Output	Drive	**Head Select** Selects head 0 or head 1 on a dual sided disk.
LCT/DIR	38	Input	Drive	**Low Current/Direction** Signals that the recording head has been positioned over the inner cylinders (44-47) of the floppy disk in the Read/Write mode. (The write current must be lowered when recording on the physically shorter inner cylinders of the disk. Most drives do not track the actual head position and require that the FDC supply this signal. Determines the head step direction in the Seek mode. In the Seek mode, a high level on this pin steps the read/write head toward the spindle (step-in). A low level steps the head away from the spindle (step-out).
MFM	26	Output	PLL	**MFM Mode** Active-high output used by external logic to enable MFM double -density recording mode. When the MFM output is low, single-density FM recording is indicated.
PC_1, PC_0	31, 32	Output	Drive	**Precompensation** (pre-shift) **Control.** Write precompensation output control during MFM mode. Specifies early, late, and normal timing signals.
RDDATA	23	Input	Drive	**Read Data** FDC input data from the selected disk drive.
RDY	35	Input	Drive	**Ready** Senses the disk drive ready status.
RW/SEEK	39	Input	Drive	**Read, Write/Seek Mode Selector** A high level selects the Seek mode; a low level selects the Read/Write mode.
V_{cc}	40			+5V DC power.

Table 7.3. 8272 Interface Pin Assignments
(reprinted by permission of Intel Corporation)

Symbol	Pin No.	Type	Connection To	Name and Function
VCO	24	Output	PLL	**VCO Sync** Active-high signal indicating an 8272 interrupt service request.
WREN	25	Output	PLL	**Write Enable** Active-High output that enables the disk drive write gate.
WP/TS	34	Input	Drive	**Write Protect/Two Sided** Senses the disk write protect status in the Read/Write mode and the dual-sided media status in the Seek mode.
WRCLK	21	Input		**Write Clock** 500 kHz (FM) or 1 MHz (FM) clock with a constant pulse width of 250 ns (for both FM and MFM recording). The write clock must be present at all times.
WRD	30	Output	Drive	**Write Data** Serial data stream (combinations of clock and data bits) to be written on the disk.

7.1.3.2 Data Separator

Figure 7.5 illustrates the pins and signals of the 18-pin 9229 data separator. Table 7.3 lists the pins by name and number and explains their function.

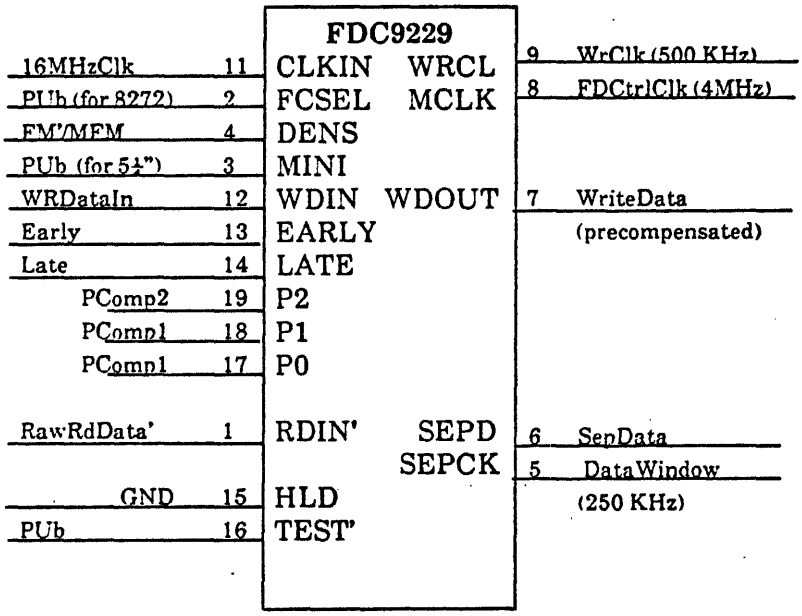


Figure 7.5. 9229 data separator pins and signals

Table 7.4. 9229 Pin Assignments

Symbol	Pin No.	Type	Connection To	Name and Function
CLKIN	11	Input	16 MHz Clock	**Clock In** Connects to a 16 MHz crystal.
DENS	4	Input	FDC	**Density** Indicates configuration for double density (MFM) floppy disk drive interfaces.
EARLY	13	Input	FDC	**Early Precompensation** When high, writes WriteData pulse to the floppy disk.
FCSEL	2	Input	Pull-Up	**Floppy Controller Select** Set high to indicate that data separator is programmed for 8272 controller.
HLD	15	Input	N/C	Not used.
LATE	14	Input	FDC	**Late Precompensation** When high, the current WriteData pulse is written late to the disk. When both Early and Late are low, the current WriteData pulse is written early to the floppy disk.
MCLK	8	Output	FDC	**FDC Clock** The 4 MHz master clock to the floppy disk controller.
MINI	3	Input	Control Register	**8"/5¼ Select** Indicates 5¼" FDC interface.
P0-P2	17-19	Input	Control Register	**Precompensation Value** Ses 250 ns as amount of precompensation applied to the write data.
RDIN'	1	Input	Drive	**Read Data In** (active low) Receives raw read data from the floppy disk drive.
SEPCK	5	Output	FDC	**Separated Clock** Outputs a 250 KHz square wave window clock signal.
SEPD	6	Output	FDC	**Separated Data** Regenerated data pulse derived from the raw data input.
TEST'	16	Input	Pull-Up	**Test Mode** Tied high for normal operation.
WRCLK	9	Output	FDC	**Write Clock** Writes 500 KHz clock with 250 ns pulse width to the floppy disk controller.
WDOUT	7	Output	Drive	**Write Data Out** Generates precompensated WriteData stream to the floppy disk drive.
WDIN	12	Input	FDC	**Write Data In** Receives write data stream from the floppy disk controller.

7.2 Theory of Operations

The floppy disk controller (FDC) converts high level disk commands to the appropriate sequence of disk drive control signals.

Figure 7.6 illustrates the data paths of the floppy disk subsystem. The number in parentheses is the number of the subsection that discusses the component.

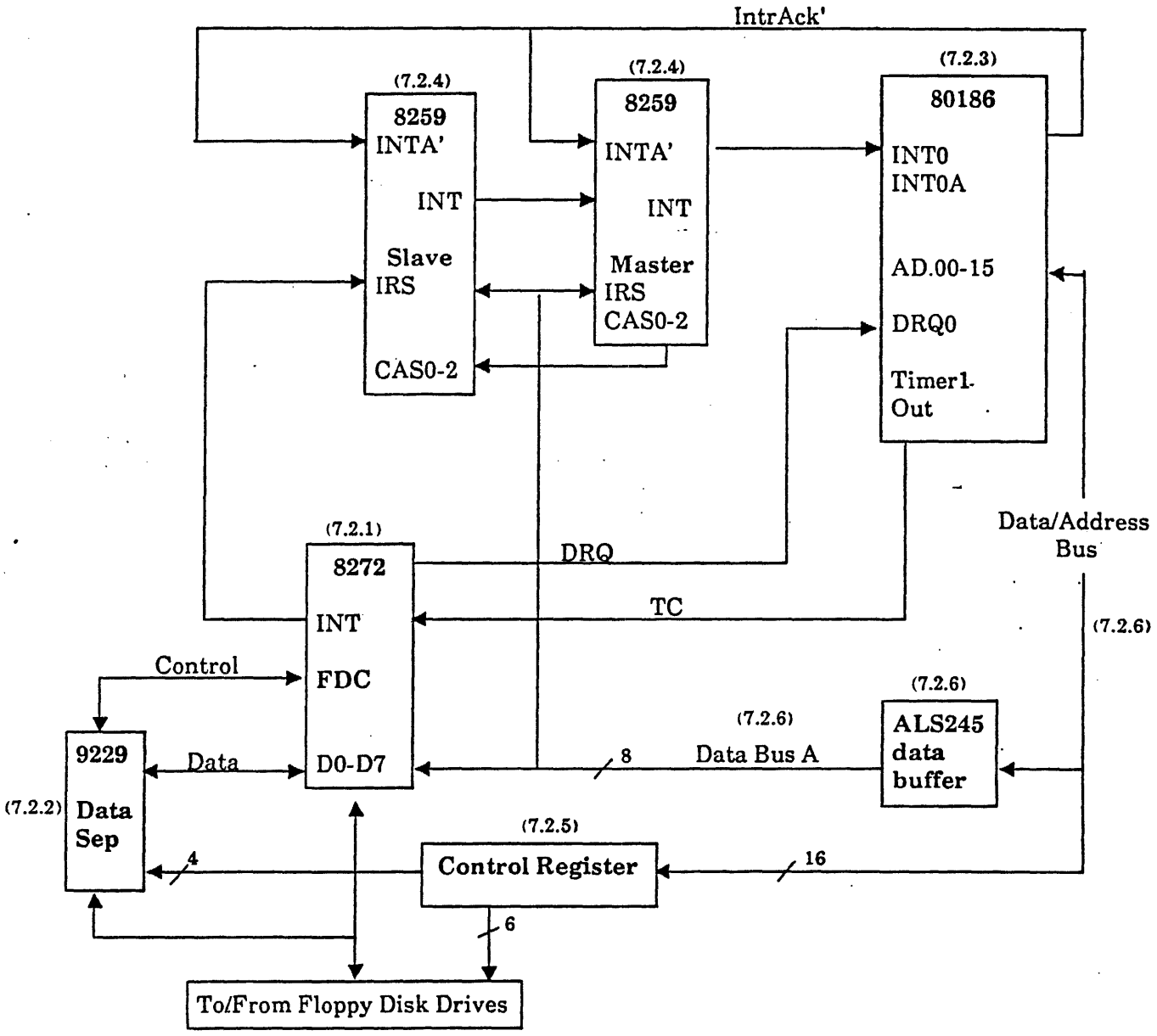


Figure 7.6. System interface

7.2.1 Floppy Disk Controller

The floppy disk controller translates IOP commands, such as format, read, and write, into the required control signals and sequences for the floppy disk. The controller is instructed by the IOP where, what, and how much to read.

Requests are exchanged by way of a DMA circuit consisting of a DMA request line, a terminal count line, and a DMA acknowledge line. The DMA request line is active only during data transfer.

When the floppy disk controller needs a byte for a write operation or holds a byte from a read operation, it activates the DMA request line. This signal (DRQ) notifies the 80186 that a DMA transfer is required.

When the DMA reads or writes the data byte to the floppy disk controller, the DMA acknowledge line is activated by the DMA. (Where is the acknowledge line on the figure?) The floppy disk controller deactivates the FDC request line approximately 200 ns after the acknowledge signal is received. At the end of the transfer, the DMA deactivates the DMA acknowledge line.

7.2.2 Data Separator

The data separator helps complete data separation for the floppy disk drives. Incoming read data is separated into a data signal and a data window signal, as required by the 8272 for reading data.

The data separator also adjusts precompensation on the write data stream, under the control of the floppy disk controller. (as directed by the control register?)

7.2.3 80186 Processor (DMA and Timer)

The 80186 processor has two DMA channels: a DMA request and a DMA controller. When the floppy disk controller activates the DMA request line connected to the DMA channel 0 request line, the request goes directly to the 80186 DMA controller. The DMA controller then determines how many bytes to transfer, and sets up address pointers to store the data.

The DMA controller does not notify the floppy disk controller that the last byte has been transferred. Instead, the DMA request line from the floppy disk controller is also connected to the 80186 timer. The timer in the 80186 has an input line assigned to monitor the DMA, and has a timer-in and timer-out pin. At each DMA request, the timer's count is incremented by 1. When the counter reaches a value preset in the max count register A, the timer outputs a pulse on the timer-out pin, which is connected to the floppy disk controller terminal-count pin. This output (TC) signals the floppy disk controller to end requests for further DMA cycles. TC is issued to the FDC interface approximately 18 µs after the last DmaAck signal.

7.2.4 Interrupt Controllers

The IOP contains three interrupt controllers: a slave interrupt controller, a master interrupt controller, and the 80186 interrupt controller. When an interrupt occurs, the floppy disk controller sends an interrupt signal to the slave interrupt controller; slave interrupts are then funneled into one interrupt line, which becomes one input to the 8259 master. All master interrupts are finally funneled into one interrupt pin on the 80186.

7.2.5 Control Register

Twlve bits in the IOP control register are used in the floppy disk subsystem. Seven bits control floppy disk drive signals; that is, drive motor function, the "in-use" line on the drive interface, and drive selection. Four bits control the data rate expected by the data separator and data separator precompensation. One bit (not indicated on the figure) enables timer 1 for TC signal generation.

7.2.6 Buses

Commands are written to the 8272 via the Data Bus A, which receives data through a data buffer from a data/address bus.

7.3 Programmer Interface

This section describes the floppy disk controller registers and associated registers, interrupts, and reset. Diskette format is described at the end of the section.

7.3.1 Registers

The floppy disk controller holds a status register and a data register. The status register provides information regarding the floppy controllers current status. The data register is the location for reading data, I/O result status, or writing data and commands.

The DMA data register accesses the floppy disk controller data register via the DMA channel. Timer registers hold preset count data and record DMA accesses. Finally, eleven bits on the IOP control register affect the floppy disk controller. The registers are discussed in the following subsections.

Table 7.5 lists the floppy disk controller registers and addresses. The registers listed in brackets are external to the floppy disk controller.

Table 7.5. Registers and Addresses

Register Name	Address
Status	50H
Data	52H
[DMA Data]	54H
[IOP Control]	[80H]

7.3.1.1 FDC Registers

The floppy disk controller has a status register and a data register, selected by signal A0. A0 high designates a data register; A0 low designates a status register.

Status register

The floppy disk controller sets the status register bits prior to the command and result phase. Figure 7.7 illustrates the floppy disk controller status register. Each address is described below; the bit name (?) is given in parenthesis after each bit.

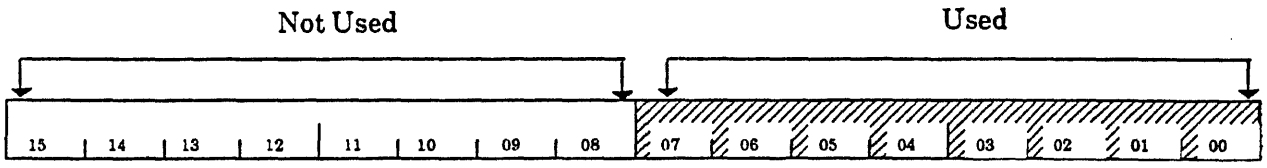


Figure 7.7. FDC Status Register (Read only I/O Addr = 50 hex)

Bit 0 (D0B) When set, indicates Drive 0 is doing a seek.

Bit 1 (D1B) When set, indicates Drive 1 is doing a seek.

Bit 2 (D2B) When set, indicates Drive 2 is doing a seek.

Bit 3 (D3B) When set, indicates Drive 3 is doing a seek.

Bit 4 (CB) When set, indicates that the floppy disk controller is doing a read or write operation for the the floppy disk.

Bit 5 (NDM) When set, indicates that the floppy disk controller is in non-DMA mode. This bit is only valid during the execution phase of an operation.

Bit 6 (DIO) Indicates the direction of the next access of the data register, as follows:
0 = data is written to the data register. When a command byte is issued, the DIO bit is 0, indicating that a transfer occurred from the 80186 to the floppy disk controller data register.
1 = data is read from the data register. When a status byte is read, the DIO bit is 1, indicating that a transfer occurred from the floppy disk controller to the 80186.
Note: In both cases, bit 7 (RQM) is 1.

Bit7 (RQM) When set, indicates that the floppy disk controller is ready to accept a byte from the 80186 or is ready to have a byte read by the 80186.

The status register may be read at any time. The operations necessary to obtain status are done by the floppy disk controller asynchronously to any processor functions. Therefore, the main status register should be checked prior to issuing any command byte or obtaining any result bytes for an I/O. Bits 6 and 7 (DIO and RQM) should also be checked to ensure that the floppy disk controller will move data in the correct direction.

If multiple reads or writes to the floppy disk controller are done in rapid succession, a delay of at least 12 microseconds is necessary from the end of a command or result byte

access. This delay allows the floppy disk controller to change the RQM bit from 1 to 0. The status register should not be read during this time because the RQM bit will erroneously indicate that the floppy disk controller is ready. After this delay, the software can continue to read the status register for the next access.

Data register

The 8-bit data register is the location for reading or writing commands, status and data.

The floppy disk controller executes the following commands through this register:

Read: data, deleted data, a track, ID
Write: data, deleted data
Format a track
Scan: equal, low or equal, high or equal
Recalibrate
Sense: interrupt status, drive status
Specify
Seek
Invalid

Each command involves: 1) multi-byte transfers from the 80186 to the floppy disk controller for command information and 2) multi-byte transfers from the floppy disk controller to the 80186 for result information. Commands can be considered as having three phases: command, execution, and result.

Each command can generally be considered as having three phases: command, execution, and result. However, not all operations have all three phases. For example, commands such as seek and recalibrate have only command and execution phases. Instead of a result phase, a sense interrupt status command must be issued to the floppy disk controller.

The execution phase always begins after the floppy disk controller receives the last command byte. The floppy disk controller then executes the requested command. When execution is completed, the floppy disk controller interrupts the 80186 to indicate the beginning of the result phase.

Appendix G2 provides the command instruction sets and timing for the command phases.

7.3.1.2 DMA Registers

The 80186 integrated DMA controller handles all data transfers for read, write, or format operations. Channel 0 of the 80186 DMA controller is dedicated to the floppy disk subsystem.

Table 7.6 lists the name and address of the DMA registers that affect the floppy disk subsystem .Each register is word length and is located in the 80186 processor. The function and use of each register is discussed in Section 2, titled "IOP Processor."

The DMA registers must be set up and enabled before command bytes are issued to the floppy disk controller. This initialization readies the DMA for the execution phase.

When the floppy disk controller begins a read or write operation, it requests DMA cycles until it reaches the end of the track or the cylinder. However, if the terminal count (TC) signal becomes active, the active signal is sent to the floppy disk controller to indicate that

Table 7.6. DMA Registers

Register Name	Register Address
Control Word	$FFCAH
Transfer Count	$FFC8H
Destination Pointer (Upper 4 bits)	$FFC6H
Destination Pointer	$FFC4H
Source Pointer (Upper 4 bits)	$FFC2H
Source Pointer	$FFC0H

the required amount of data has been transferred. The floppy disk controller then halts activity.

Note: The 80186 DMA controller does not supply a TC signal. The DMA acknowledge line is connected to an 80186 integrated timer as well as to the DMA controller. When the requested number of DMA cycles have been completed, the 80186 timer 1 counts DMA cycles and provides a signal on the TC signal line.

7.3.1.3 Timer Registers

Timer 1 in the 80186 counts DMA cycles and provides the TC signal. Timer 1 must be initialized prior to starting the DMA or floppy disk controller.

The timer output line must also be enabled onto the floppy disk controller TC line. Enabling is done by setting bit 11 in the IOP (?) control register to 1.

Maximum count register A is preprogrammed for the value that will stop DMA requests.

Table 7.7 lists the registers and addresses of the timer. The function and use of each register is discussed in Section 2, titled "IOP Processor."

Table 7.7. Timer Registers

Register Name	Register Address
Count Register	$FF58H
Max Count Register A	$FF5AH
Max Count Register B	$FF5CH
Control Register	$FF5EH

7.3.1.4 IOP Control Register

Several bits in the IOP Control Register (80H) must be set or reset prior to floppy disk operations. Since this write-only register may not be read, an image of what was written into the register should be maintained in memory.

Figure 7.8 illustrates the IOP control register as applied to floppy disk functions. The register bits are described below.

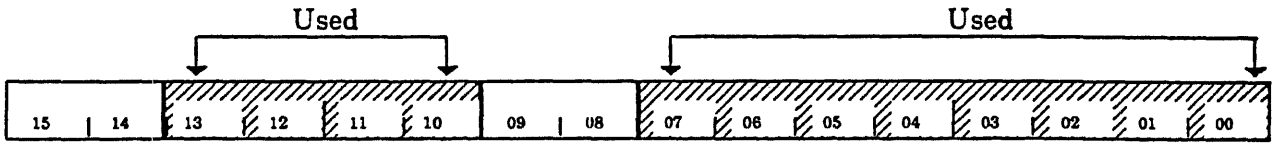


Figure 7.8. IOP control register (I/O write-only address = 80 hex)

Bit 13 - FddMotorOn - Turns on the floppy disk drive (FDD) drive motor for all drives, as follows:
0 = FDD drive motors for all drives are shut off
1 = FDD drive motors for all drives are turned on

A 500 ms delay must be allowed before reading and writing.

For maximum motor life, deactivate this line if immediate disk activity is required.

Bit 12 – FddInUse – Controls the "in use" interface line, as follows:
0 = in use to 1
1 = in use to 0

Bit 11 – AllowTmrTC – Enables timer 1 output so that it may be used to generate the TC signal to the floppy disk controller.

Bit 10 – High/Low' – Controls the speed of the floppy disk drive (for SA475 high-speed drives only), as follows:
0 = high speed (360 RPM)
1 = low speed (300 RPM)

Bits 7-4 – DriveSel4 - DriveSel1

– Control the drive select lines for the four possible drives. When a driveSel bit is set to 1, the respective drive will be selected. These lines should be set prior to any disk operation, and should be reset after all disk activity is completed. Software should ensure that only one drive select is set active at any given time. More than one active drive select line will cause a conflict on the drive interface.

Bit 3 – 5H/8L – Controls the data rate that the data separator will use on data transfers, as follows:
0 = 500 kkb/s (SA475 at high speed only)
1 = 250 kb/s

Bits 2-0 – Precomp 2-0 – Control the amount of write precompensation that the data separator will use during write operations. Precomp2 is the most significant bit; Precomp0 is the least significant bit.

At high speed, precompensation can be set in increments of 62.5 ns. At low speed, precompensation can be set in increments of 125 ns.

For SA455, SA465, and SA475 (at low speed), precompensation should be set at 250 ns; that is, Precomp2-0 set to 0,1,0, respectively.

For SA475 at high speed, precompensation should be set to 125 ns; that is, Precomp2-0 set to 0,1, respectively.

7.3.2 Interrupts

Interrupts from the floppy disk controller are generated by: 1) completion of a read, write command at the begining of the result phase; 2) the end of an asynchronous command (seek, recalibrate, etc); or 3) an abnormal termination.

The two interrupt causes are differentiated by the floppy disk controller busy bit in the status register. If the bit is 0, then the interrupt was caused by an asychronous event. If the bit is 1, then the interrupt was caused by the result phase of a read or write command. In either case, the 80186 continues issuing Sense Interrupt commands after the interrupt is serviced in order to search for any "hidden" interrupts.

The floppy disk controller is capable of stacking up several interrupts internally, thus requiring the repeated Sense Interrupt commands. The sense interrupt command should be repeated until invalid command code (ST0 = 80H) is received. This command code indicates that all interrupts have been serviced.

7.3.3 Reset

The reset signal for the floppy disk subsystem is connected to bit 2 of the reset control register (C0H). This bit is cleared to activate the reset line. The reset line in turn is held active for at least 4 microseconds.

After Reset, the floppy disk controller updates the drive status for all four possible drives and asserts the interrupt line. The reset routine issues Sense Interrupt commands until the floppy disk controller indicates that all status addresses have been read (ST0 = 80H). The routine then initializes the floppy disk controller with the Specify command.

7.3.4. Diskette Format

New floppy diskettes must be written or formatted by the controller with a fixed data pattern or format before any data can be stored on them. The controller does the formatting track by track. Formatting destroys any information that was previously on the diskette.

Figure 7.9 illustrates the floppy diskette format, and subsections following the figure describe the format.

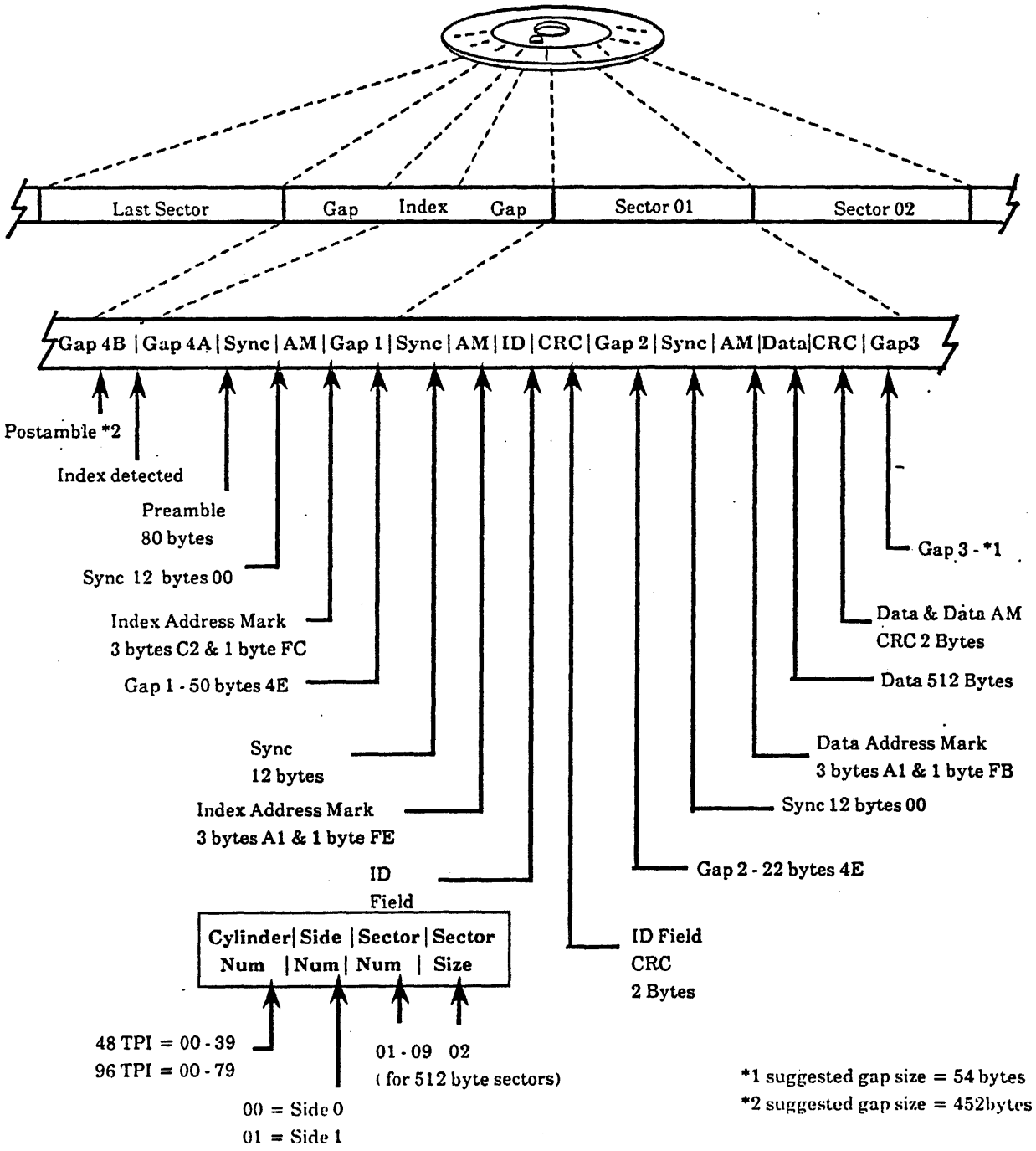


Figure 7.9. Floppy disk format

7.3.4.1 Preambles and Postambles

The sections of the diskette immediately before and after index hole detection are called the postamble and preamble, respectively. These diskette regions are written only at format time.

Preambles

The preamble gap begins at the index hole and consists of 80 bytes of 4EH. The gap is followed by 12 bytes of 00H that are collectively called the sync field. Data recovery circuits use this field to synchronize themselves for the upcoming address mark information.

The address mark immediately follows the sync field. Composed of three bytes of C2H and 1 byte of FCH, this address mark indicates the beginning of a data track. The address mark is followed by 50 bytes of 4EH that generate another gap.

Postambles

The postamble immediately follows the last sector, extends to the point where the index hole is first detected, and is filled with a 4EH pattern.

7.3.4.2 Sectors

The diskette has nine sectors. Each sector has the following ten divisions:

1) Each sector begins with twelve bytes of 00H as a sync field.

2) The sync field is followed by a 4 byte ID address mark that indicates the beginning of the sector ID information. The ID address mark is composed of three bytes of A1H and 1 byte of FEH. The address mark is followed by the ID field for that particular sector..

3) The ID field contains four bytes. The first byte indicates the cylinder on which the sector resides. The second byte indicates the side of the diskette on which the sector resides. The third byte indicates the sector number: 01H for sector 1, 02H for sector 2, and so on. The last byte indicates the number of data bytes in the sector. This byte is set to 02H to indicate 512 bytes per sector.

4) The ID field is followed by a two byte CRC field. These two bytes are the accumulated CRC over the ID field.

5) The ID field CRC is followed by a gap of 22 bytes of 4EH. The gap separates the ID information from the data information.

6) The sync field consists of 12 bytes of 00H. The field is used by phase locked loop to synchronize itself for the upcoming data address mark information.

7) The address mark immediately follows the sync field and is composed of three bytes of A1H and 1 byte of FBH. The mark is used to indicate the beginning of the data section.

8) The next section is the actual data section.

7.3.4. Diskette Format

New floppy diskettes must be written or formatted by the controller with a fixed data pattern or format before any data can be stored on them. The controller does the formatting track by track. Formatting destroys any information that was previously on the diskette.

Figure 7.9 illustrates the floppy diskette format, and subsections following the figure describe the format.

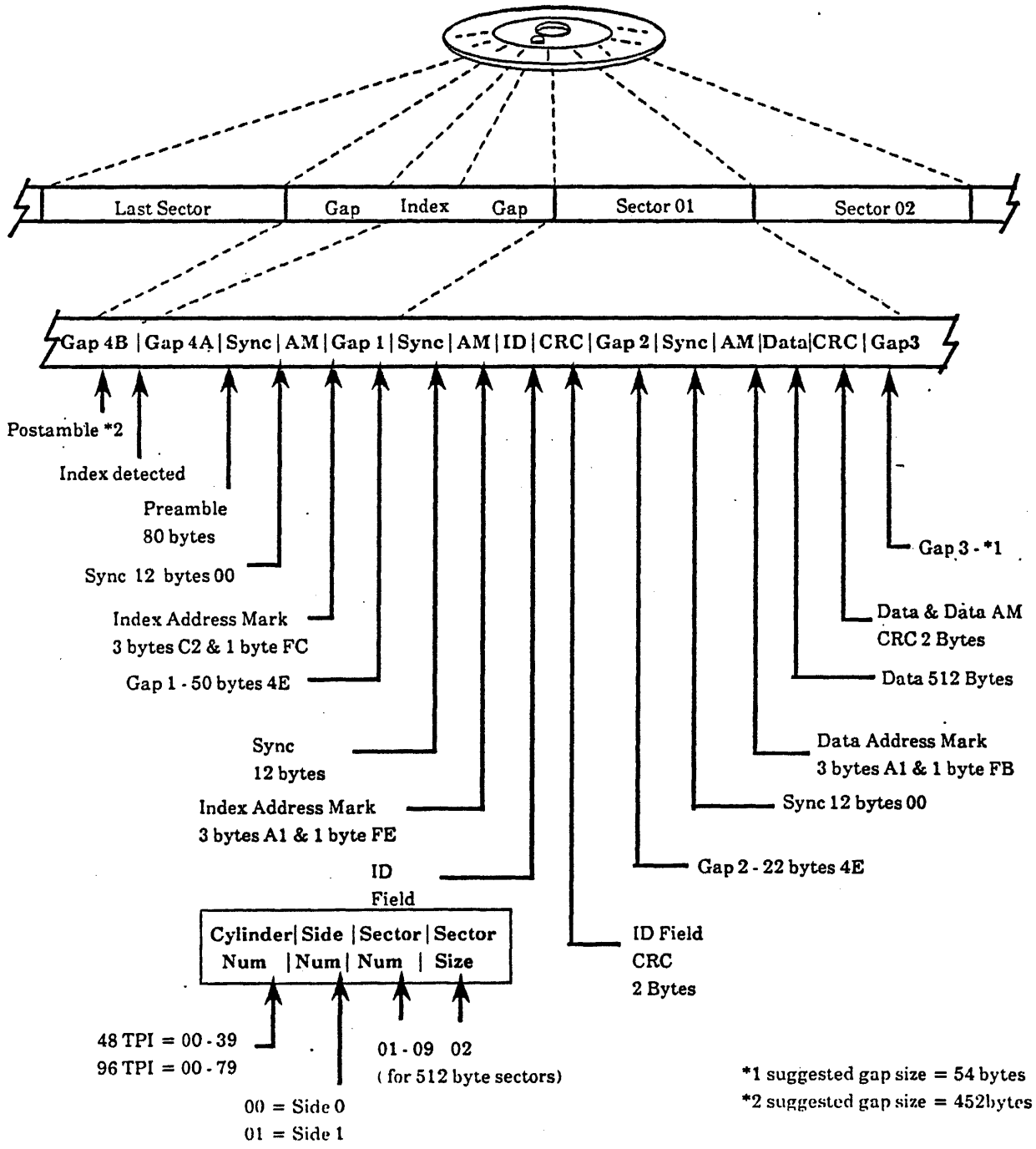


Figure 7.9. Floppy disk format

7.3.4.1 Preambles and Postambles

The sections of the diskette immediately before and after index hole detection are called the postamble and preamble, respectively. These diskette regions are written only at format time.

Preambles

The preamble gap begins at the index hole and consists of 80 bytes of 4EH. The gap is followed by 12 bytes of 00H that are collectively called the sync field. Data recovery circuits use this field to synchronize themselves for the upcoming address mark information.

The address mark immediately follows the sync field. Composed of three bytes of C2H and 1 byte of FCH, this address mark indicates the beginning of a data track. The address mark is followed by 50 bytes of 4EH that generate another gap.

Postambles

The postamble immediately follows the last sector, extends to the point where the index hole is first detected, and is filled with a 4EH pattern.

7.3.4.2 Sectors

The diskette has nine sectors. Each sector has the following ten divisions:

1) Each sector begins with twelve bytes of 00H as a sync field.

2) The sync field is followed by a 4 byte ID address mark that indicates the beginning of the sector ID information. The ID address mark is composed of three bytes of A1H and 1 byte of FEH. The address mark is followed by the ID field for that particular sector..

3) The ID field contains four bytes. The first byte indicates the cylinder on which the sector resides. The second byte indicates the side of the diskette on which the sector resides. The third byte indicates the sector number: 01H for sector 1, 02H for sector 2, and so on. The last byte indicates the number of data bytes in the sector. This byte is set to 02H to indicate 512 bytes per sector.

4) The ID field is followed by a two byte CRC field. These two bytes are the accumulated CRC over the ID field.

5) The ID field CRC is followed by a gap of 22 bytes of 4EH. The gap separates the ID information from the data information.

6) The sync field consists of 12 bytes of 00H. The field is used by phase locked loop to synchronize itself for the upcoming data address mark information.

7) The address mark immediately follows the sync field and is composed of three bytes of A1H and 1 byte of FBH. The mark is used to indicate the beginning of the data section.

8) The next section is the actual data section.

9) A two byte CRC follows. The CRC is accumulated over the data field and the data address mark.

10) Another gap follows the CRC field. This gap is 54 bytes of 4EH at format time, and separates the end of the current sector from the beginning of the next sector.

8 RS-232 Controller

The RS-232 controller provides a channel for communication between the Dove machine and external devices like print servers or printers. Figure 8.1 illustrates the RS-232 controller as it applies to the overall IOP board.

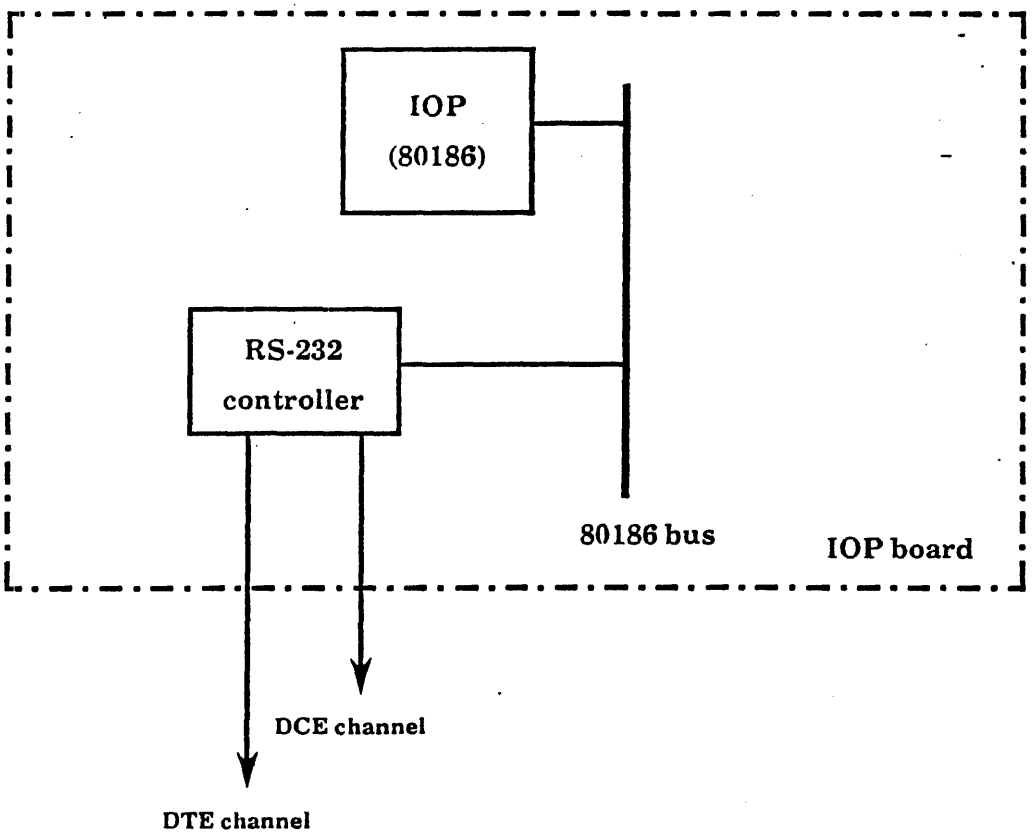


Figure 8.1. RS-232-C block diagram

8.1 Hardware

The RS-232 controller consists of the following components:

- Serial controller
- System interface
- Two serial channels

8.1.1 Serial Controller

The RS-232 controller consists of an integrated multi-protocol Intel 8274 serial controller and an Intel 8254 timer. The 8274 chip supports RS-232 requirements directly and provides parity and CRC generation and checking. Two bytes of buffering are provided in the controller. The 8254 chip provides the timing function for the system.

8.1.1.1 Controller

Figure 8.2 illustrates the pins and signals of the serial controller. Table 8.1 lists the pins by name and number and explains their function in the controller system.

CLK	1	40	Vcc
RESET'	2	39	CTSA'
CDA	3	38	RTSA'
RxCB'	4	37	TXDA'
CDB'	5	36	TXCA'
CTSB'	6	35	RxCB'
TxCB'	7	34	RxDB'
TxDB'	8	33	SYNDETA'
RxDB'	9	32	RDYA/TxDRQA
SYNDETB/RTSB	10	31	DTRA'
RDYB/TxDRQA	11	30	IPQ/TxDRQB
DB7	12	29	IPI/RxDRQB
DB6	13	28	INT'
DB5	14	27	INTA'
DB4	15	26	DTRB'
DB3	16	25	A0
DB2	17	24	A1
DB1	18	23	CS'
DB0	19	22	RD'
GND	20	21	WR'

Figure 8.2. 8274 serial controller pins and signals

Table 8.1. 8274 Serial Controller Pin Assignments

Symbol	Pin #	Type	Name and Function
CLK	1	I	Clock: System clock, TTL compatible
RESET'	2	I	Reset: A l w signal on this pin forces the MPSC to an idle state. TxDx and TxD8 are forced hig.. The modern interface output signals are forced high. The MPSC will remain idle until the control registers are initialized. Reset must be true for one complete CLK cycle.
CDA	3	I	Carrier Detect (Channel A): This interface signal is supplied by the modem to indicate that a data carrier signal has been detected and that a valid data signal is present on the RxDA line. If the auto enable control is set the 8274 will not enable the serial receiver until CDA' has been activated
RxCB'	4	I	Receive Clock (Channel B): The serial data are shifted into the receive data input (RxDB') on the rising edge of the receive clock.
CDB'	5	I	Carrier Detect (Channel B): This interface signal is supplied by the modem to indicate that a data carrier signal has been detected and that a valid data signal is present on the RxDa line. If the auto enable control is set the 8274 will not enable the serial receiver until CDB' has been activated.
CTSB'	6	I	Clear to Send (Channel B): This interface signal is supplied by the modem to indicate that a data carrier signal has been detected and that a valid data signal is present on the RXDB' line . If the auto enable control is set the 8274 will not enable the serial receiver until CDB' has been activated.
TxCB'	7	I	Transmit Clock (Channel B): The serial data are shifted out from the transmit data output (TxD8) on the falling edge of the transmit clock.
TxDB'	8	O	Transmit Data (Channel B): The serial data are shifted out from the transmit data output (TxDB) on the falling edge of the transmit clock.
RxDB'	9	I	Receive Data (Channel B): This pin transmits serial data to the communications channel (Channel B).
'SYNDETB/RTSB	10	I/O	Synchronous Detection (Channel B): This pin is used in byte synchronous mode as either an internal sync detect (output) or as a means to force external synchronization (input). In SDLC mode, this pin is an output indicating flag detection. In asynchronous mode it is a general purpose input (Channel B). Request To Send (Channel B): General purpose output, generally used to signal that Channel B is ready to send data. SYNDETB' or RTSB selection is done by WR2; D7 (Channel A)
DB7-DB0	12 - 19	I/O	Data Bus: The Data Bus lines are bidirectional three state lines which interface with the system's Data Bus.
CTSA'	39		Clear to Send (Channel A): This interface signal is supplied by the modem in response to an active RTS' signal. CTS' indicates that the data terminal/computer equipment is permitted to transmit data. In addition, if the auto enable control is set, the 8274 will not transmit data bytes until CTS' has been activated.
RTSA'	38	O	Request to Send (Channel A): general purpose output commonly used to signal that a Channel A is ready to send data.
TxDA	37	O	Transmit Data (Channel A): This pin transmits serial data to the communications channel (Channel A).
TxCA'	36	I	Transmit Clock (Channel A): The serial data are shifted out from the transmit data output (TxDA) on the rising edge of the receive clock.
RxCB'	35	I	Receive Clock (Channel A): The serial data are shifted out from the transmit data output (TxDA) on the rising edge of the receive clock.

Pin 20 = GND
Pin 40 = Vcc (+5 V power supply)
Pins 11, 30, 32, and 33 are not used.

Table 8.1. 8274 Serial Controller Pin Assignments

Symbol	Pin #	Type	Name and Function
RxDB'	34	I	**Receive Data (Channel A):** This pin receives serial data from the communications channel (Channel A).
DTRA'	31	O	**Data Terminal Ready (Channel A):** General Purpose output.
IPI/RxDRQB	29	I/O	**Interrupt Priority In/Receiver DMA Request (Channel B):** In modes 0 and 1 IPI' is Interrupt Priority In. A low on IPI' means that no higher priority device is being serviced by the controlling processor's interrupt service routine. In mode 2 this pin is RxDRQB and is used to request a DMA cycle for a receive operation (Channel B). In interrupt mode, this pin must be tied low.
INT'	28	O	**Interrupt:** The interrupt signal indicates that the highest priority internal interrupt requires service (open collector). Priority can be resolved via an external interrupt controller or a daisy-chain scheme.
INTA'	27	I	**Interrupt Acknowledge:** This interrupt Acknowledge signal allow the highest priority interrupting device to generate an interrupt vector. This pin must be pulled high (inactive) in non-vector mode.
DTRB'	26	O	**Data Terminal Ready (Channel B):** This is a general purpose output.
A0	25	I	**Address:** This line selects Channel A or B during data or command transfers. A low selects Channel A.
A1	24	I	**Address:** This line selects between data or command information transfer. A low means data.
CS'	23	I	**Chip Select:** This signal selects the MSPC and enables reading from or writing into its registers.
RD'	22	I	**Read:** Read controls a data byte or status byte transfer from the MPSC to the CPU.
WR'	21	I	**Write:** Write controls transfer of data or commands to the MPSC.

Pin 20 = GND
Pin 40 = Vcc (+5 V power supply)
Pins 11, 30, 32, and 33 are not used.

8.1.1.2 Timer

The serial controller contains a 24-pin chip that provides the internal timing for the RS-232-C subsystem.

Table 8.2 lists the appropriate time constraints that should be used to initialize the 8254 in order to achieve a desired baud rate. Because the 8254 input is driven by a fixed system clock, there is usually a small error between the exact clock frequency required for a given baud rate and the clock frequency produced by the 8254. The amount of expected error is also shown in table 8.2.

Table 8.2. Baud Rate Constraints

Baud Rate	Time Const	Error
9600	26	0.16 %
7200	35	0.79%
4800	52	0.16%
3600	69	0.64%
2400	104	0.16%
2000	125	0%
1800	139	0.08%
1200	208	0.16%
600	417	0.08%
300	833	0.04%
150	1667	0.02%
134.5	1859	0.01%
110	2272	0.03%
75	3333	0.01%
50	500	0%

Figure 8.3 illustrates the pins and signals of the 8254 timer. Table 8.3 lists the pins and explains their function.

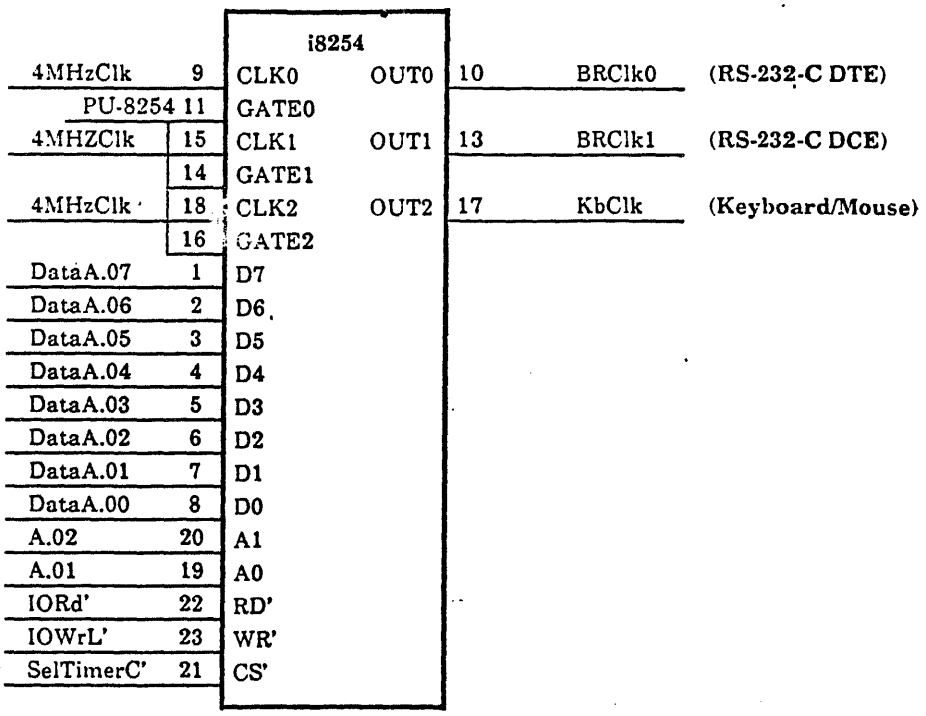


Figure 8.3. 8254 timer pins and signals

Table 8.3. 8254 Timer Pin Assignments

Symbol	Pin #	Type	Name and Function
A1-A0	19,20	I	**Address:** Used to select one of the three counters or the control word register for read or write operations. Normally connected to the system address bus. A1 A0 Selects 0 0 Counter 0 0 1 Counter 1 1 0 Counter 2 1 1 Control Word Register
Clk0	9	I	**Clock 0:** 4 MHz clock. from Counter 0.
CLK1	15	I	**Clock 1:** 4 MHz clock from Counter 1.
CLK2	18	I	**Clock 2:** 4 MHz clock fromCounter 2.
CS'	21	I	**Chip Select:** A low on this input enables the 8254 to respond to RD' and WR' signals. RD' and WR' are ignored otherwise.
D7-D0	1-8	I/O	**Data:** Bi-directional three state data bus lines, connected to system data bus.
Gate0	11	I	**Gate 0:** High.
Gate1	14	I	**Gate 1:** High.
Gate2	16	I	**Gate2:** High.
Out0	10	O	**Output 0:** Output of Counter 0 to RS-232-C DTE.
OUT1	13	O	**OUT1:** Output of Counter 1 to RS-232-C DCE.
OUT2	17	O	**OUT 2:** Output of Counter 2. to Keyboard/Mouse.
RD'	22	I	**Read Control:** Low during CPU read operations.
WR'	23	I	**Write Control:** Low during CPU write operations.

Pin 12 is Ground

Pin 24 is Vcc (+5 V power supply connection)

8.1.2 Interfaces

The 8274 operates in an interrupt-driven mode and a vector mode. Reset must be true for one complete system clock cycle for proper interrupt reset (250 ns). Wr' signal must be greater than 250 ns. One wait state is necessary to satisfy tRR, tWW, and tRD set-up and hold requirements. The leading edge of RD' sets the in-service latch. CPU should read RR2 of ChannelB to determine which internal source requested service (in non-vector mode). Before leaving the interrupt service routine, an End-of-Interrupt (EOI) must be sent to the 8274 to reset the highest priority source under service.

8.1.2.1 System Interface Signals

Figure 8.4 illustrates how the serial controller interfaces to the system. Table 8.4 briefly summarizes the signals (described in Table 8.1).

The top part of the figure illustrates IPI' generation. Note the following:

1. For a non-INTA cycle, IntrAckCycle is low, so IPI' is normally low, allowing the 8274 to generate an interrupt.

2. During an INTA cycle, if the IOP is not servicing this 8274, then 232IntrSeled' remains high, causing IPI' to be high during the second INTA pulse. This state aborts the 8274 INTA sequence and prevents delivery of the interrupt type.

3. During an INTA cycle, if the IOP is servicing this 8274, then 232IntrSeled' becomes low during the first INTA pulse, and remains low for the rest of the INTA cycle. IPI' thus remains low, and the 8274 issues the interrupt type to the bus.

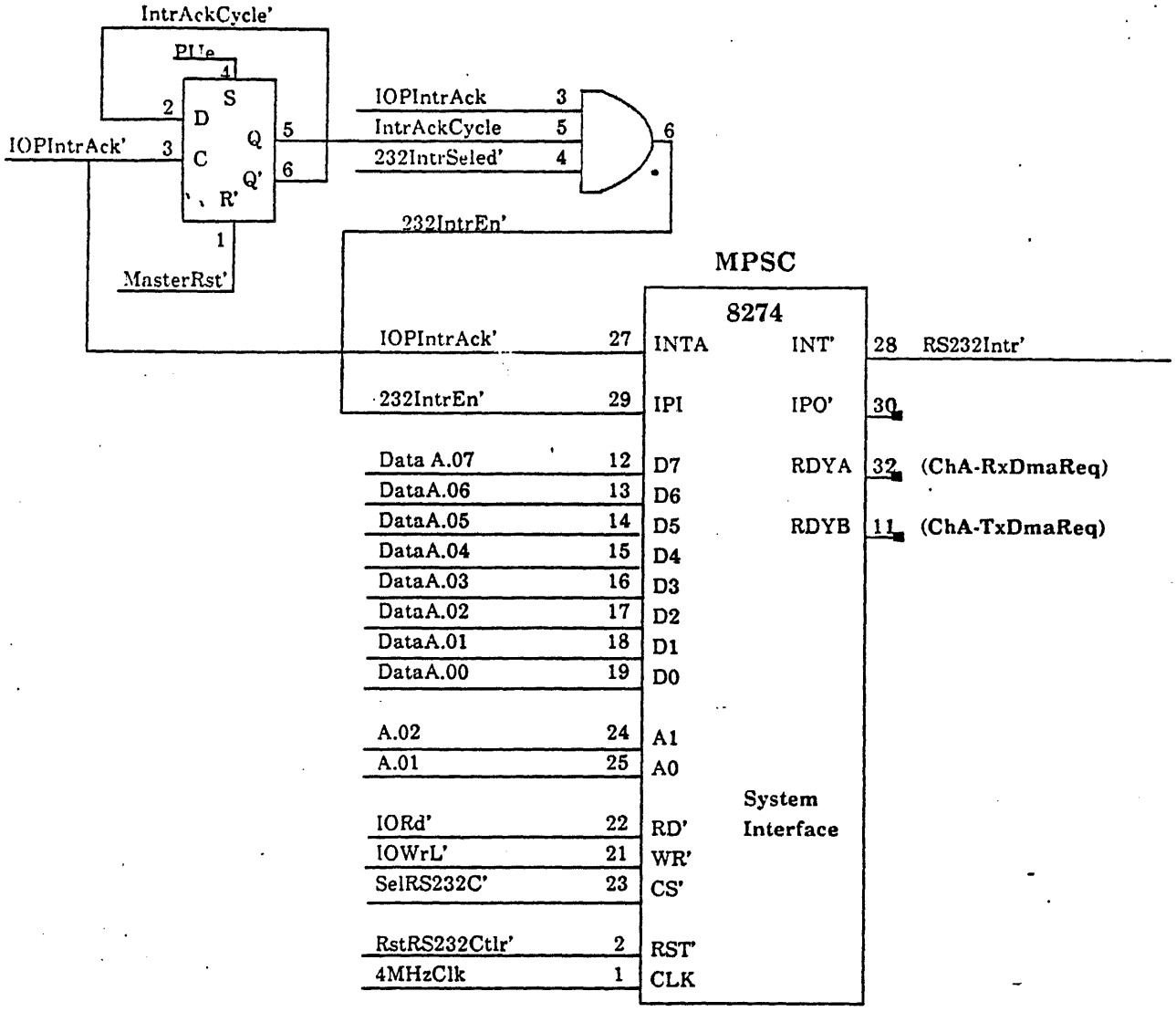


Figure 8.4. RS-232-C system interface pins and signals

Table 8.4. System interface pin assignments

Symbol	Pin #	Type	Name and Function
INTA	27	I	Allows the 8274 to generate an interrupt vector when the CAS lines decode to "4" (100). For polling, INTA' = H.
IPI	29	I	Indicates to the 8274 that the current interrupt acknowledge cycle is intended for the 8274. Refer also to text.
D7-D0	12-19	I/O	Three-state data lines used by the 80186 to communicate with the 8274.
INT'	28	O	Indicates that an interrupt condition has been encountered in the 8274.
A1	24	I	Selects between data (0) or command (1) registers during an access cycle to the 8274.
A0	25	I	Selects between channel A (0) or channel B (1) registers during an access cycle to the 8274.
RD'	22	I	Indicates that the current access cycle is a read cycle.
WR'	21	I	Indicates that the current access cycle is a write cycle.
CS'	23	I	Enables the 8274 for the current access cycle.
RST'	2	I	Resets the 8274 when driven to 0.
CLK	1	I	4 MHZ clock used by the 8274 as its internal system clock.

Pins 11, 30, and 32 are not used.

8.1.2.2 8274 Interface Ports

The 8274 interface has two ports: Channel A and Channel B. Channel A port is configured as a DTE port; Channel B is configured as a DCE port. The DCE (?) port connects to communication equipment for remote and standalone workstations. The DTE (?) port is primarily for local printers, and operates in asynchronous mode as a TTY port.

Figure 8.5 illustrates how the 8274 serial controller interfaces to the RS-232C channel A DTE port. Figure 8.6 illustrates the same for the RS-232 channel B DTE port. The 8274 signals are described in Table 8.1.

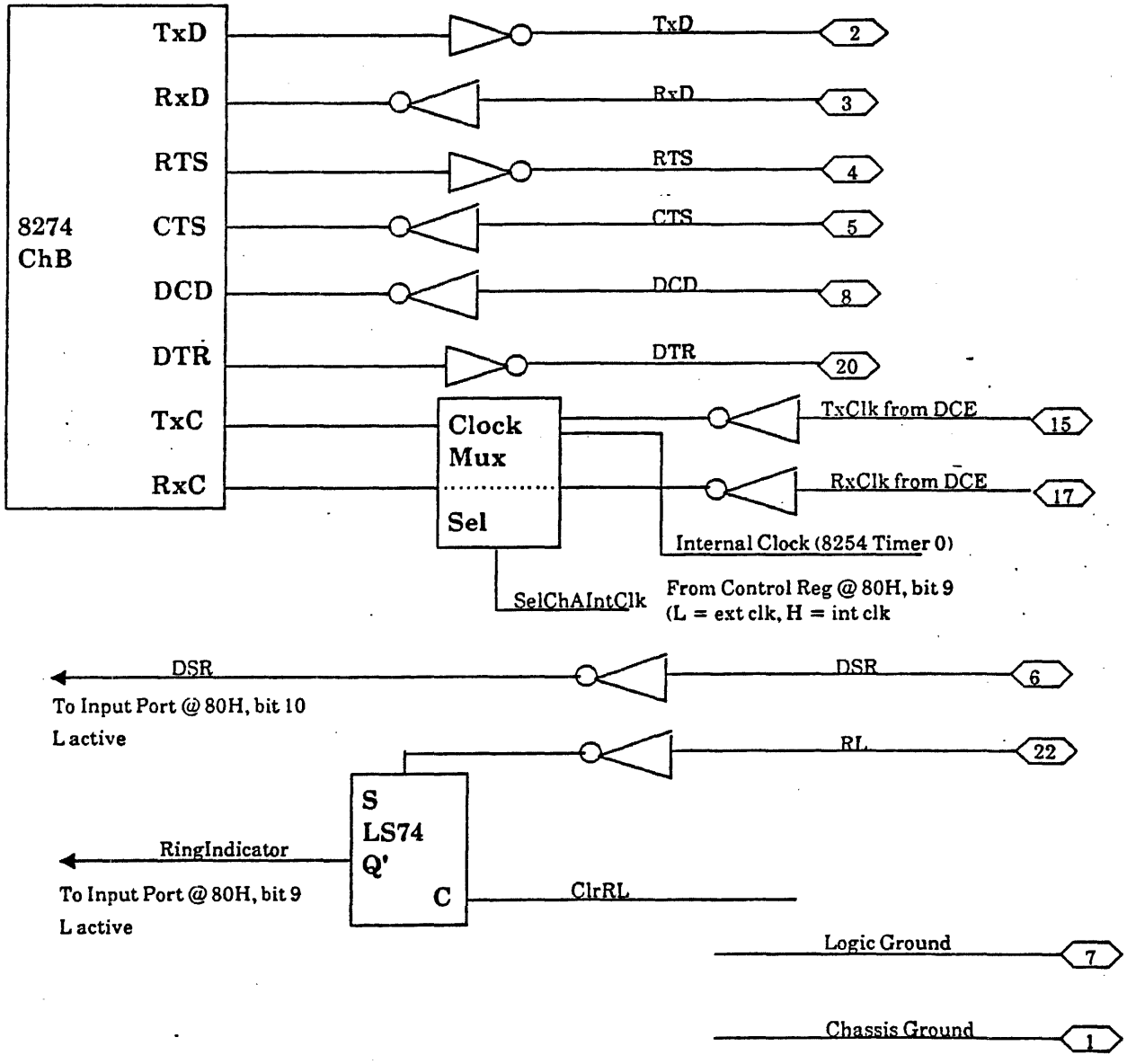


Figure 8.5. RS-232-C channel A DTE port

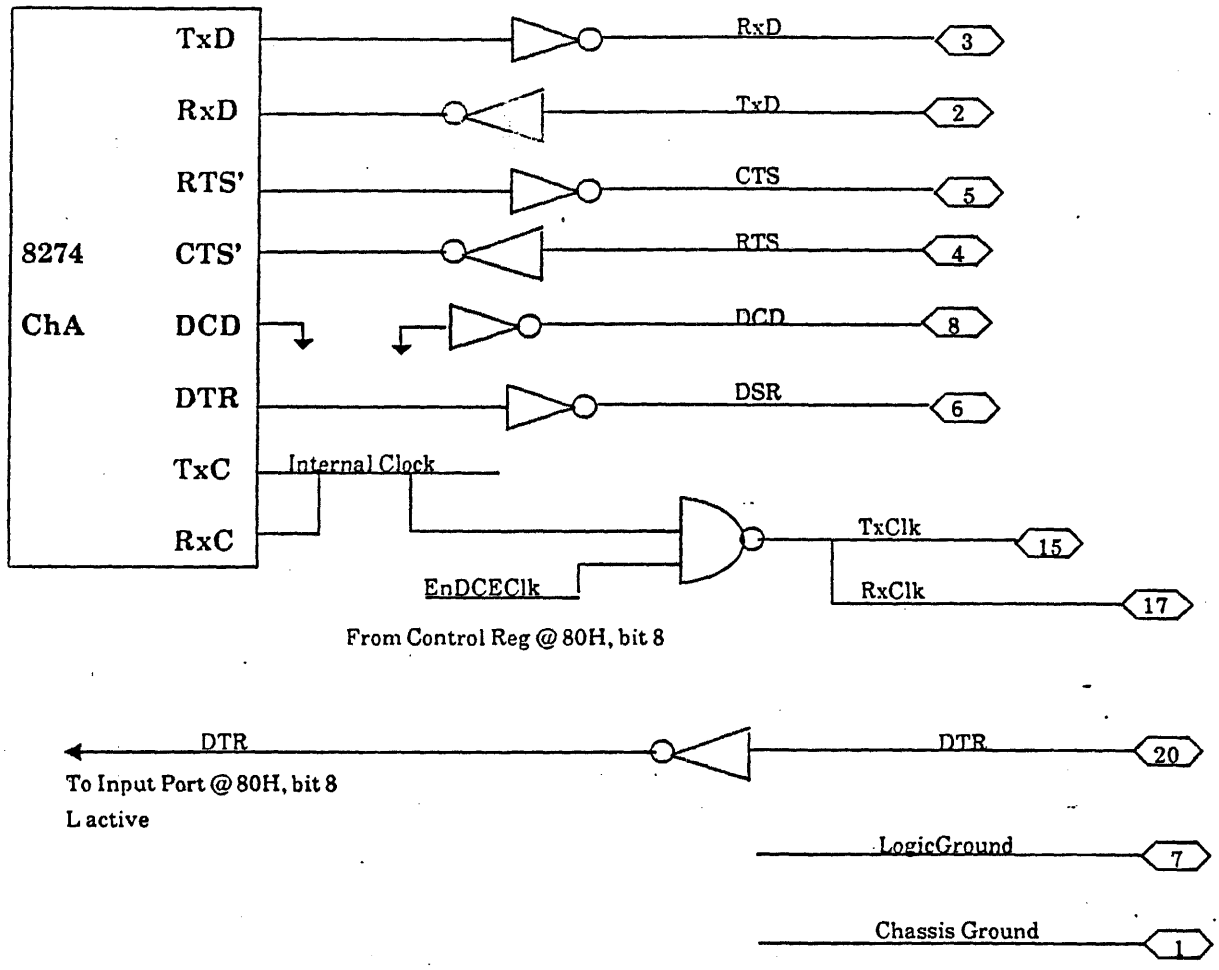


Figure 8.6. RS-232-C channel B DCE port

8.1.2.3 Interface Connectors

The RS-232-C interface uses a 25-pin DB type connector. Typically, DTE uses a male connector, and DCE uses a female connector. RS-232-C signal names are written to and read from the DTE.

Table 8.5 summarizes the interface signals that are available on the connectors. The direction of flow for signals between a DCE and a DTE port is also listed.

Table 8.5. Interface Signals

Connector Pin No	Signal Name	Direction of Flow	Signal Type	Active Level
1	GND Chassis Ground	DTE-DCE		
2	TxD Transmitted Date	DTE→DCE	Data	-12v
3	RxD Received Data	DTE←DCE	Data	-12v
4	RTS Request to Send	DTE→DCE	Control	+12v
5	CTS Clear To Send	DTE←DCE	Control	+12v
6	DSR Data Set Ready	DTE←DCE	Control	+12v
7	SG Signal Ground	DTE-DCE		
8	DCD Data Carrier Detect	DTE←DCE	Control	+12v
15	TxC Transmitter Clock	DTE←DCE	Timing	
17	RxC Receiver Clock	DTE←DCE	Timing	
20	DTR Data Terminal Ready	DTE→DCE	Control	+12v
22	RI Ring Indicator	DTE←DCE	Control	+12v

8.1.3 Serial Channels

The ports provide two independent full duplex channels; the channels are also illustrated in figure 8.5. Asynchronous, byte-and bit-synchronous transfers at data rates up to 9600 bps are possible. Byte-synchronous protocols include IBM Bisync; bit-syncronous protocols will include SDLC/HDLC.

Both channels provide asynchronous, byte-synchronous and bit-synchronous operations. Asynchronous transmissions have a baud rate up to 9600. Synchronous transmissions have a baud rate up to 9600. Both channels are interrupt-driven operation vector modes.

Signal voltages on the interface lines are nominally +12 volts and – 12 volts. There are two types of signals: data signals and control signals. Data signals are TxD and RxD. All other signals are control signals.

8.2 Theory of Operations

DMA resources are not available to the 8274, which is an interrupt-driven device. As an interrupt-driven device, the 8274 operates as a slave interrupt controller to the master interrupt controller. In this mode the 8274 produces up to eight different interrupt vectors for the 80186, depending on the kind of interrupt condition present in the chip.

Figure 8.7 illustrates the data paths of the RS-232 controller. The circled number is the sequence number, as described in the following outline of the 8274 acknowledge procedure.

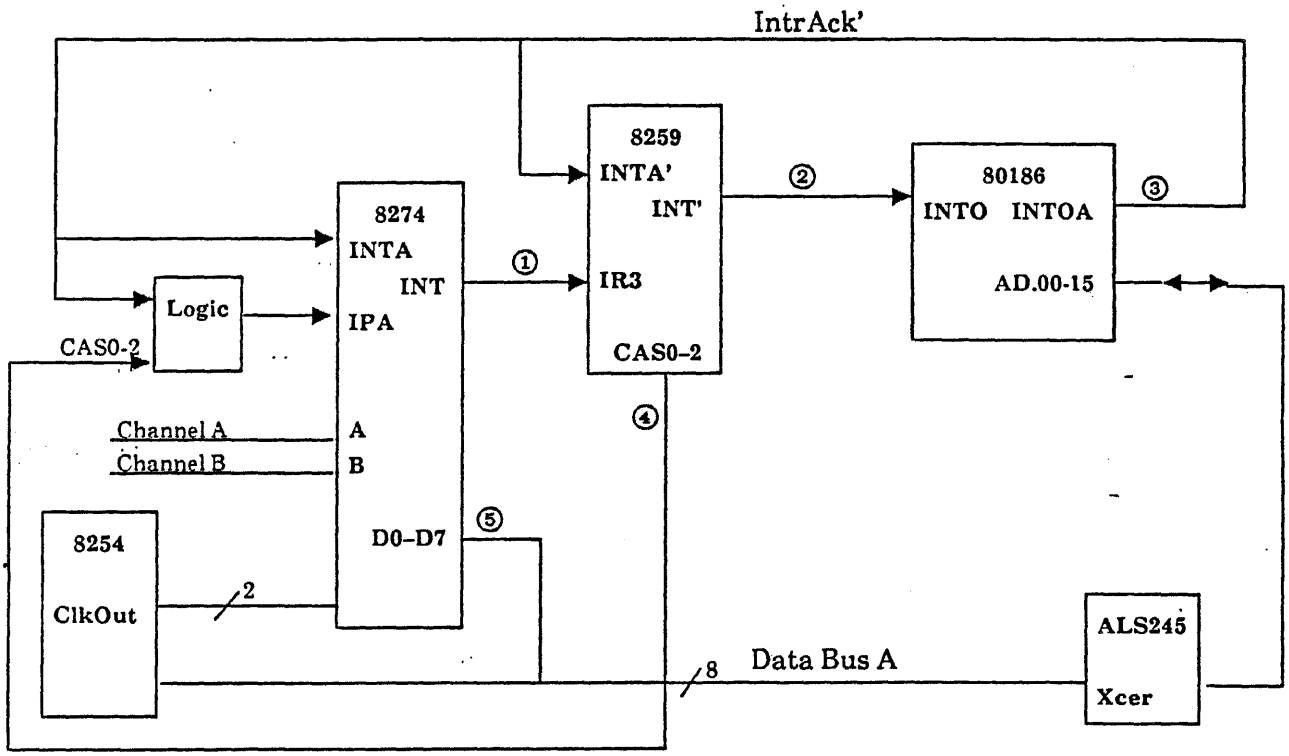


Figure 8.7. RS-232 system interface

1) When an internal interrupt condition is sensed by the 8274, the 8274 sends an interrupt request to the master interrupt controller 8259.

2) If the interrupt request meets the conditions of that time (which are?), then the 8259 sends an interrupt to the 80186 microprocessor.

3) When the 80186 is able to service the interupt, the 80186 responds with two contiguous (Consecutive? one signal with two branches?) interrupt acknowledge cycles.

4) The 8259 receives the first interrupt (INTA') pulse and generates the cascade signals, CAS0-2. These signals are then decoded by the 8274.

5) If CAS0-2 are decoded for the 8274, then the 8274 delivers the interrupt vector on the data bus-A during the second INTA cycle. The 80186 then processes the interrupt vector to obtain the address of the interrupt service routine in memory. The 80186 will begin execution at that location.

Figure 8.8 illustrates timing for the acknowlege operation.

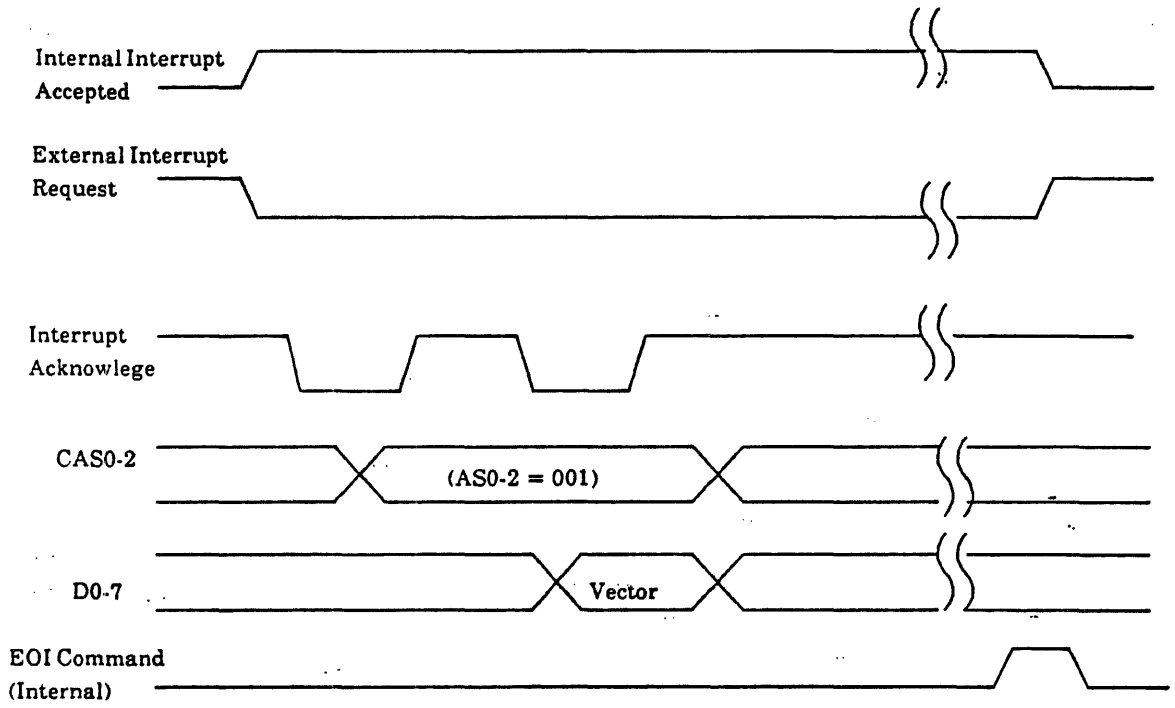


Figure 8.8. Timing for 8274 interrupt acknowledge

8.3 Programmer Interface: Registers

This section describes the 8274 serial controller registers, the 8254 timer registers, and the IOP registers that affect RS-232 operation.

8.3.1 External Registers

The IOP control and reset registers contain bits for RS-232-C operation.

The DTE port (CHA) has a ring indicator whose output (active high) may be obtained at bit 9 of the input register 80H. The ring indicator may be reset by an "IN" instruction to address A0H.

Also available at the input register are Channel A DSR signal state (active low) on bit 10 and the Channel B DTR signal (active low) on bit 8.

The DTE channel may use an internal receive and transmit clock from the 8254, or external clocks from the interface connector (SDLL). Bit 9 of the control register (80H) determines which clock is used, as follows:

0 = external clocks
1 = internal clock

The transmit and receive clock of the DCE port may be enabled or disabled from the interface connector by bit 8 of the control register (80H), as follows:

0 = drive is disabled, and pins 15 and 17 are held at an RS-232-C high level; that is, 12 V.
1 = clock driver enabled; drives the DCE clock onto pins 15 and 17 or to the DCE connector.

8.3.2 8274 Serial Controller Registers

The serial controller registers appear as eight I/O registers to the 80186. Each register is accessed by reading from or writing to a physical register location. Table 8.6 lists the register locations.

Table 8.6 Serial Controller Registers

Register	R/W	Address
Channel A Receive Data	R	40H
Channel A Transmit Data	W	40H
Channel B Receive Data	R	42H
Channel B Transmit Data	W	42H
Channel A Status	R	44H
Channel A Command	W	44H
Channel B Status	R	46H
Channel B Command	W	46H

To access a given register, the lower three bits of WR0 must be set to indicate which register will be read from or written to next. For example, to read RR3 or write WR3, xxxxx011 must be written to WR0; when the 8274 is reset, WR0 will point to 0, and the first access will read RR0 or write WR0.

Each channel of the serial controller has eight write registers for commands and three read registers for status.

8.3.2.1 Write Registers

Write registers on the 8274 should be initialized prior to use, as their reset states are not stable. Write register 4 (WR4) must be the first register initialized. Write xxxxx100 initializes WR4.

In addition, the 8254 should be initialized to provide the proper transmit or receive clock for the desired baud rate (refer to Table 8.2).

Tables 8.7 - 8.15 list the bit assignments for the write registers.

Table 8.7. Write Register 0 (WR0)

Bit	Assignment
D0-D2	Command Status Pointer Register Pointer.
D5-D3	Command 000 Null Code 001 Send Abort (SDLC) 010 Reset EXT/Status Interrupts 011 Channel Reset 100 Enable interrupt on next Rx Character 101 Reset TxINT/DMA Pending 110 Error Reset 111 End of Interrupt
D6-D7	Reset 00 Null Code 01 Reset Rx CRC Checker 10 Reset Tx CRC Generator 11 Reset Tx Underrun/EOM Latch

Table 8.8. Write Register 1 (WR1)

Bit	Assignment
D0	Ext Interrupt Enable
D1	TxInterrupt/DMA Enable
D2	Status Affects Vector Channel B only (Null Code CH A) 1 Variable Vector 0 Fixed Vector
D3-D4	Set Interrupt 00 RxINT/DMA Disable 01 RxInt on first Char or special condition. 10 INT ON ALL Rx Char (Parity Affects Vector) or Special Condition. 11 INT on all Rx char (parity does not affect vector or special condition).
D5	1 WAITON Rx 0 Disable
D6	Must be 0
D7	WAITENABLE 1 Enable 0 Disable

Table 8.9. Write Register 2 (WR2): Channel A

Bit	Assignment
D1-D0	Interrupt/DMA 00 Both Interrupt 01 A DMA, B INT 10 Both DMA 11 Illegal
D2	Priority 1 Priority RxA, RxB, TxA, TxB EXTA, EXTB 0 Priority RxA, TxA, RxB, TxB, EXTA, EXTB
D4-D3	Mode 00 8085 Mode 1 01 8085 Mode 2 10 8086/88 Mode 11 ILLEGAL
D5	Specify Interrupt type 1 Vectored Interrupt 0 Non-Vectored Interrupt
D6	Must be 0
D7	Pin select 1 Pin 10 SYNDET' 0 Pin 10 RTS

Table 8.10. Write Register 2 (WR2): Channel B

Bit	Assignment
D0-D7	All assigned as interrupt vectors.

Table 8.11. Write Register 3 (WR3): Channel B

Bit	Assignment
D0	Rx Enable
D1	SYNC char load inhibit
D2	ADDR SRCH MODE (SLDC)
D3	Rx CRC Enable
D4	Enter Hunt Mode
D5	Auto Enables
D6-D7	Bits/Char 00 Rx 5 Bits/Char 01 Rx 7 Bits/Char 10 Rx 6 Bits/Char 11 Rx 8 Bits/Char

Table 8.12. Write Register 4 (WR4)

Bit	Assignment
D0	Parity 1 Enable Parity 0 Disable Parity
D1	Parity 1 Even Parity 0 Odd Parity
D2-D3	Sync Modes 00 Enable Sync Modes 01 1 Stop Bit 10 1.5 Stop Bits 11 2 Stop Bits
D4-D5	Sync 00 8 Bit Sync Char 01 16 Bit Sync Char 10 SDLC/HLDC Mode (1111110) Flag 11 External Sync Mode
D6-D7	Clock 00 X1 Clock 01 X16 Clock 10 X22 Clock 11 X64 Clock

Table 8.13. Write Register 35 (WR5)

Bit	Assignment
D0	Tx CRC Enable
D1	RTS
D2	SDLC/CRC-16 (CRC Mode)
D3	Tx Enable
D4	Send Break
D5-D6	Transmit Character Length 00 Tx 5 Bits or Less/ Char 01 Tx7 Bits/Char 10 Tx 6 Bits/Char 11 Tx 8 Bits/Char

Table 8.14. Write Register 6 (WR6)

Bit	Assignment
D0-7	Least significant sync byte (address in SLDC/HLDC Mode)

Table 8.15. Write Register 7 (WR7)

Bit	Assignment
D0-7	Most significant sync byte. Must be 0111110 in SDLC/HDLC Mode

8.3.1.2 Read Registers

Tables 8.16 - 8.18 list the bit assignments for the three read registers.

Table 8.16. Read Register 0 (RR0)

Bit	Assignment
D0	Rx Char available
D1	IntPending (Char Only)
D2	Tx Buffer Empty
D3	Carrier Detect
D4	Sync/Hunt
D5	CTS
D6	Tx Underrun/EOM
D7	Break/Abort

Table 8.17. Read Register 1 (RR1): Special receive condition mode

Bit	Assignment		
D0	All sent		
D1-D3		I Field Bits Previous Byte	I Field Bits 2nd Previous Byte
	000	2	8
	001	0	6
	010	0	4
	011	0	8
	100	0	3
	101	0	7
	110	0	5
	111	1	8
D4	Parity Error		
D5	Rx Overrun Error		
D6	CRC/Framing Error		
D7	End of Frame(SDLD/HLDC Mode)		

Table 8.18. ReadRegister 2 (RR2)

Bit	Assignment
D0-D7	All assigned as interrupt vectors. A status variable determines vector mode.

8.3.2 8254 Timer Registers

Table 8.19 summarizes the read and write operation for the 8254 timer.

Table 8.19. Read/Write Operations Summary

CS'	RD'	WR'	A1	A0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 1
0	0	1	0	1	Read from Counter 1

9 Keyboard/Mouse Controller and Maintenance Panel

The keyboard controller supports a low-profile keyboard/mouse interface and a maintenance panel interface. The keyboard interface is an 8251A UART-based controller, an asynchronous serial interface with a data rate of 9600 bps. The IOP communicates through this interface to the keyboard processor.

The keyboard itself contains the mouse controller. The information transferred over the keyboard link contains both keyboard and mouse data. The maintenance panel puts out diagnostics and status information.

9.1 Hardware

Figure 9.1 illustrates the 8251A keyboard controller. Table 9.1 lists the pins and signals and describes their function.

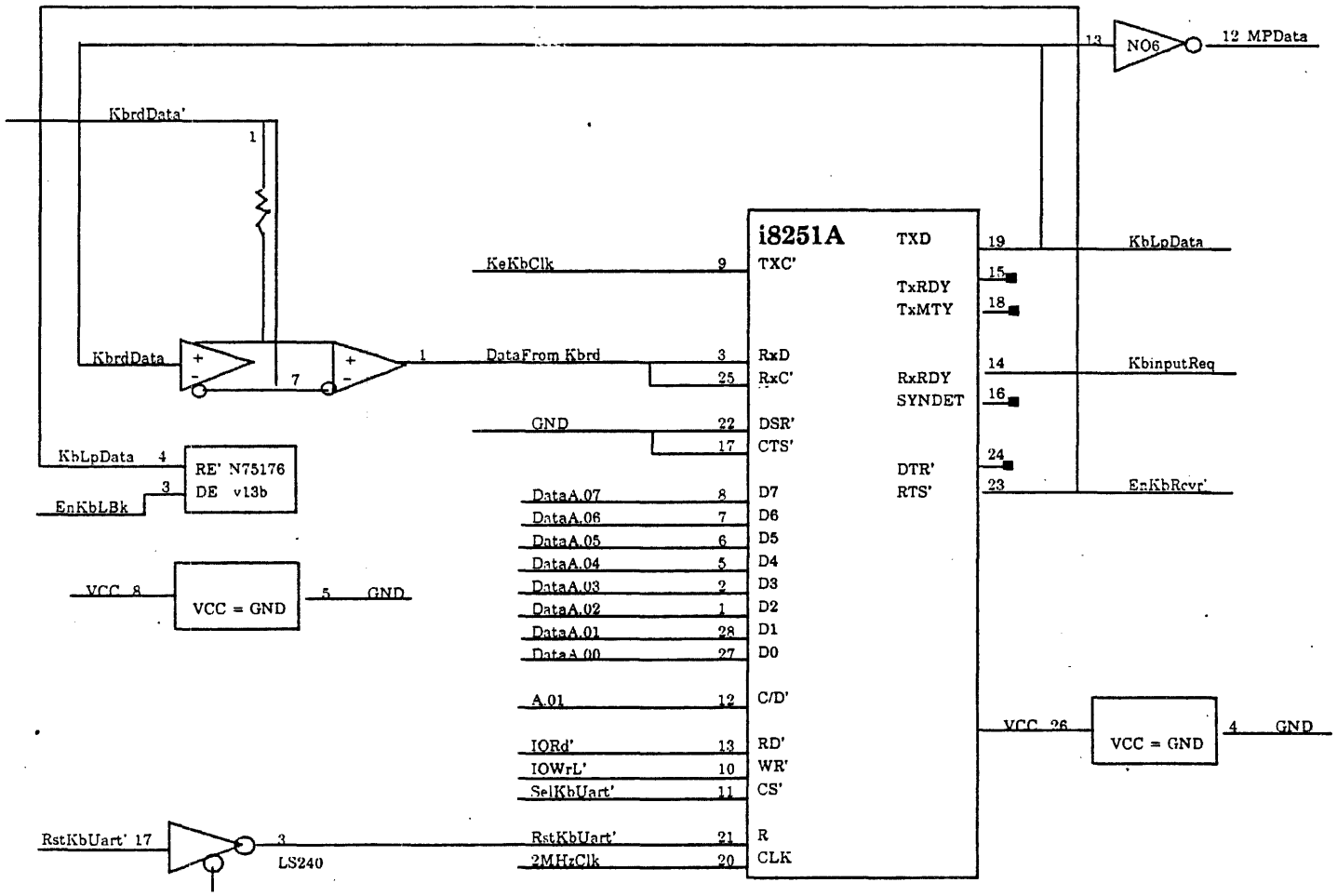


Figure 9.1. Keyboard interface

Table 9.1. 8251A Pin Assignments

(reprinted by permission of Intel Corporation)

Symbol	Pin No.	Connection To	Name and Function
TXC'	9	KbCLK	**Transmitter Clock:** The transmitter clock controls the rate at which characters are transmitted. This clock should be 16 times the data transfer rate.
RxD	3	DataFromKbrd	**Receive data:** Data line from the key board.
RxC'	25	KbClk	**Receive Clock:** Controls rate at which characters arrive at the 8251 RxD pin. This signal should be 16 times the data transfer rate.
D7	8	DataA.07	**Data 07-00:** Data bus connecting the 8251A to the 80186 for data and command exchange.
D6-4	7-5	Data.A06-04	
D3-2	2-1	Data.A03-02	
D1-0	28-27	Data.A01-00	
C/D'	12	A.01	**Command/Data Select:** Indicates type of access: 0 = data access; 1 = command access.
RD'	13	IORd'	**Read:** Control term indicating a read operation.
WR'	10	IOWrl'	**Write:** Control term indicating a write operation.
CS'	11	SelKbUart'	Chip Select or enable.
R	21	RStKbUart	8251A reset line.
CLK	20	2MHzClk	**Clock:** Input for the 8251A's system clock.
TXD	19	KbLpData	**TxD:** Data line to the Maintenance Panel Port.
RxRDY	14	KbinputReq	**INT:** the 8251A's interrupt line for initiating interrupts when a character is received.
RTD'	23	EnKbRcvr'	**RTS':** Control term used to enable the receiver on the keyboard data line.

Pins 15, 16, 18, and 24 are not used. Pins 17 and 22 are not used and are grounded.

9.2 Theory of Operations/Programming Interface

An introduction will be provided here.

9.2.1 Keyboard and Mouse Interface

The keyboard/mouse interface uses the receive half of the i8251A interface chip. Keyboard/mouse information is received from the keyboard as a differential signal pair by a 75176A receiver chip. The receiver chip translates the differential signals to a ttl signal and inputs the data to the i8251A via the RxD pin.

The transmit and receive clock is used by the 8251A: 1) to transmit and receive clocks used by the 8251A, and 2) to transmit and receive characters generated by timer 2 of the IOP 8254 timer chip. Refer to Section 8 (RS-232C).

9.2.2 Maintenance Panel

The maintenance panel (MP) runs at 9600 baud, 1 start bit, 1 stop bit, and no parity bit. Since these variables are the same as the keyboard variables, no differences occur when the i8251A is initialized.

The MP is connected to the transmit data pin (unused by the keyboard) on the i8251A. The MP is also connected to the keyboard reset line, so resetting the keyboard resets the MP board as well.

The MP has an 80-character buffer, but can only display 16 contiguous ASCII characters at a time.

The MP displays twelve preset messages by sending a 1-byte code for the desired message and the hex data for the message arguments. Table 9.2 lists the messages and their respective arguments.

Table 9.2. Maintenance Panel Code Message

Code Message	
Normal Commands	Special Commands
00H "Ex Status XXXX"	FB Define Character
01H "Ob Status XXXX"	FC Clear Maintenance Panel display
02H "Ex Data XXXX"	FD Rotate all characters in buffer
03H "Ob Data XXXX"	FE Maintenance Panel Command Literal
04H "To XXXXXX"	FF Maintenance Panel Data Literal
05H "From XXXXXX"	
06H "Test Number XXXX"	
07H "Fault Code XXXX"	
08H "Loc XXXXXX"	
09H "IO Command XXXX"	
0AH "Data XXXX"	
0BH "Status XXXX"	
0CH "Parity Err XXXX"	

The following subsections describe the operations for normal commands and special commands.

9.2.2.1 Normal Commands

If, for example, the desired message is "Status 1234", the following byte stream is sent by the IOP via the 8251A: 0B,12,34,1B. 0B is sent first to indicate which message is desired; then the arguments, 12 and 34, are sent. The message is terminated with a 1B.

The MP parses 0B and displays the appropriate message and the correct number of argument bytes. The MP stores all characters that are received in a buffer until an escape (esc) character (1BH) is received. The esc character is the signal to the MP to start parsing the message buffer.

Bit 7 modifies the meaning of all the standard messages (00-0C). If the bit is 0, then the maintenance panel clears the display. To clear a message from the display panel before the maintenance panel displays a new message, send 00HHHH1B. If the bit is 1, then the maintenance panel appends to the existing display buffer. To append the message to whatever is in the display, send 80HHHH1B. The display then places the new character on the right side of the display, and automatically scrolls the entire message to the left.

All data or argument bytes are sent in hex or binary format. The panel converts the data to ASCII just before it is placed in the display buffer.

9.2.2.2 Special Commands

Command FB defines the bitmap for custom characters. The maintenance panel can display up to four custom characters. All characters are displayed in a 5 x 11 matrix with the cursor being logically ORed with the bottom or 11th line.

To define a character, send the FB command byte followed by the character ID byte (00-03). This command is followed by 11 bytes that define the character's bitmap. Because the bitmap is only 5 bits wide, only the least significant 5 bits are used for the bitmap. The byte string is terminated with a 1BH.

Command FC causes the maintenance panel to clear its display. This command has no parameters.

Command FD causes the display buffer to be rotated or scrolled through the LCD display. This command has no parameters.

Command FE allows the programmer to send commands directly to the display controller. This command always appears as FEXX1B, where XX is the display controller command.

Command FF allows the programmer to create his own messages by following the FF with any string of ASCII characters; the message must be completed with an esc (1BH) character. For example, to display the message "A b C", the byte stream FF,41,20,62,20,43,1B must be sent.

All command messages must end in an esc 1BH character to initiate the parsing of a message. Multiple messages may be sent together; for example, 00XXXX81XXXX1B forms the message "Ex Status XXXX Ob Status XXXX" in the buffer. Because the display is only 16 characters long, only the first 16 characters are displayed. To display the rest of the message, the programmer can send an FD command to cause the display buffer to be rotated through the the LCD display.

9.2.3 Registers

The 8251A is one register with four register within it, thus appearing as four I/O registers to the 80186. The registers and their addresses are listed in Table 9.3.

Table 9.3. 8257A Registers

Register	Direction	Address
Receive Data Reg	Read	30H
Transmit Data Reg	Write	30H
Status Reg	Read	32H
Command Reg	Write	32H

9.2.3.1 Command Register

After a reset occurs, the 8251A operating mode must be set in the command register. The chip also expects the command register to be used for operation commands. Figure 9.2 illustrates the initialization process for the 8251A; Figure 9.3 illustrates the bit definitions for the command register during mode initialization.

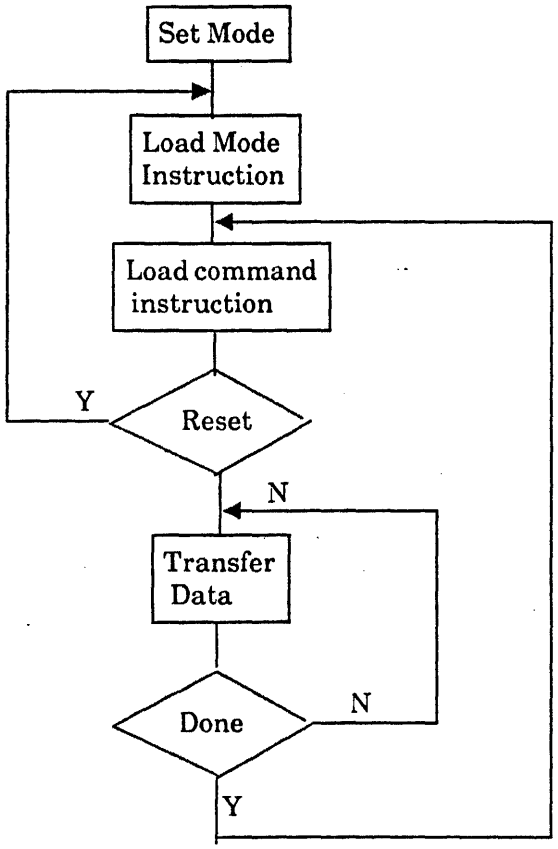


Figure 9.2. 8251A initialization

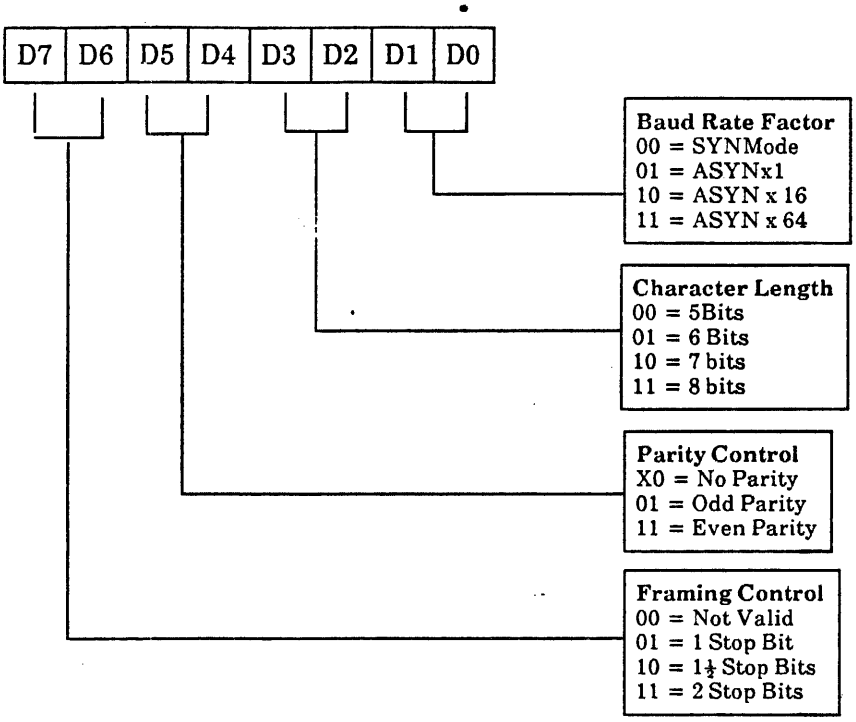


Figure 9.3 Command register during mode initialization
(Reprinted by permission of Intel Corporation)

Figure 9.4 illustrates the bit definitions for the command register during normal operation.

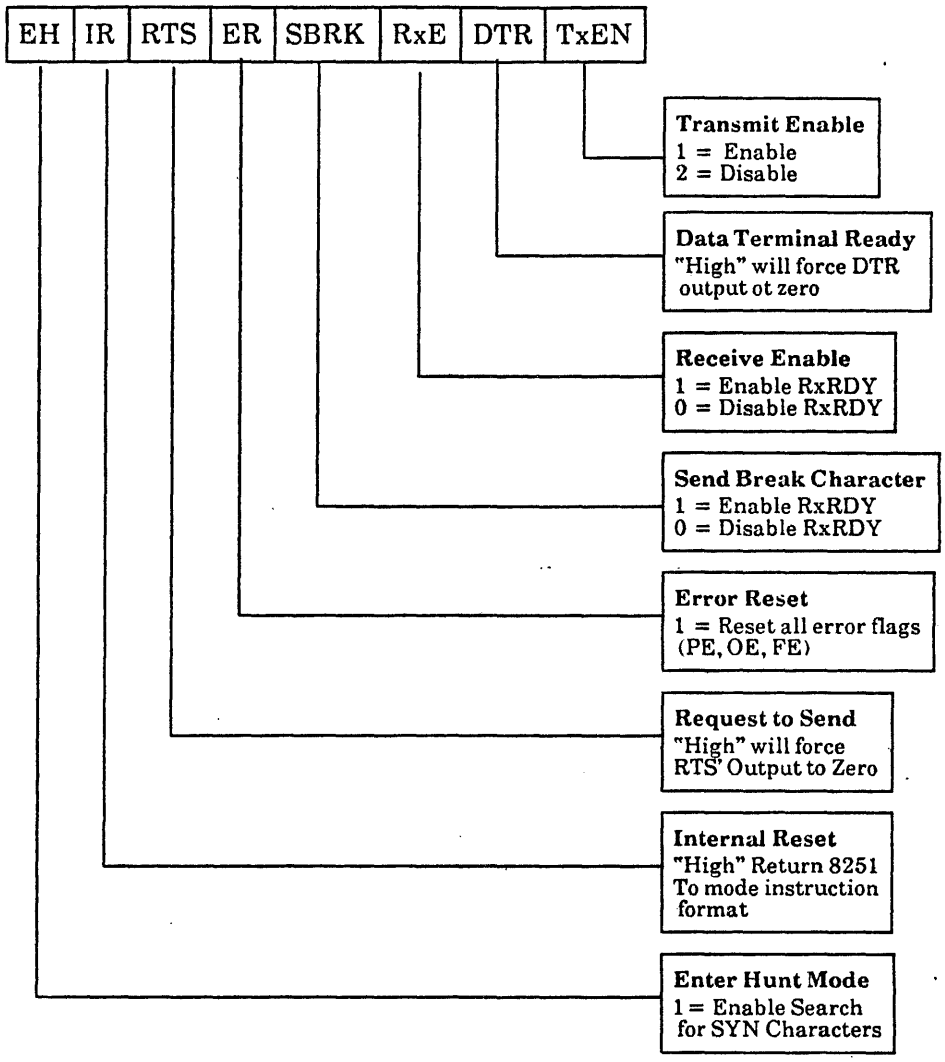


Figure 9.4. Command register during normal operation
(Reprinted by permission of Intel Corporation)

9.2.3.2 Status Register

The 8251A generates an interrupt at level IR3 when a character is received. There is no interrupt generated to indicate that the transmit buffer is empty. Because of this, programs that send information to the maintenance panel must "poll" the status register for the TxE bit go active indicating a transmit buffer empty condition.

The 8251A may be reset by driving bit 3 of the reset control register (C0H) to a logic of 0.

A diagnostic loopback that enables the transmit half of the 75176 (Where is this chip) to drive the receiver may be enabled by driving bit 14 of the reset control register to a logic 1. In this way, the receive logic can be fully tested.

Figure 9.5. illustrates the bit definitions for the status register. This register may be used to obtain the current status of the 8251A.

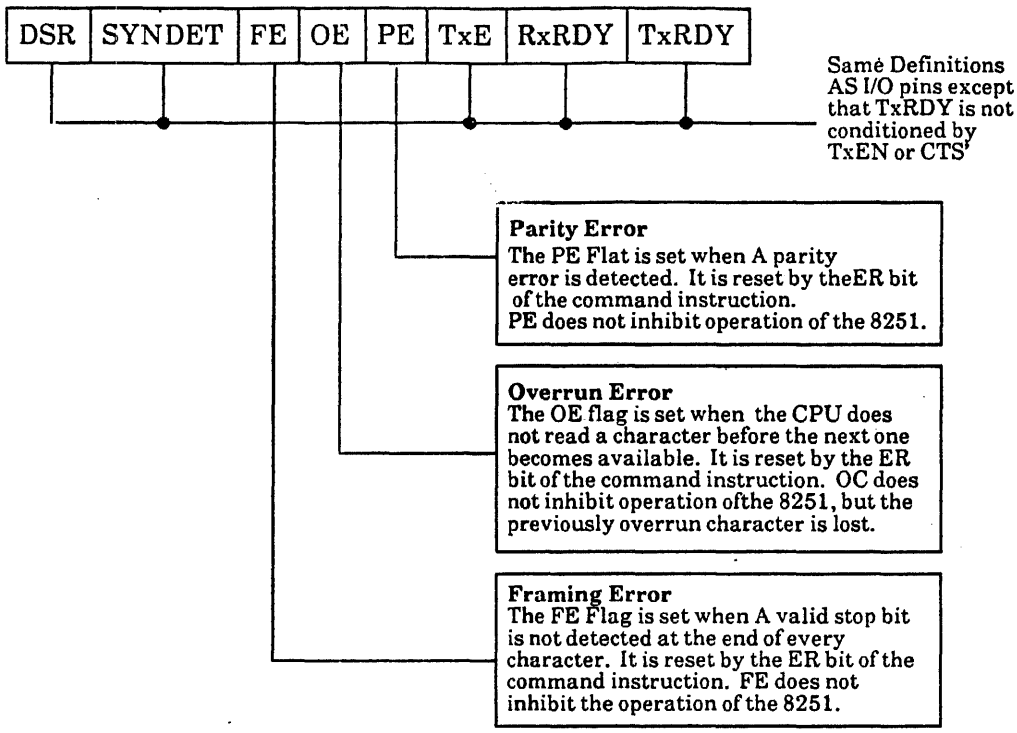


Figure 9.5 Status register

(Reprinted by permission of Intel Corporation)

10 IOP Options

The IOP architecture is extendable to provide for future addition of devices to the system. These devices, termed Options, are of two types, viz. the special PC emulation (PCE) option, and the general options. The PCE emulation function employs an additional 808186 processor and interfaces directly to the inner 80186 bus. The general options interface to a derived IOP bus. The options devices are housed in the Options slots. The Daisy workstation has 3 general options slots and the special PCE slot. The Daybreak workstation has one general options slot, together with the special PCE slot.

10.1 PC Emulation option

The PC emulation option uses a coprocessor identical to the IOP (i.e. an 80186). It shares the inner IOP system bus with the IOP and acts as a bus master when executing. It executes from a restricted memory address space in main memory. The allocation of bus control to the IOP 80186 or the PCE 80186 is controlled by the arbiter and mode control (see section 2.3).

All PCE 808186 I/O operations are trapped and serviced by the IOP. The PCE processor does therefore not have direct control of the PCE I/O devices. In addition, there is Mesa microcode support for the PCE display emulation.

The main components of the PCE function are as follows:

PCE 80186: This is the processor that executes the actual PC software. It executes using the 80186 system bus, and executes only out of main memory.

IOP/PCE Mode control: This logic determines which of the 808186 processor should have control of the 80816 bus. (The logic is located on the IOP board.)

I/O Trapper: This device latches all PCE I/O operations, latches 80186 status to distinguish read or write operations, and converts I/O operations to memory operation in the PC bank in main memory. The IOP is then interrupted, and the PC processor is removed as the bus master.

Display Trapper: A 16Kbyte area of main memory contains the PC display bit map. The PC display memory is divided into 50 areas with a dirty bit for each area in the display trapper. A master dirty bit is set by the IOP to alert Mesa processor microcode of any change to the bitmap. Mesa microcode will maintain the PC display bit map in PCE window.

10.2 General options

A buffered IOP bus is provided at the general options slots. This bus is thus not the internal systems bus, but is a derived IOP bus. Thus, not all I/O system functions are available on the options bus. The bus has a buffered address

bus (16 bits), and a buffered data bus (16 bits). 16 KBytes of 80186 I/O address space is allocated to the otions slots. 8 interrupts are available. No bus master or DMA capability is currently provided.

The options devices are thus intended to be moderate to low bandwidth devices. No high speed data paths are provided to the system, since the bandwidth availability is limited.

No general options have been defined in detail, but likely candidates that have been identified are: digital voice and telephone management, time and date support, and data encryption.

Appendix A Parts List for the IOP/RDC Board (Etch 2)

XEROX#	Mfg. PN	QTY	Mfger	Description
Western Digital Chipset				
733W02332	WD1100V-01	1	Western Digital	Serial-to-Parallel Converter (plastic package)
733W02333	WD1100V-03	1	Western Digital	Addr Mark Detector
733W02327	WD1100V-05	1	Western Digital	Parellel-to-Serial Converter
733W02328	WD1100V-06	1	Western Digital	ECC/CRC Logic
733W02329	WD1100V-07	1	Western Digital	Host Interface Logic
733W02330	WD1100V-09	1	Western Digital	Data Separator Support Logic
733W02331	WD1100V-12	1	Western Digital	MFM Generator
LSI Chips				
733W02361	N8X305N	1	Signetics	MicroController
	DP8460N-4	1	National	PLL Chip
733W01946	P8251A-KK	1	Intel	UART, KK-stepping
733W01594	P8254	1	Intel	PIT, Timer
733W02226	P8259A	3	Intel	PIC, Interrupt Controller
733W02249	P8272A	1	Intel	FDC, Floppy Disk Controller (Note 1)
733W02284	P8274	1	Intel	Multi-Protocol Serial Comm Controller
733W02192	D8288	1	Intel	Bus Controller
733W02267	C80186	1	Intel	8 MHz uP, B3 or later stepping
733W02289	C82586	1	Intel	Ethernet Controller
733W02382	S8023	1	Seeq	Ethernet Interface Chip
733W02358	FDC9229BT	1	Standard MicroSystem	Floppy Disk Interface Chip
733W02362	Am2942DC	1	AMD	DMA Address Generator
Memory Devices				
	D2764-2	2	Intel	EProm, 8K x 8, 200 ns
733W02505	HM6264P-15	2	Hitachi	SRAM, 8K x 8, 150 ns (Note 2)
733W01558	HM7603-5	1	Harris	TTL Prom, 32 x 8, 50 ns
733W02552	MK4501N-12	2	Mostek	Fifo, 512 x 9, 120 ns (Note 3)
733W01528	N82S181A	1	Signetics	TTL Prom, 1K x 8, 55 ns
	N82S191A	3	Signetics	TTL Prom, 2K x 8, 55 ns
733W01525	93422ADC	2	Fairchild	TTL Ram, 256 x 4, 38 ns
	NMC9346	1	National	Serial EEProm, 64 x 16 (Note 4)

Note 1 : Compatible parts: uPD765A, NEC

Note 2 : Compatible parts: TC5565P-15, Toshiba

Note 3 : Motorola may become a second source.

Note 4 : May not be in production yet. If so, use NMC9345.

Appendix AParts List for the IOP/RDC Board (Etch 2)

XEROX#	Mfg. PN	QTY	Mfger	Description
Western Digital Chipset				
733W02332	WD1100V-01	1	Western Digital	Serial-to-Parallel Converter (plastic package)
733W02333	WD1100V-03	1	Western Digital	Addr Mark Detector
733W02327	WD1100V-05	1	Western Digital	Parellel-to-Serial Converter
733W02328	WD1100V-06	1	Western Digital	ECC/CRC Logic
733W02329	WD1100V-07	1	Western Digital	Host Interface Logic
733W02330	WD1100V-09	1	Western Digital	Data Separator Support Logic
733W02331	WD1100V-12	1	Western Digital	MFM Generator
LSI Chips				
733W02361	N8X305N	1	Signetics	MicroController
	DP8460N-4	1	National	PLL Chip
733W01946	P8251A-KK	1	Intel	UART, KK-stepping
733W01594	P8254	1	Intel	PIT, Timer
733W02226	P8259A	3	Intel	PIC, Interrupt Controller
733W02249	P8272A	1	Intel	FDC, Floppy Disk Controller (Note 1)
733W02284	P8274	1	Intel	Multi-Protocol Serial Comm Controller
733W02192	D8288	1	Intel	Bus Controller
733W02267	C80186	1	Intel	8 MHz uP, B3 or later stepping
733W02289	C82586	1	Intel	Ethernet Controller
733W02382	S8023	1	Seeq	Ethernet Interface Chip
733W02358	FDC9229BT	1	Standard MicroSystem	Floppy Disk Interface Chip
733W02362	Am2942DC	1	AMD	DMA Address Generator
Memory Devices				
	D2764-2	2	Intel	EProm, 8K x 8, 200 ns
733W02505	HM6264P-15	2	Hitachi	SRAM, 8K x 8, 150 ns (Note 2)
733W01558	HM7603-5	1	Harris	TTL Prom, 32 x 8, 50 ns
733W02552	MK4501N-12	2	Mostek	Fifo, 512 x 9, 120 ns (Note 3)
733W01528	N82S181A	1	Signetics	TTL Prom, 1K x 8, 55 ns
	N82S191A	3	Signetics	TTL Prom, 2K x 8, 55 ns
733W01525	93422ADC	2	Fairchild	TTL Ram, 256 x 4, 38 ns
	NMC9346	1	National	Serial EEProm, 64 x 16 (Note 4)

Note 1 : Compatible parts: uPD765A, NEC

Note 2 : Compatible parts: TC5565P-15, Toshiba

Note 3 : Motorola may become a second source.

Note 4 : May not be in production yet. If so, use NMC9345.

XEROX#	Mfg. PN	QTY	Mfger	Description
74LS-- Parts				
733W01 671	74LS00	1	TI	
733W01704	74LS08	2	TI	
733W01713	74LS11	1	TI	
733W01752	74LS27	1	TI	
733W01705	74LS32	3	TI	
733W01675	74LS74A	6	TI	
733W01749	74LS109A	1	TI	
733W01706	74LS138	4	TI	
733W01662	74LS155	1	TI	
733W01741	74LS174	3	TI	
733W01642	74LS175	2	TI	
733W01676	74LS191	2	TI	
733W01625	74LS240	2	TI	
733W01626	74LS244	8	TI	
733W01748	74LS257A	1	TI	
733W01747	74LS259	1	TI	
733W01624	74LS273	4	TI	
733W01698	74LS374	4	TI	
74S-- Parts				
733W00318	74S00	4	TI	
733W00319	74S04	2	TI	
733W01611	74S08	1	TI	
733W01606	74S10	1	TI	
733W01644	74S11	1	TI	
733W01619	74S20	2	TI	
733W01646	74S32	3	TI	
733W01621	74S51	1	TI	
733W01620	74S64	1	TI	
733W01771	74S74	17	TI	
733W01616	74S138	3	TI	
	74S153	1	TI	
733W01652	74S157	1	TI	
733W01630	74S175	2	TI	
733W01633	74S240	2	TI	
733W01925	74S244	5	TI	
733W00351	74S257	3	TI	
733W00321	74S260	2	TI	
733W01699	74S373	3	TI	
733W01640	74S374	3	TI	

XEROX#	Mfg. PN	QTY	Mfger	Description
74ALS-- & 74AS-- Parts				
	74ALS08	1	TI	
733W02312	74ALS32	1	TI	
	74ALS244A	1	TI	
733W02337	74ALS245A	8	TI	
733W02290	74ALS273	3	TI	
733W02343	74ALS373	4	TI	
	74ALS374	3	TI	
	74AS27	1	TI	
733W02387	74AS109	1	TI	
74-- Parts				
733W00097	7406	4	TI	
733W00339	7414	2	TI	
733W00127	7438	1	TI	
75-- Parts				
	SN75176A	1	TI	Diff. Bus Transceiver
733W01717	SN75188	2	TI	RS232 Line Driver (also Motorola MC1488)
733W00098	SN75189A	3	TI	RS232 Line Rcvr (also Motorola MC1489A)
Miscellaneous				
733W01972	Am26LS31C	1	AMD, TI	RS422 Driver
733W02909	Am26LS32AC	1	AMD, TI	RS422 Receiver
733W01809	K1100A, 20 MHz	1	Motorola	Oscillator, 0.01%
	K1114A, 32 MHz	1	Motorola	Oscillator, 0.05%
144P10005	LPSLDM-60	3	Engineered Components Co.	Delay Line, 12 ns per step, 60 ns total
707W00642	1N4003	1	Motorola	Diode
707W00273	1N4148	1	Motorola	Diode
707W01916	2N2905A	1	Transistor	
1044301	2N5320	1	Transistor	
705W00037		2	Inductor, 100 uH	
708W11502		1	Fuse, 15A	
708W10902		1	Fuse, 2A	
708W10302		2	Fuse, 250 mA	

XEROX#	Mfg. PN	QTY	Mfger	Description
Resistor Paks				
	110A153	3	Allen Bradley	10 pin SIP, 15K (Beckman 765-3-R15K)
	110E221331	2	Allen Bradley	10 pin SIP, 220/330 (Beckman 765-5-R220/330)
	316A103	1	Allen Bradley	16 pin DIP, 10K (Beckman 898-1-R10K)
	316A102	1	Allen Bradley	16 pin DIP, 1K (Beckman 898-1-R1K)
	316A202	1	Allen Bradley	16 pin DIP, 2K (Beckman 898-1-R2K)
	316A512	2	Allen Bradley	16 pin DIP, 5.1K (Beckman 898-1-R5.1K)
	tbd	1	tbd	Ethernet 11-pin SIP R-Pak
Discrete Resistors				
703W37088		1		100K, 5%, 1/4 W
703W33888		2		4.7K, 5%, 1/4 W
703W33088		2		2.2K, 5%, 1/4 W
703W32988		1		2.0K, 5%, 1/4 W
703W32688		2		1.5K, 5%, 1/4 W
703W32288		2		1.0K, 5%, 1/4 W
703W31488		1		470, 5%, 1/4 W
703W31088		1		330, 5%, 1/4 W
703W30588		2		200, 5%, 1/4 W
703W30288		1		150, 5%, 1/4 W
703W29888		2		100, 5%, 1/4 W
703W29188		1		51, 5%, 1/4 W
703W25406		2		3.5K, 1%, 1/4 W
703W21906		2		1.5K, 1%, 1/4 W
703W20206		1		1.0K, 1%, 1/4 W
703W11406		2		120, 1%, 1/4 W
703W06706		4		39, 1%, 1/4 W
703W28687		1		33, 5%, 1/2 W
Discrete Capacitors				
702W10701		4		35 uF, Al. Elect. Cap, 50v
702W10401		4		10 uF, Al. Elect. Cap, 50v
702W05218		10		0.1 uF, Ceramic Cap, 50v
702W05018		105		0.01 uF, Ceramic Cap, 50v
702W05718		1		1.0 uF, Ceramic Cap, 10%, 100v
702W06518		3		0.1 uF, Ceramic Cap, 10%, 100v
702W03118		4		0.01 uF, Ceramic Cap, 10%, 100v
702W02418		2		0.001 uF, Ceramic Cap, 10%, 100v
702W01818		2		330 pF, Ceramic Cap, 10%, 100v
Type CD30	FD682J03	1	Cornell-Dubilier	6800 pF, Dipped Mica Cap, 5%
Type CD10	FD151J03	1	Cornell-Dubilier	150 pF, Dipped Mica Cap, 5%
Type CD10	ED470J03	2	Cornell-Dubilier	47 pF, Dipped Mica Cap, 5%

XEROX#	Mfg. PN	QTY	Mfger	Description
Connectors				
	532431-8	1	AMP	HDI PCB Connector, 165-pin
	532431-3	1	AMP	HDI PCB Connector, 96-pin
	745784-5	1	AMP	DB-37 Connector
	745783-5	2	AMP	DB-25 Connector
	745782-5	1	AMP	DB-15 Connector
	745781-5	1	AMP	DB-09 Connector
	1-640098-4	1	AMP	MTA-100 Post Header, 14 pins
Sockets				
	268-5400	1	TexTool/3M	68 pin JEDEC Chip Carrier (for 80186)
	Pn#: 664	1	EG&G Wakefield Eng.	80186 Heatsink (1/4 in. high)
	PPS68-AG1D	1	Augat	68-pin Pin-Grid Socket (for 80186)
	248-AG1A	1	Augat	48-pin DIP socket (for 82586)
	21950-47-446	1	EMC	50-pin DIP socket , 0.90" width (for 8x305)
	504-AG10D	2	Augat	4-pin DIP sockets for oscillator
	508-AG10D	1	Augat	8-pin DIP sockets
	514-AG10D	tbd	Augat	14-pin DIP sockets
	516-AG10D	tbd	Augat	16-pin DIP sockets
	518-AG10D	tbd	Augat	18-pin DIP sockets
	520-AG10D	tbd	Augat	20-pin DIP sockets
	522-AG10D	3	Augat	22-pin DIP sockets (for 2942 & 93422)
	524-AG10D	4	Augat	24-pin DIP sockets (for 82S121)
	528-AG10D	8	Augat	28-pin DIP sockets (for EProm & SRAM)
	540-AG10D	2	Augat	40-pin DIP sockets
	510-AG90D-10-3	5	Augat	10-pin SIP sockets (for SIP R-paks)

Appendix G Floppy Disk Controller Command Instruction Set

The tables in Appendix G list the command and result bytes for the floppy disk controller (FDC) operations, as follows:

G.1 Read data
G.2 Read deleted data
G.3 Write data
G.4 Write deleted data
G.5 Read a track
G.6 Read ID
G.7 Format a track
G.8 Scan equal
G.9 Scan low or equal
G.10 Scan high or equal
G.11 Recalibrate
G.12 Sense interrupt status
G.13 Specify
G.14 Sense drive status
G.15 Seek
G.16 Invalid

The command bytes for each operation must be written to the floppy disk controller in the exact order indicated in the table. The result bytes for each operation must be read in the exact order indicated in the table. Failure to write all command bytes or read all result bytes will leave the floppy disk controller inoperative.

The figures following the tables illustrate the necessary sequences in the command, execution, and result phases of each of the instructions.

Figure G.1 Command phase
Figure G.2 Execution phase (read and write instructions)
Figure G.3 Result phase (read and write instructions)
Figure G.4 Seek, recalibrate, sense interrupt status, and invalid instructions

Table G.1. Read Data Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	MT MF SK 0 0 1 1 0 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL DTL	Sector ID information prior to Command execution.
Execution			Data transfer between the FDD and main-system.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.2. Read Deleted Data Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	MT MF SK 0 1 1 0 0 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL DTL	Sector ID information prior to Command execution.
Execution			Data transfer between the FDD and main-system.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.3. Write Data Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	MT MF 0 0 0 1 0 1 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL DTL	Sector ID information prior to Command execution.
Execution			Data transfer between the FDD and main-system.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.4. Write Deleted Data Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	MT MF 0 0 1 0 0 1 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL DTL	Sector ID information prior to Command execution.
Execution			Data transfer between the FDD and main-system.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.5. Read a Track Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	0 MF SK 0 0 0 1 0 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL DTL	Sector ID information prior to Command execution.
Execution			Data transfer between the FDD and main-system. FDC has read all of the cylinder's contents from index hole to EOT.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.6. Read ID Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	0 MF 0 0 0 0 1 0 X X X X X HD US1 US0	Commands
Execution			The first correct ID information on the cylinder is stored in the Data Register.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution.

Table G.7. Format a Track Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W	0 MF SK 0 0 0 1 0	Command Codes
	W	X X X X X HD US1 US0	
	W	——N——	Bytes/Sector
	W	——SC——	Sectors/Track
	W	——GPL——	Gap 3
	W	——D——	Filler Byte
Execution			FDC formats an entire cylinder.
Result	R	——ST0——	Status Information after
	R	——ST1——	Command execution.
	R	——ST2——	
	R	——C——	In this case, the ID
	R	——H——	information has no
	R	——R——	meaning.
	R	——N——	

Table G.8. Scan Equal Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W	MT MF SK 1 0 0 0 0	Command Codes
	W	X X X X X HD US1 US0	
	W	——C——	Sector ID information
	W	——H——	prior to Command
	W	——R——	execution.
	W	——N——	
	W	——EOT——	
	W	——GPL——	
	W	——DTL——	
Execution			Data compared between the FDD and main system.
Result	R	——ST0——	Status Information after
	R	——ST1——	Command execution.
	R	——ST2——	
	R	——C——	Sector ID information after
	R	——H——	Command execution
	R	——R——	
	R	——N——	

Table G.9. Scan Low or Equal Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	MT MF SK 1 1 0 0 1 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL STP	Sector ID information prior to Command execution.
Execution			Data compared between the FDD and main system.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.10. Scan High or Equal Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	MT MF SK 1 1 1 0 1 X X X X X HD US1 US0	Command Codes
	W W W W W W W	C H R N EOT GPL STP	Sector ID information prior to Command execution.
Execution			Data compared between the FDD and main system.
Result	R R R R R R R	ST0 ST1 ST2 C H R N	Status Information after Command execution. Sector ID information after Command execution

Table G.11. Recalibrate Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	0 0 0 0 0 1 1 1 X X X X X HC US1 US0	Command Codes
Execution			Head retracted to Track 0.

Table G.12. Sense Interrupt Status Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W	0 0 0 0 0 0 0 0	Command Codes
Result	R R	——————STO—————— ——————PCN——————	Status information at the end of seek operation about the FDC.

Table G.13. Specify Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W W	0 0 0 0 0 0 0 0 ——SRT——→←———HUT——— ——HLT————————→ND	Command Codes

Table G.14. Sense Drive Status Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W	0 0 0 0 0 1 1 1 X X X X X HD US1 US0	Command Codes
Result	R	——————ST3——————	Status information about FDD.

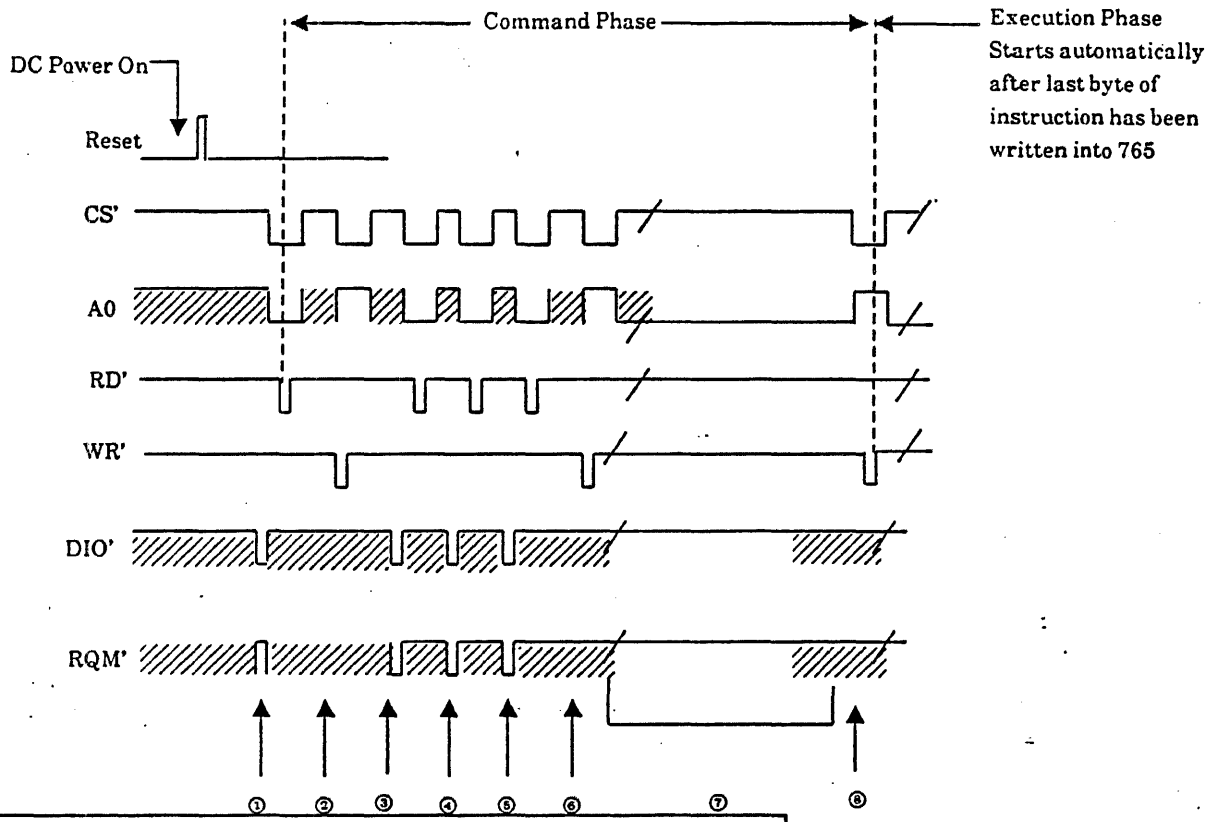
Table G.15. Seek Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W W W	0 0 0 0 1 1 1 1 X X X X X HD US1 US0 NCN	Command Codes
Execution			Head is positioned over proper Cylinder on Diskette

Table G.16. Invalid Instruction Set

Phase	R/W	Data Bus D7 D6 D5 D4 D3 D2 D1 D0	Remarks
Command	W	Invalid Codes	Command Codes
Result	R	ST0	ST0-80 (16)

Note: Shaded portion indicates
don't care state



Key:
① Processor reads main status reg. (MSG) Does RQM =1 and DIO = 0, Yes, then write first byte of instruction into 765.
② First byte of instruction written into 765 by processor.
③ Processor reads MSR, does RQM = 1 and DIO = 0, No then do it again.
④ No Do again, RQM still = 0.
⑤ Does RQM = 1 and DIO = 0, yes then write second byte of instruction into 765.
⑥ Second byte of instruction written into 765 by processor.
⑦ Repeat steps 3 through 6 until all bytes in instruction have been written into 765
⑧ Last byte of instruction written into 765 by processor.

Figure G.1. Command phase

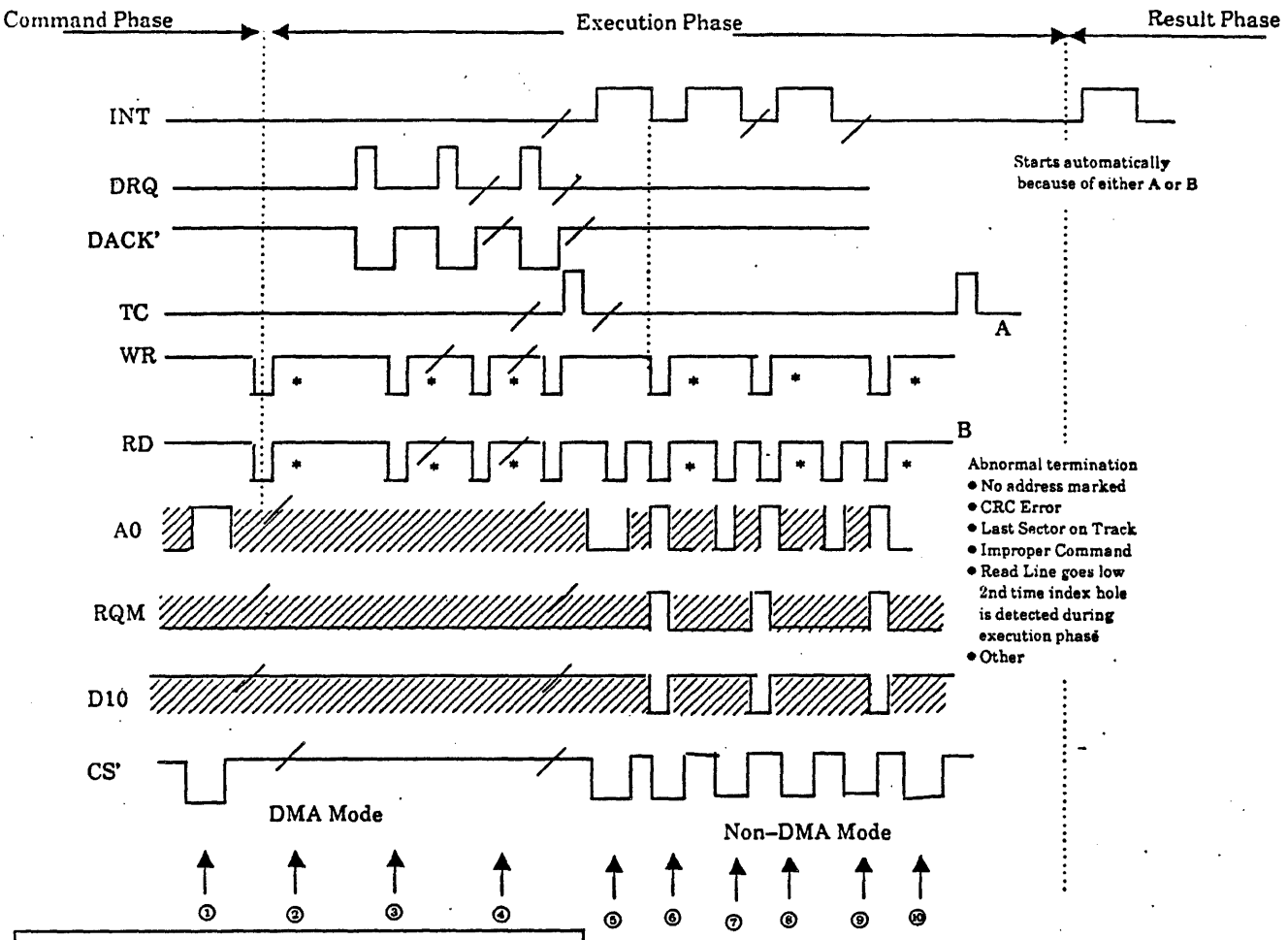
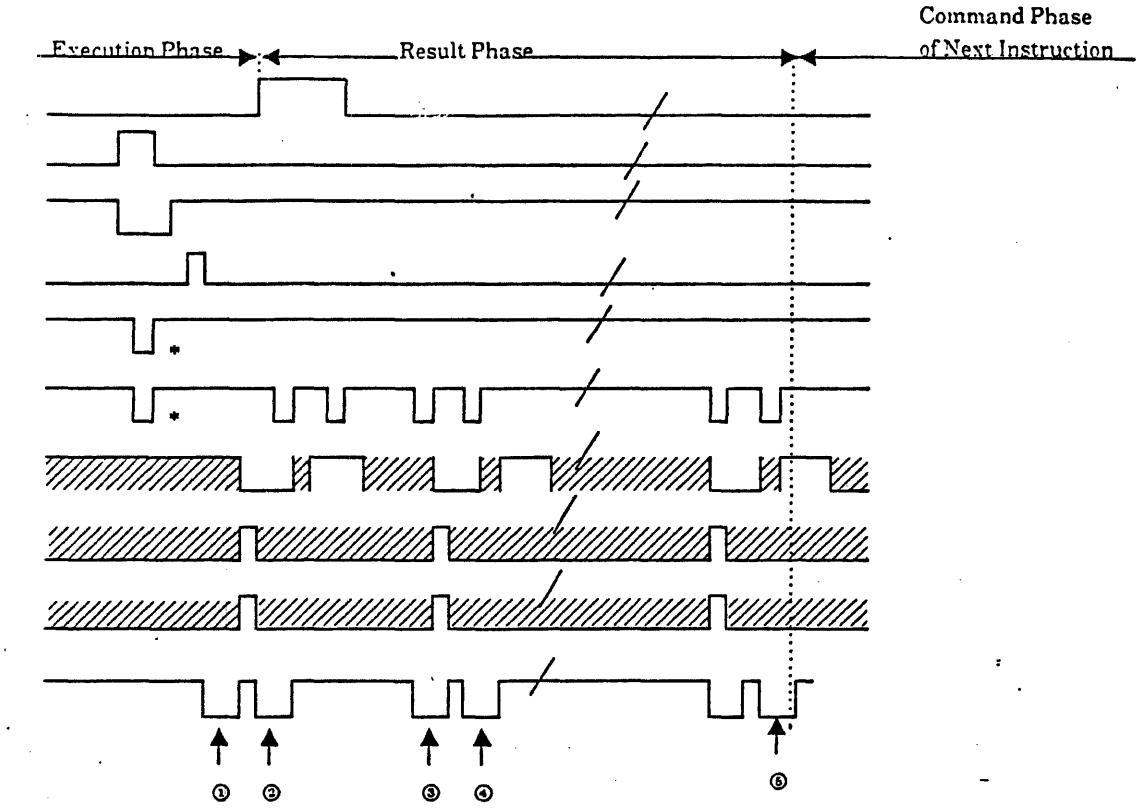
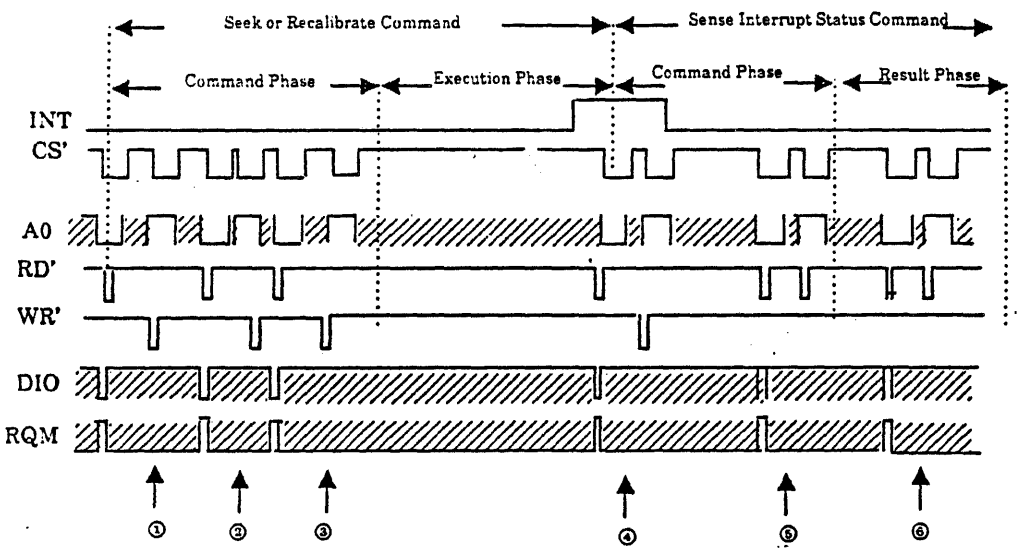


Figure G.2. Execution phase (read and write instructions)



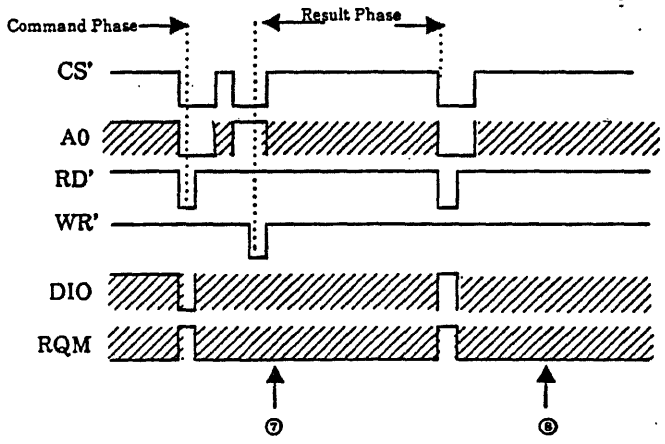
Key:
① Processor Reads MSR, Does RQM = 1 and DIO = 1, yes then
② processor read 1st byte from result phase.
③ Same as Step ①
④ Same as step ② (second byte)
⑤ Same as step ② (last byte)
* Either RD' (if it is a read instruction)
or WR' (if it is a write instruction) but never both.

Figure G.3. Result phase (read or write instructions)



Seek, recalibrate, and Sense Interrupt Status

Key:
① OP code for instruction written into 765
② HD/Drive No written into 765
③ INC written into 765
④ OP code for instruction written into 765
⑤ Status Register STO read by processor
⑥ RCN read by processor
⑦ Invalid instruction issued to 765
⑧ Status register STO read by processor



Invalid

Figure G.4. Seek, Recalibrate, Sense Interrupt Status and Invalid Instructions