Designing an ESDI (Enhanced Small Device Interface) Disk Controller Subsystem with National's DP8466A (Disk Data Controller)

National Semiconductor
Application Note 500
November 1987

Designing an ESDI (Enhanced Small Device Interface) Disk Controller Subsystem with National's DP8466A (Disk Data Controller)

AN-500

1.0 INTRODUCTION

The ESDI (Enhanced Small Disk Interface) is designed to handle a variety of 5¼″ Winchester disk, tape and optical drives. It opens the door to higher performance system designs by incorporating more intelligence onto the drives, and by allowing higher data transfer rates—10 Mbits/s to 24 Mbits/s. This is achieved by incorporating data separation, data encoding and decoding in the drive itself and the smarter interface protocol allows dissemination of more information between the drive and the controller. Thus by removing the restrictions placed by ST506 on 5¼″ hard disk transfer rates, the ESDI interface clears the way for higher recording densities and ultimately, higher storage capacities in the 5¼″ form factor, up to 700 Mbytes and beyond.

National Semiconductor's DP8466A Disk Data Controller integrates a number of functions originally supported by discrete logic in conventional disk controller designs. This results in a decrease in complexity and parts count in a disk controller design. The DP8466A is a data path controller and hence can support the various disk interfaces viz. ST506, ESDI, SMD, etc. By its versatility and programmability, it greatly simplifies the task of designing a disk controller. There are basically two types of ESDI drives viz. hard sectored ESDI drives and soft sectored ESDI drives. This application note discusses the ESDI interface and the various steps involved in designing an ESDI disk controller with the DP8466A. The emphasis is predominantly on the disk side as that is of utmost relevance with respect to intefacing the DP8466A.

2.0 ENHANCED SMALL DEVICE INTERFACE (ESDI)

The ESDI consists of a control cable and a data cable. The control cable allows for a daisy chain connection of up to seven drives with only the last drive being terminated. The data cable must be attached in a radial fashion. *Figure 1* shows a typical connection in a multiple drive system. All control lines are digital in nature (open collector TTL) and either provide signals to the drive (input) or signals to the controller (output). The data transfer signals are differential in nature and provide data either to, (write) or from, (read), the drive.

2.1 Control Cable

The control cable definition is shown in *Figure 2*. It basically consists of some input lines and some output lines. The control input signals are of two kinds; those to be multiplexed in a multiple drive system and those intended to do the multiplexing. The control input signals to the drive to be multiplexed are WRITE GATE, READ GATE, HEAD SELECT 2^0, 2^1, 2^2, 2^3, TRANSFER REQUEST and COMMAND DATA. The signals to do the multiplexing are the DRIVE SELECT 1,2,3. ADDRESS MARK ENABLE (AME) is a control input on the radial data cable and is not multiplexed. The drive select lines accept a binary input combination decoded internally to allow 1–7 drives. Decode 000 is a no select condition. The four Head select lines allow selection of each individual read/write head in a binary coded sequence. Heads are numbered 0 thru 15. Write Gate and Read Gate are control signals which initiate writing and reading of data respectively at the disk.

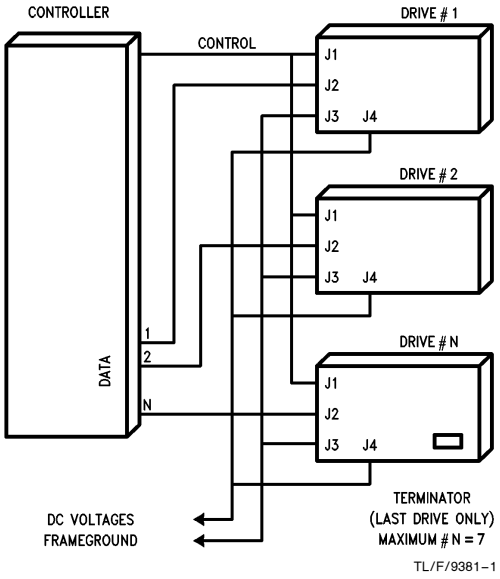


FIGURE 1. Typical Connection in a Multiple Drive System

RRD-B30M105/Printed in U. S. A.

Control Cable		Data Cable	
Pin No.	Signal	Pin No.	Signal
2	Head Select 2^2	1	Drive Selected
4	Head Select 2^1	2	Sector/AMF
6	Write Gate	3	Command Complete
8	Config/Status	4	Address Mark Enable
10	Transfer ACK	7,8	Write Clock ±
12	Attention	10, 11	Read/Reference CLK ±
14	Head Select 2^0	13,14	NRZ Write Data ±
16	Sector/AMF	17,18	NRZ Read Data ±
18	Head Select 2^3	20	Index
20	Index	Ground = 5,6,9,12,15,16,19	
22	Ready		
24	Transfer Request		
26	Drive Select 1		
28	Drive Select 2		
30	Drive Select 3		
32	Read Gate		
34	Command Data		
Ground = All Odd # Pins			

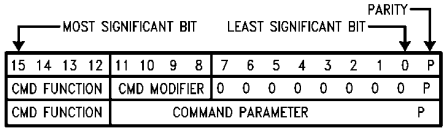
FIGURE 2. ESDI Cable Assignments

The ESDI is an intelligent interface and provides for certain commands, to do certain specific functions, thereby freeing the controller from a number of mundane tasks, and also from being tied to a particular drive. These commands are presented to the drive as 16 information bits of serial data, plus a parity bit. The transfer of this data is controlled by the handshake protocol with the TRANSFER REQUEST and TRANSFER ACKNOWLEDGE signals. Upon receipt of this serial data, the drive will perform the required function as specified by the bit configuration. *Figure 3* lists the various commands supported by ESDI.

The Address Mark Enable (AME) signal behaves differently for soft and hard sectored drives. In soft sectored drives this signal, when active with Write Gate, writes an Address Mark on the disk. When AME is asserted without Write Gate or Read Gate, it causes a search for Address Marks. The address mark written is usually a gap of no flux transitions, exactly 24 bits long. In case of hard sectored drives, the AME does not cause an Address Mark to be written on the media. The trailing edge of AME with Write Gate asserted, initiates the writing of the ID PLO sync field.

The output control signals are driven with an open collector output stage capable of sinking a maximum current of 48 mA. They consist of the DRIVE SELECTED, READY, ATTENTION, INDEX, SECTOR, TRANSFER ACKNOWLEDGE, and CONFIGURATION & STATUS. COMMAND COMPLETE is a control output which allows the host to monitor the drive's command completion status. The ATTENTION line is activated whenever there is an erroneous condition at the drive. In response to the Request Configuration and Request Status commands, the drive provides some status information which is sent to the controller in a

Command Data Word Structure



TL/F/9381–2

Command Data Definition

Command Function Bit 15	14	13	12	Command Function Definition	Command Modifier Applicable Bits 11–8	Command Parameter Applicable Bits 11–0	Status Data Returned to Host
0	0	0	0	Seek	No	Yes	No
0	0	0	1	Recalibrate	No	No	No
0	0	1	0	Request Status	Yes	No	Yes
0	0	1	1	Request Config	Yes	No	Yes
0	1	0	0	Select Head Group*	No	Yes	No
0	1	0	1	Control	Yes	No	No
0	1	1	0	Data Strb Offset*	Yes	No	No
0	1	1	1	Track Offset	Yes	No	No
1	0	0	0	Init Diagnostics*	No	Yes	No
1	0	0	1	Set Bytes/Sector*	No	Yes	No
1	0	1	0	Reserved			
1	0	1	1	Reserved			
1	1	0	0	Reserved			
1	1	0	1	Reserved			
1	1	1	0	Set Config*	No	Yes	No
1	1	1	1	Reserved			
*Optional Commands					All Unused Bits Set to Zero		

FIGURE 3. ESDI Command Structure

serial manner over the CONFIGURATION & STATUS line using a handshake protocol between TRANSFER REQUEST and TRANSFER ACKNOWLEDGE.

Index and sector are interface signals from the drive which indicate the start of a track and sector respectively. In case of a soft sectored drive there are no sector pulses, but if an address mark is found then it signals the end of an address mark, indicated by the AMF signal from the drive.

2.2 Data Cable

All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. Four pairs of balanced signals are used for the transfer of data and clock signals: NRZ Write Data, NRZ Read Data, Write Clock, and Read/Reference Clock. Differential drivers and receivers are needed to interface the controller to the drive, like National's DS8922A/DS8923A. Connection details are shown in *Figure 13*. The NRZ Write Data is clocked by the Write Clock signal while the NRZ Read Data is clocked by the Read Clock signal. The Reference Clock signal from the drive will determine the data transfer rate. The transitions from Reference Clock to Read Clock must be performed without glitches, i.e. the clock should not violate the minimum allowable specifications of the controller chip.

2.3 ESDI Format Rules

The ESDI essentially supports a fixed sector implementation, (Drive hard sectored) and a soft sectored implementation, (Drive soft sectored). The record format on the disk is under control of the controller, however, the ESDI standard recommends a certain format structure which must be implemented. In a hard sectored drive the index pulse signifies the start of a track, while the sector pulse signifies the start of a sector. *Figure 4* shows a fixed sector format and associated timings. In a soft sectored drive, the index signifies the beginning of a track while the beginning of each sector is defined by an Address Mark, followed by the ID field which contains the header information. The AME/AMF handshake is utilized to detect these address marks. *Figure 5* shows the soft sectored format and associated timings. In a hard sectored drive, the beginning of the ID PLO sync field is specified by the trailing edge of the AME, when Write Gate is active. From *Figures 4, 5* and *6* it can be seen that there are some minor differences between the DDC's format and the ESDI recommended format. The ESDI recommended format supports a post index/sector gap field and a write splice field between the ID and DATA fields, which is not directly supported by the DDC. Also Write Gate needs to be optionally deasserted in the write splice area between the ID and DATA segments as shown in *Figures 4* and *5.* These shortcomings can however be overcome through a combination of hardware and software considerations as discussed in the following sections.

3.0 CONTROLLER DESIGN—DISK SIDE

Perhaps of greatest significance to system designers is the fact that the ESDI drives provide the data separation function internally. The performance benefits attained by putting the data separator on the drive more than offset the cost in terms of system efficiency and reliability. Data is transferred over the interface in NRZ format. This results in the use of high density encoding schemes to be implemented in the drive, like 2,7 RLL, etc. These factors greatly simplify the task of designing a controller for ESDI drives. This task is even more simplified with the availability of VLSI disk controller IC's like the DP8466A, which integrates numerous

ESDI Hard Sectored Format

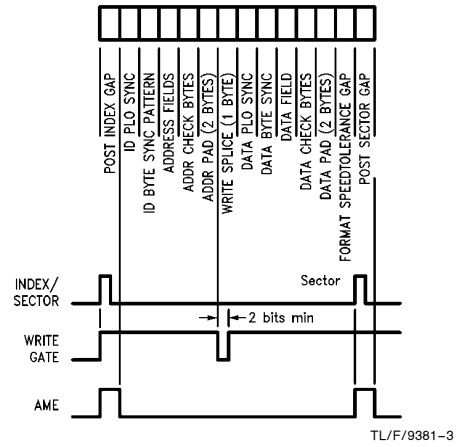


FIGURE 4. ESDI Fixed Sector Format and Relevant Timing

functions in the disk path. The disk side involves the interface of the DDC to a disk interface, like the ESDI. This is made up of two main paths—the data path and the control path. The control path is responsible for the disk related functions like sending commands, etc (as discussed in Section 2.1), while the data path is responsible for the data transfer, (refer to Section 2.2). The DP8466A is a disk data path controller which does not involve itself with the slower tasks of the control path. It features full format programmability, fully programmable ECC, 16-bit dual channel DMA and a 32-byte FIFO. Data is transferred from the FIFO in selectable bursts, which minimizes bus occupancy and can thereby accommodate some degree of latency. For more details refer to AN-413. As mentioned in the previous section, the disk formats suggested by the ESDI standard produce some compatibility problems with the DP8466A, mostly in the area of control line timing with the drive. The techniques to handle them with minimum overhead is discussed below for the two types of ESDI drive systems.

ESDI Soft Sectored Format

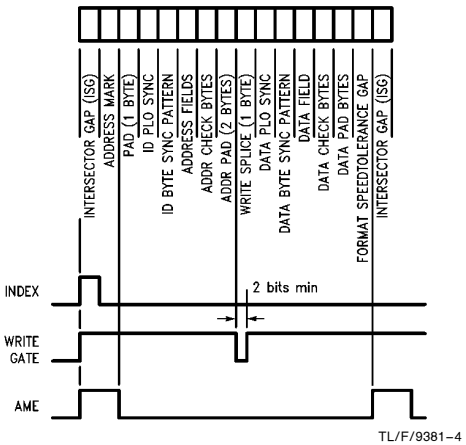


FIGURE 5. ESDI Soft Sectored Format and Relevant Timing

3.1 ESDI Hard Sectored Drive

For an ESDI drive which is hard sectored, the specification calls for an Inter-Sector Gap (ISG) which is to precede and follow the index/sector pulse,also referred to as the post index/sector pulse. This gap is needed to provide the drive with an area for the embedded servo (if used) and also gives the controller time to assert read gate. While formatting the drive, the end of the ISG (post index/sector gap), is indicated by the trailing edge of AME. This is needed by the drive to indicate the beginning of the PLO sync (preamble) field, necessary when the drive encodes the PLO sync field with a non-standard preamble pattern, e.g. as in 2,7 encoding with 3T or 4T preambles. The DDC generates the AME signal with the necessary timing during a format operation. To do so the DDC must be programmed to be in the hard sectored mode and have the start with address mark bit enabled in the Disk Format Register, (HSS and SAM bits in the DF register). *Figure 6* illustrates the manner in which the DDC format parameter registers need to be manipulated for a format operation and a read/write operation. It should be noted that the ISG field is implemented by the DDC's ID preamble field while the PLO sync is implemented by the ID sync #1 field. This feature is needed only during formatting, and hence should be disabled at other times. On the other hand when a disk read/write operation is attempted, the header field is compared (or read) usually. The DDC asserts read gate just after the index or sector pulse, which initiates the data separator to start locking the PLL to the preamble. Since there is the ISG field after the index/sector pulse, the read gate to the drive needs to be delayed until after the ISG. This can be achieved in two ways. One technique is to delay index and sector to the DDC by the length of the ISG, so that the DDC would be asserting Read Gate in the preamble area. *Figure 6* shows the format and control signals for this situation. This is the technique adopted for the combined solution proposed in section 3.3 to implement both hard and soft sectored ESDI.

Alternate Technique

The other technique for implementing the post index sector gap is explained below and could be used when only ESDI hard sectored drives are under consideration. Essentially when formatting the drive, the DDC is set up to use ID preamble field as the ISG, the ID sync #1 as the PLO synch and the ID sync #2 as the sync byte. Hence when reading or writing, the DDC is set up to have ID preamble as the PLO sync, ID sync #1 is skipped and ID sync #2 as sync byte. External hardware is used to delay the read gate at the beginning of every sector, by the length of the ISG so that it gets asserted over the preamble on the drive. It should however be noted that the read gate needs to be delayed only at the beginning of the sector and not in the middle of the sector before the data field. *Figure 7* outlines the format manipulating and control signals behaviour.

3.2 ESDI Soft Sectored Drive

The soft sectored specification of ESDI provides two major stumbling blocks for the DDC. First there is a need for providing the handshaking for the AME and AMF lines. The controller needs to raise AME when it wishes to be notified of the start of a sector. The controller asserts Read Gate on detecting AMF, generated by the drive on finding an Address Mark of 24 bits. After seeing the drive assert AMF, the controller removes AME, finishing the handshake. Since the DCC does not provide the AME/AMF handshake, external hardware is required to do so when the DDC is not formatting. As the Address Mark is at the beginning of every sector, the AMF signal is seen by the DDC as a sector pulse. The DDC is programmed to believe that it is looking at a hard sectored drive, and thereby the soft sectored ESDI can be handled in this pseudo fashion.

The second stumbling block is that the soft sectored ESDI drives generates an AMF signal for each sector on the disk, including the first. Because the DDC starts its operation on either index or sector, (AMF instead of sector in this case), it will believe that there is a sector between index and the first AMF. External circuitry is needed to eliminate the first AMF pulse and replace it with a delayed version of index.

One other problem with soft sectored ESDI, is formatting the drive. In these drives there is usally a ISG following the index pulse. In addition to this is the Address Mark field, which fits between the ISG and the PLO sync fields. The DDC has only two fields (ID preamble and ID sync #1) before the sync byte, with which these three fields have to be created. One way to implement it with external circuitry is to use the ID preamble field of DDC for writing the ISG and AM. The external circuitry delays the AME from the DDC by the length of the ISG. The address mark (AM) field will hence be the length of the ID preamble less the delay and the PLO sync is implemented using the ID sync #1 field. *Figure 8* shows the format manipulation required and the control signals behaviour.

3.3 Combined Solution

A combined solution to the above problems, (for a system supporting both soft and hard sectored ESDI), can be provided by a single PAL® device and something to provide a delay (possibly another PAL device). The interface solutions can be grouped into two main areas, implemented as state machines.

1. Index and sector pulse generation to the DDC
2. AME/AMF handshaking between the DDC and the drive.

3.3.1 The Index/Sector Machine

There are essentially 4 types of drive operations which are encoded using the two PAL inputs (soft/hard/ and format). These are outlined below:

1. Hard sectored drive, read or write data operation
   PAL Inputs: soft/hard/ = 0 and format = 0

In this case the index/sector machine essentially delays both the index and sector pulse from the drive by the length of the ISG (post index/sector gap) and then present it to the DDC. *Figure 10* shows the state diagram and truth table for this part of the index/sector machine. Three state variables are used: F1 and F2 to indicate the state, and G1 to signal if an index pulse has been received from the drive. *Figure 9* shows the state diagram, truth tables and timing relationships for this part of the machine.

2. Hard sectored drive format operation
   PAL inputs: soft/hard/ = 0 and format = 1

In this case the index/sector machine essentially lets the index and sector pulses from the drive flow through to the DDC.

3. Soft sectored drive, read or write operation
   PAL inputs: soft/hard/ = 1 and format = 0

In this case the index/sector machine essentially follows the AMF from the drive, translating it as the sector input to the DDC; however if an index pulse occurs from the drive then it waits for an AMF from the drive and then generates an index and no AMF/sector pulse to the DDC. This is the scenario which happens in the sector just after the index

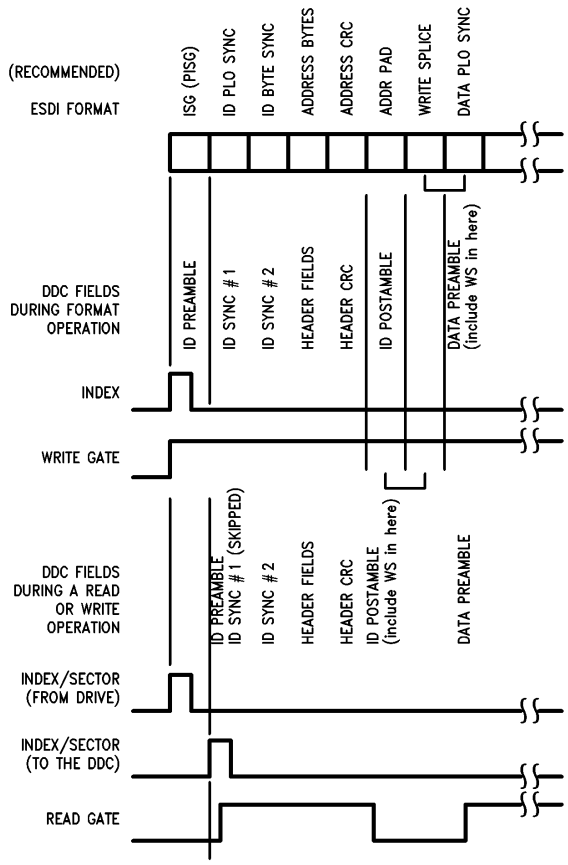


FIGURE 6. Programming the DDC Format Parameters in the Case of an ESDI Hard Sectored Implementation

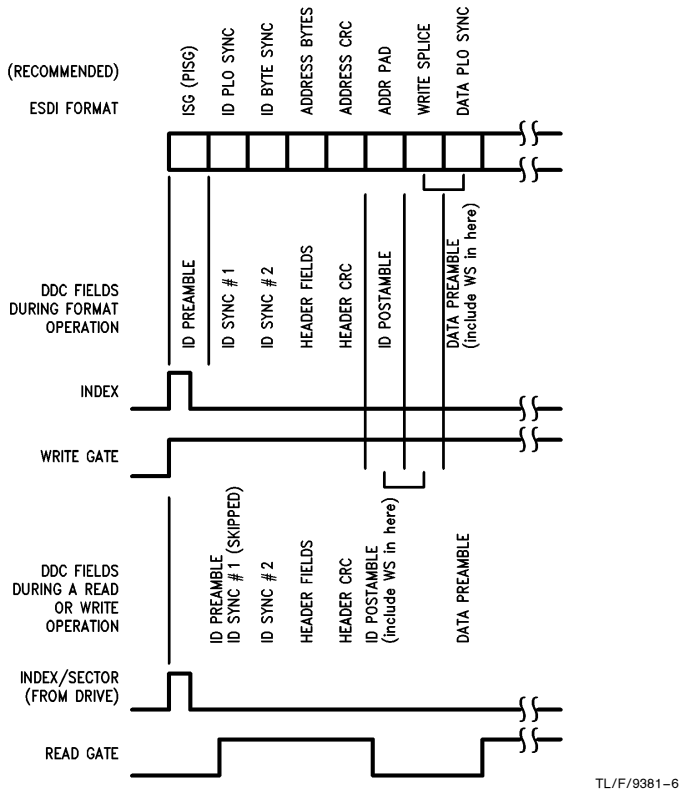


FIGURE 7. Programming the DDC Format Parameters in the Case of an ESDI Hard Sectored Implementation (Alternative Technique)

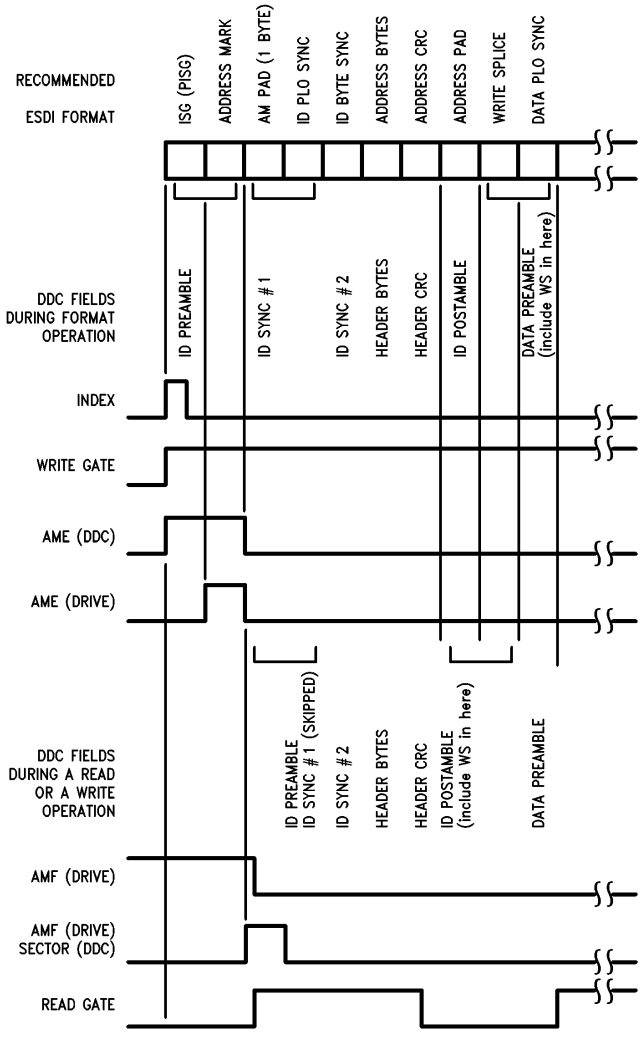
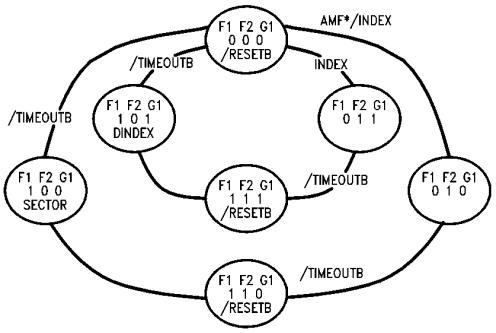


FIGURE 8. Programming the DDC Format Parameters in the Case of an ESDI Soft Sectored Implementation

State Machine

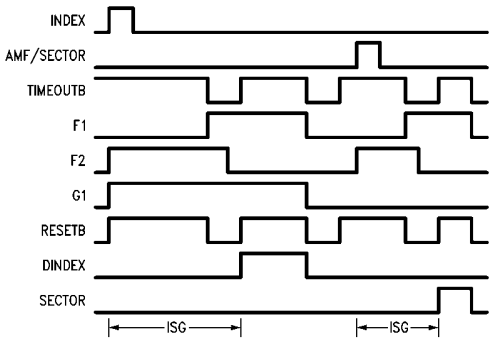


TL/F/9381–8

Truth Table

F1	F2	G1	Index	AMF	TIMEOUTB	F1	F2	G1	RESETB	DINDEX	Sector
Present State			Inputs			Next State			Outputs		
0	0	X	0	0	X	0	0	0	0	0	0
0	0	0	1	X	X	0	1	1	1	0	0
0	0	0	0	1	X	0	1	0	1	0	0
0	1	0	X	X	1	0	1	0	1	0	0
0	1	1	X	X	1	0	1	1	1	0	0
0	1	0	X	X	0	1	1	0	0	0	0
0	1	1	X	X	0	1	1	1	0	0	0
1	1	0	X	X	X	1	0	0	1	0	0
1	1	1	X	X	X	1	0	1	1	1	0
1	0	0	X	X	1	1	0	0	1	0	1
1	0	1	X	X	1	1	0	1	1	1	0
1	0	X	X	X	0	0	0	0	1	0	0

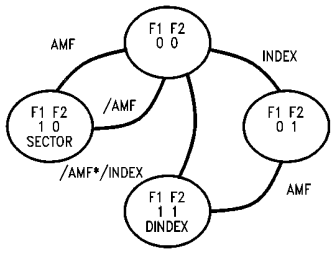
Index/Sector Machine Timing Diagrams



TL/F/9381–9

FIGURE 9. Index/Sector Machine Hard Sectored ESDI (Non-Format) State Diagram, Truth Tables and Timing

State Diagram



TL/F/9381–10

Truth Table

F1	F2	Index	AMF	F1	F2	DINDEX	Sector
Present State		Inputs		Next State		Outputs	
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	X	0	0	1	0	0
0	1	X	1	1	1	1	0
1	1	1	X	1	1	1	0
1	1	X	1	1	1	1	0
1	1	0	0	0	0	0	0
0	0	X	1	1	0	0	1
1	0	X	1	1	0	0	1
1	0	X	0	0	0	0	0

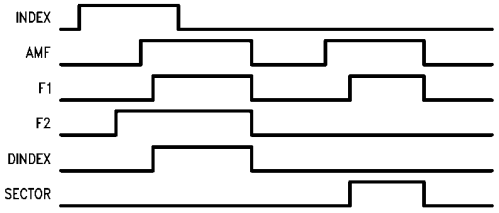
hole. *Figure 10* shows the state diagram, truth table and timing relationships for this section of the Index/Sector machine. It uses two state variables, F1 and F2.

4. Soft sectored drive, format operation
   PAL inputs: soft/hard/ = 1 and format = 1

In this case the index/sector machine essentially follows the index pulse from the drive and passes it on to the DDC while it forces the sector input to the DDC to be inactive (low).

3.3.2 The Address Mark Machine

The Address Mark Machine consists of a pair of multiplexers which feed the AME input to the drive and translate the AMF from the drive as the sector pulse input to the DDC. Once again as before there are 4 types of drive operations, encoded using the two PAL inputs soft/hard/ and format.



TL/F/9381–11

FIGURE 10. Index/Sector Machine Soft Sectored ESDI (Non-Format) State Diagram, Truth Table and Timing

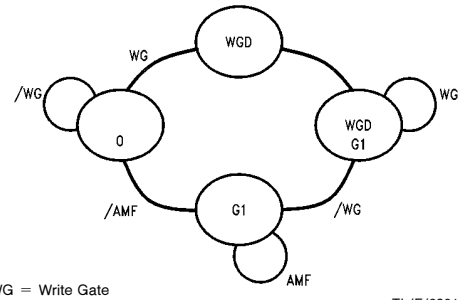
The working of the address mark machine in these cases is outlined below.

1. Hard sectored drive, Read and Write operation
   PAL inputs: soft/hard/ = O and format = O.

The AME to the drive is kept inactive (low).

2. Hard sectored drive, format operation
   PAL inputs: soft/hard/ = O and format = 1

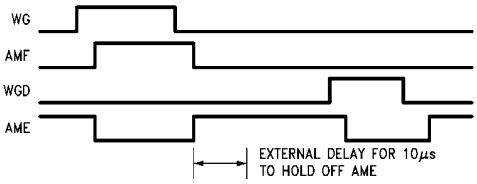
State Diagram



WG = Write Gate
WGD = Write Gate Delayed by One Clock

TL/F/9381–12

Timing Diagrams—Soft Sectored (Non-Format) Operation



TL/F/9381–13

In this case the AME to the drive follows the DDC generated AME.

3. Soft sectored drive, read or write operation
   PAL inputs: soft/hard/ = 1 and format = O

In this case the PAL generates AME and handshakes with the AMF from the drive, translating it to the DDC as the sector input.

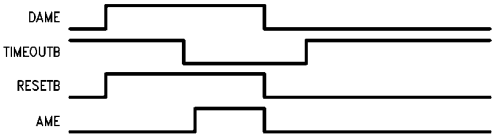
4. Soft sectored drive format operation
   PAL inputs: soft/hard/ = 1 and format = 1

In this case the leading edge of AME generated by the DDC is delayed by the length of the ISG (post index/sector gap) and presented to the drive.

Note: The trailing edge is not delayed.

Figure 11 shows the state diagram and timing relationships for the Address Mark Machine. Given below are the PAL equations for this control PAL implemented in a 16R4. These include some simplifications from the above information. In particular the hard sector non-format equations for F1, F2 and G1 can drop the use of the term /FORMAT, since during the format operation it is quite acceptable to have these output behaving as for non-format since the index to the DDC follows the index from the drive. The equations are written in PLAN format.

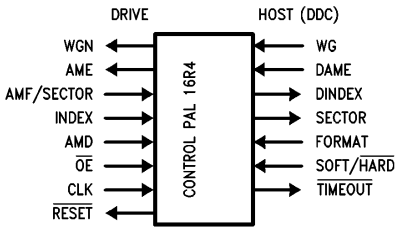
Timing Diagrams—Soft Sectored Format Operation



TL/F/9381–14

FIGURE 11. Address Mark Machine Details

Control PAL Pin Description



TL/F/9381–15

Control PAL Equations (Plan Format)
Version 1.3 Dated: 05/12/86

```
/wgd := /wg

gl    := (soft*wgd*wg)+(soft*wgd*gl)+
         (soft*/wgd*/wg*amf)+
         (/soft*/gl*index)+(/soft*gl*/fl)+
         (/soft*gl*f2)+(soft*gl*/timeout)

fl    := (soft*/format*f2*amf)+
         (soft*/format*fl*ff2*index)+
         (soft*/format*fl*/f2*amf)+
         (soft*/format*/fl*/f2*/index*amf)+
         (/soft*/fl*f2*timeout)+(/soft*fl*f2)+
         (/soft*fl*/f2*/timeout)

f2    := (soft*/format*/fl*/f2*index)+
         (soft*/format*/fl*f2)+
         (soft*/format*fl*f2*amf)+
         (soft*/format*fl*f2*index)+
         (/soft*/fl*/f2*index)+
         (/soft*/fl*/f2*amf)+
         (soft*/fl*f2)

if (vcc) /ame   = (soft*/format*gl)+
                  (soft*/format*wg)+
                  (soft*format*/dame)+
                  (soft*format*/timeout)+
                  (/soft*/format)+
                  (/soft*format*/dame)+(amd)

if (vcc) reset  = (soft*/format)+
                  (soft*format*/dame)+
                  (/soft*/format*/fl*/f2)+
                  (/soft*/format*fl*f2)+
                  (/soft*format)

if (vcc) /dindex =(soft*/format*/fl)+
                  (soft*/format*/f2)+
                  (format*/index)+
                  (/soft*/format*/fl)+
                  (/soft*/format*f2)+
                  (/soft*/format*/gl)+
                  (/soft*/format*timeout)

if (vcc)/sector  =(/format*/fl)+
                  (soft*/format*f2)+
                  (soft*format)+
                  (/soft*/format*f2)+
                  (/soft*/format*gl)+
                  (/soft*/format*timeout)+
                  (/soft*format*/amf)
```

3.3.3 Timer PAL

This is essentially the PAL used to generate the delay. When the RESET input to the PAL is low it does not count. It starts counting and when the desired time delay is reached it produces an active low output ($\overline{\text{TIMEOUT}}$). The value of the delay is variable from 1 to 32 byte times. The counter is clocked by the disk's RCLK which is divided by 8 to provide a byte-rate clock to a five-stage counter. When $\overline{\text{RESET}}$ is low (active), the inverse of the values GA–GE are loaded into the counter, and one byte-time after the counter reaches 11111 the TIMEOUT output goes low and counting stops. The value on GA–GE (GA is least significant bit) should be set to (number of bytes delay—1). Given below are the equations for this PAL 20X10). Note that PLAN requres the use of a dummy term to complete the OR function before the XOR. Hence for example, in the equation for F1 the term ''F1*/F1'' is put in, which always equals to 0.

Timer PAL Equations (Plan Format)
Version 1.3 Dated: 05/12/86

```
f3 := (/timeout*/reset)+
      (f3*/f3):+:(f3*/reset)

f2 := (/timeout*/reset*f3)+
      (f2*/f2):+:(f2*/reset)

fl := (/timeout*/reset*f2*f3)+
      (fl*/fl):+:(fl*/reset)

qa := (/reset*f3*f2*fl)+
      (reset*ga):+:(qa*/reset)

qb := (/reset*f3*f2*fl*/qa)+
      (reset*gb):+:(qb*/reset)

qc := (/reset*f3*f2*fl*/qa*/qb)+
      (reset*gc):+:(qc*/reset)

qd := (/reset*f3*f2*fl*/qa*/qb*/qc)+
      (reset*gd):+:(qd*/reset)

timeout :=
   (/reset*f3*f2*fl*/qa*/qb*/qc*/qd*/qe)+
   (timeout*/timeout):+:(timeout*/reset)
```

3.3.4 Some Other Timing Considerations

The ESDI specification imposes some additional timing restrictions which have to be accommodated for with external logic. These are outlined below:

1. AME to the drive cannot be asserted till at least 10 μs, after deassertion of Write Gate to the drive.
2. In the PAL solution discussed above, when the drive drops AMF at the end of the handshake, the PAL reasserts AME. The MAXTOR doesn't seem to like that and hence the AME to the drive must be held off for at least 8 μs, from the trailing edge of AMF.
3. On a similar token, the MAXTOR drive doesn't like AME to be active when Read Gate is asserted, (it usually activates ATTENTION on the drive). Hence AME needs to be held off at least 8 μs from trailing edge of Read Gate. This is accomplished using a mono shot, which generates a disable signal (Address Mark Disable, AMD), for 10 μs and while this is active the control PAL disables AME to the drive. Also since we are presenting the drive like a hard sectored one to the DDC, we could safely assume that AME need not be active with Read Gate. This is accomplished by gating the AME to the drive with the read gate. Hence if read gate is active, AME gets disabled to the drive. These could be incorporated within the control PAL if desired.

3.4 Handling the Optional ESDI Format Specifications

The ESDI has certain optional specifications which are not directly supported by the DDC. These are discussed below with explanations of the way they can be implemented.

3.4.1 Optional Deassertion of Write Gate between the ID and DATA Fields in a Format Operation

This option is not supported by the DDC, as once its starts the format operation, the Write Gate remains asserted for the entire track. The purpose of the deassertion of Write Gate between the ID and Data fields is usually to indicate to the 2,7 RLL encoder, the start of the data preamble. This enables the encoder to substitute the 3T or 4T preamble pattern. This feature could be implemented in two ways. Using external logic, the trailing edge of the SDV (Serial Data Valid) is used to gate the logic, count until the header postamble has been written and then force the Write Gate low for the required two bit times. This problem could also be circumvented in software by incorporating a two pass format. The first pass involves a regular format operation which will write the headers for all the sectors, but since there will be no de-assertion of Write Gate before the data fields, the proper data preamble will not be written. In the second pass a compare header-write data operation is done, where the Write Gate edge is used to initiate the drive generated preamble. It should be noted (as also pointed out in the ESDI specification), that this is necessary only if a read will be attempted after a format.

3.4.2 Handling the Write Splice Field between the ID and DATA Fields in the ESDI Format

The ESDI format specification recommends a 2-byte header postamble and a 1-byte write splice. The DDC does not have a separate field to implement the write splice. It is accomplished by software manipulation as follows. The format is programmed to have a 2-byte header postamble and a data preamble which is one byte longer than the desired length. This byte is taken as the write splice (a floating Byte). During a write operation, this floating byte is considered as part of the data preamble. As Write Gate is asserted 3 RCLKS into the data preamble, the ''write splice'' associated with Write Gate assertion, due to write driver turn on time, etc, occurs during the 1 byte of the data preamble which is the floating byte. When the sector is being read, this byte is attached to the header postamble. Since Read Gate is reasserted 11.5 RCLKS into the data preamble, this ensures that it doesn't get asserted in the splice. From the above data it can be concluded that for a normal operation in the DDC, Read Gate and Write Gate assertion in the data field are separated by 8.5 RCLKS in the data preamble, hence automatically taking care of the write splice as the first byte of the data preamble.

3.5 Critical Read and Write Parameters

There are a number of drive dependent parameters which must be met in order to ensure proper operation with an ESDI drive. These are summarized below, for consideration during actual design.

3.5.1 Read Function Parameters

1. A read operation may not be initiated until 15 μs following head switch.
2. Read Gate may not be asserted during a write splice or within ± 1 bit time of a write splice.
3. Read gate must be asserted within 16 bit times from the write gate assertion point when the current field was written.
4. Data (read) at the interface could be delayed by up to 9 bit times from the data recorded on the disk media.
5. RCLK and RDATA are valid within the number of PLO sync field bytes specified by the drive configuration after read enable and a PLO sync field is encountered.

3.5.2 Write Function Parameters

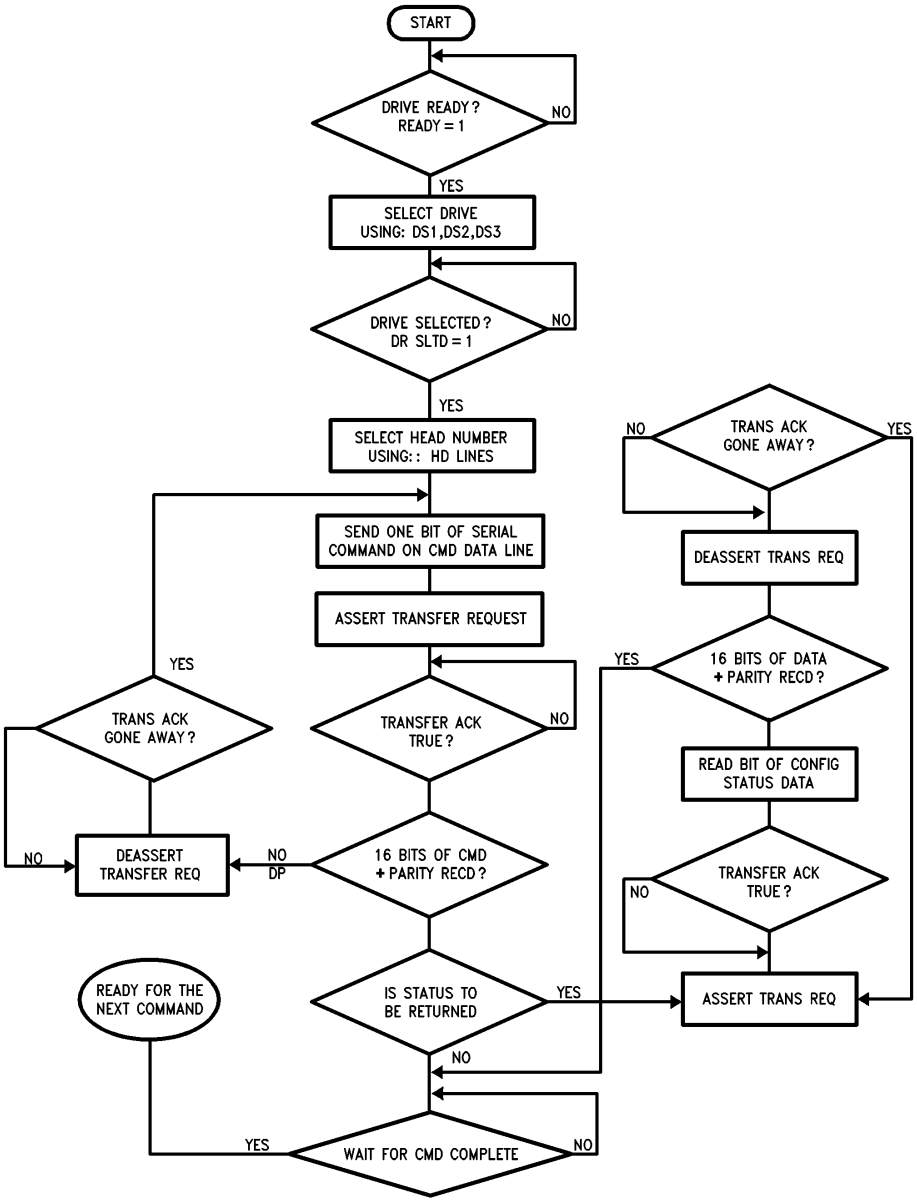
1. Assuming head selection is stabilized, the time lapse from deassertion of Read Gate to assertion of Write Gate shall be five reference clock periods minimum.
2. Write Clocks must precede Write Gate by a minimum of two and a half Reference Clock periods.
3. Write driver plus data-encoder turn on time (write splice width) is between 3 and 7 reference clock periods.
4. To account for data-encoding delays, write gate must be held on for at least 2 byte times after the last bit of information to be recorded. This implies a minimum data postamble length of 2 bytes.
5. The time lapse before Read Gate or AME can be asserted after deassertion of Write Gate is defined by the ''ISG bytes after index/sector'' in the configuration data response (10 μs).
6. Write Gate must be deasserted at least 1 μs before a head change and shall not be asserted untl 15 μs after a head change or command complete.
7. Write data received at the I/O connector will be delayed by the encoder by up to 8 bits maximum prior to being recorded on the media.

Some of these parameters were accommodated in the combined PAL solution, while others need to be accommodated in firmware. The above discussion covered the hardware and relevant firmware considerations for designing the disk side data path section of the controller subsystem. The other aspect of the disk side design is the control path which is discussed in the following sections.

3.6 Disk Side Control Path Design

Since the DDC is a data path controller, the control path functions of the drive have to be controlled by the local intelligence in the disk controller subsystem. This offers more versatility and is less of a handicap, for it allows the DDC to be used with any of the disk interface standards. Also the control path functions like seeks, etc, are very slow operations and by not handling them, the DDC is able to achieve maximum operating speeds of 25 Mbits/sec. Any μP or μC with simple I/O ports would suffice for the control path functions. This usually is no extra overhead, because a local μP or μC is necessary to handle the protocol over the system bus and accordingly set up and activate the DDC anyway.

The control signals are sent over the drive's ''A'' control cable. Details of the respective signals are discussed in section 2.1. These signals are sent/received, to/from the drive through industry standard open collecter drivers (DP8311) and receivers (74ALS240), with appropriate terminations in accordance with the ESDI specification. These are also shown in the complete design, schematic *(Figure 13).* In the system design under consideration, the control path functions are controlled by the NSC800, through the NSC810 programmable I/O port. The ideal flow of operations for the control path is shown in *Figure 12* and is self-explanatory.



TL/F/9381–16

FIGURE 12. Control Path Flow of Operations

The above discussion covered the hardware and relevant firmware considerations for designing the disk side of the disk controller subsystem. The other aspect of the disk controller subsystem design is the system side. This essentially involves the interface of the DDC to the local buffer memory and interface of this local buffer over the system bus to the host system. Discussion of the system side design follows in the next section.

4.0 CONTROLLER DESIGN—SYSTEM SIDE

4.1 System Side Hardware Considerations

The system side architecture in controller subsystems usually consists of two buses. A local bus accommodates the DDC, local microprocessor or microcontroller and the buffer memory while the remote bus is usually a standard bus like VME, STD100, MULTIBUS®, SCSI, etc, which connects the controller to the host system. The buffer memory on-board is a disk buffer which could have a maximum size of 64k. The DDC supports two 16-bit DMA channels, local and remote DMA. The local DMA transfers the data between the on-board FIFO and local buffer memory, while the remote DMA transfers data between the local buffer memory and the host over the system bus.

In the controller subsystem under consideration *(Figure 13),* the local μP is the NSC800™. This can access the registers of the DDC in the peripheral mode and executes program code from an EPROM. All the peripheral chips are memory mapped and the 74HC138 decodes the address lines to generate the various chip selects. A PAL has been designed to provide arbitration between the DDC and the NSC800 for use of the local bus. This design was intended to provide a good exerciser for the DP8466 and hence a terminal is connected to the μP (NSC800) through a UART (NSC858). A monitor has been developed which allows exercise of the DDC in a lot of modes and interacts with the user through the terminal. Hence the remote DMA is not really used. Also in order to be able to do both byte-wide and word-wide transfers, a detection logic was implemented. Only an 8k local buffer RAM is used, accommodating 8k bytes or 8k words. LEDs are provided for visual indications of certain disk parameters.

As mentioned above, from a general design point of view, this could be easily extended to a standard system bus like VME, MULTIBUS II, etc, using the remote DMA channel of the DDC and using the local microprocessor to handle the communications protocol to the DDC and to the host operation system over the system bus.

4.2 System Side Programming Considerations

The firmware in conjunction with the microprocessor is essentially responsible for deciphering the protocol sent over the system bus and then based on the requested operations, set up the control path through the I/O ports and set up the DDC to initiate the disk operation. The driver routines to handle the control path are usually very simple and need to do the defined task as outlined in Section 3.6. The driver would consist of various blocks to implement the different ESDI commands and interpret the status reported; however, the flow would be as shown in *Figure 12*. Once the drive is positioned at the right track and ready to start the operation, control is handed over to the DDC. Before the DDC can be instructed to initiate a disk operation it has to be prepared, i.e., all registered set up to achieve the desired task.

Figure 14 shows the sequence of actions to be done with parameter RAM is initialized with the pattern and count values in accordance with the format desired. The DMA registers should be set up next starting with the LT and RT registers followed by the DMA Address Byte #, indicating start boundaries in memory and finally the sector byte count # and remote data byte count # registers. The ECC registers (preset, taps and control) are set up and the DF register is set up accordingly. Having set up the various registers, the format registers are manipulated if desired, sector counter, NSO counters initialized, and DC register loaded to initiate a disk operation. The interrupts are monitored with reads of the status register to determine the results of the disk operation. If desired, certain fancy operations like error correction cycle, interlock mode of operations, FIFO table formatting could be implemented with additional firmware.

An effort has been made in this application note to introduce the designer to the ESDI interface, and explain the intricacies of designing to its specifications with the DDC. The emphasis was on the disk side, as system side requirements may vary with different design situations. A representative design has been included which was built and tested at National (shown in *Figure 13*). For more details on the ESDI standard, refer to the official specifications provided by the ESDI committee. For more details on the DP8466A refer to AN-413 and the data sheet.

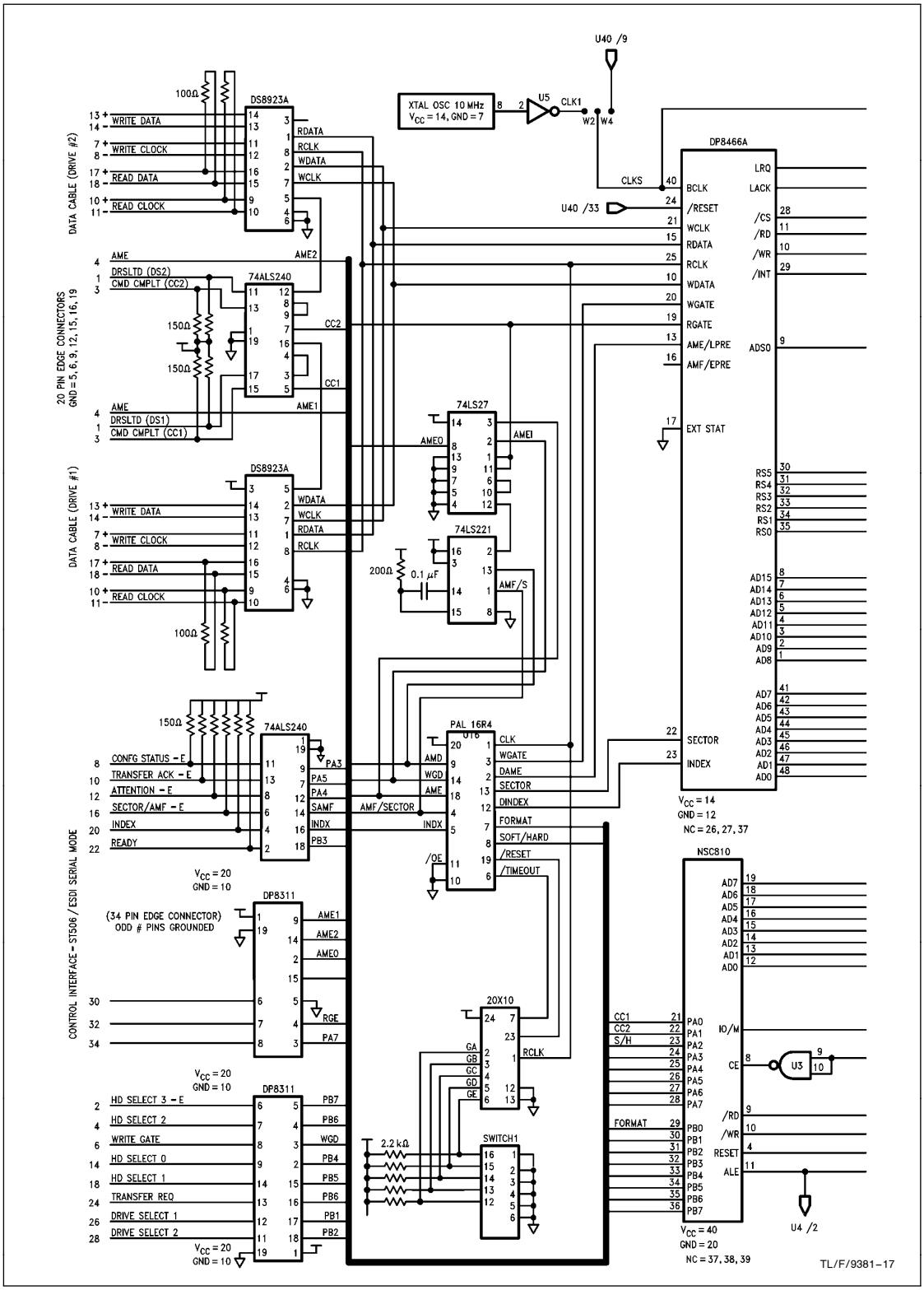
APPENDIX
Schematic of a Representative Design of an ESDI Disk Controller

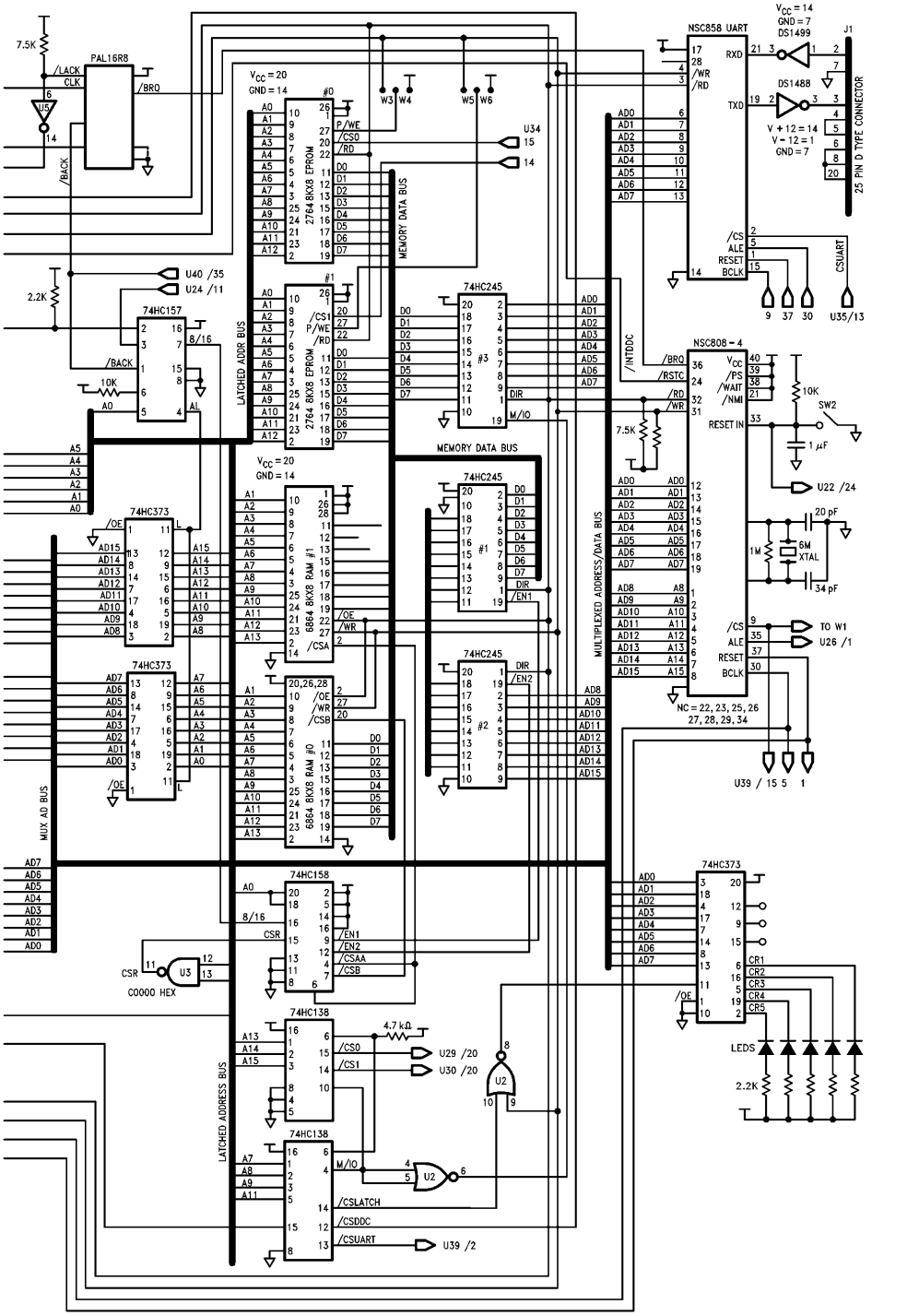
Shown in the attached schematic is the design of an ESDI disk controller system. Emphasis has been placed on the disk side of the design, as that is of foremost concern with respect to the DDC. The design incorporates the PALs as described in the previous sections, to implement the shortcomings of the DDC, in order to completely support the Enhanced Small Device Interface specifications.

The control signals are generated using the NSC810 programmable I/O port. They are connected to the ''A'' cable through industry standard open collector drivers, (DP8311) and receivers, (74ALS240). The receivers support a termination of 150Ω. The Data path signals are directly controlled by the DDC. They are connected to the ''B'' cable through industry standard differential line drivers and receivers (DS8923A). The appropriate terminations required are shown in the schematic. The design supports two drives and the appropriate data cable is enable by the drive selected line as shown.

The local microprocessor used is the NSC800, which is responsible for controlling the NSC810 to generate the appropriate control signals, and also programming the DDC to initiate the desired disk operation. This design is essentially a demonstration system and hence supports a UART (NSC858), through which the user is allowed to interact through a special monitor to initiate specific operations for the DDC. The memory design has been done so that the system could be operated in both the byte-wide transfer mode and also the word-wide transfer mode. The arbitration PAL is a simple one to arbitrate the bus between the DDC's DMA activity and the microprocessor accesses of the DDC.

It should be noted that the DDC has the capability of two 16-bit DMA channels, local and remote. The local DMA is used to transfer data between the FIFO and the local buffer memory, while the remote DMA is used to transfer data between the local buffer memory and the host system over a system bus like VME, MULTIBUS, SCSI, etc. Hence the design could easily be extended to these situations, if desired.


DP8466A
DS8923A
74ALS240
74LS27
74LS221
PAL 16R4
U16
DP8311
NSC810
20X10
SWITCH1
XTAL OSC 10 MHz
VCC = 14, GND = 7
U40 /9
U40 /33
U4 /2
U3
U5
CLK1
W2
W4
CLKS
DATA CABLE (DRIVE #2)
DATA CABLE (DRIVE #1)
20 PIN EDGE CONNECTORS
GND = 5, 6, 9, 12, 15, 16, 19
CONTROL INTERFACE – ST506 / ESDI SERIAL MODE
(34 PIN EDGE CONNECTOR)
ODD # PINS GROUNDED
100Ω
150Ω
200Ω
0.1 μF
2.2 kΩ
WRITE DATA
WRITE CLOCK
READ DATA
READ CLOCK
AME
DRSLTD (DS2)
CMD CMPLT (CC2)
DRSLTD (DS1)
CMD CMPLT (CC1)
AME2
AME1
CC2
CC1
RDATA
RCLK
WDATA
WCLK
CONFG STATUS – E
TRANSFER ACK – E
ATTENTION – E
SECTOR/AMF – E
INDEX
READY
PA3
PA5
PA4
SAMF
INDX
PB3
AMF/SECTOR
AMD
WGD
AME
AMEO
AMEI
AMF/S
CLK
WGATE
DAME
SECTOR
DINDEX
FORMAT
SOFT/HARD
/RESET
/TIMEOUT
/OE
GA
GB
GC
GD
GE
RCLK
RGE
PA7
PB7
PB6
PB4
PB5
PB1
PB2
HD SELECT 3 – E
HD SELECT 2
WRITE GATE
HD SELECT 0
HD SELECT 1
TRANSFER REQ
DRIVE SELECT 1
DRIVE SELECT 2
VCC = 20
GND = 10
VCC = 14
GND = 12
NC = 26, 27, 37
VCC = 40
GND = 20
NC = 37, 38, 39
LRQ
LACK
/CS
/RD
/WR
/INT
ADSO
RS5
RS4
RS3
RS2
RS1
RS0
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
BCLK
/RESET
WCLK
RDATA
RCLK
WDATA
WGATE
RGATE
AME/LPRE
AMF/EPRE
EXT STAT
SECTOR
INDEX
CC1
CC2
S/H
FORMAT
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
IO/M
CE
RESET
ALE
TL/F/9381–17


PAL16R8
/LACK
CLK
/BRQ
/BACK
7.5K
2.2K
10K
74HC157
8/16
74HC373
74HC158
74HC138
74HC245
NSC858 UART
NSC808-4
DS1488
25 PIN D TYPE CONNECTOR
2764 8KX8 EPROM
6864 8KX8 RAM
LATCHED ADDR BUS
MEMORY DATA BUS
MUX AD BUS
MULTIPLEXED ADDRESS/DATA BUS
LATCHED ADDRESS BUS
/INTDDC
SW2
1 µF
20 pF
6M XTAL
1M
34 pF
LEDS
4.7 kΩ
C0000 HEX
TO W1
U26 /1
U22 /24
U34
U40 /35
U24 /11
U29 /20
U30 /20
U39 /2
U39 / 15 5 1
CSUART
/CSLATCH
/CSDDC
/CSUART
NC = 22, 23, 25, 26
27, 28, 29, 34

TL/F/9381–18

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408