WD1050 SMD Controller/Formatter Application Notes

INTRODUCTION

Prior to the introduction of 5¼ and 8 inch Winchester disks drives in the late 1970's, minicomputers and mainframes were the only systems that utilized rigid disks. These drives were relatively expensive; sometimes as high as $200 per megabyte. They offered the minicomputer designer a fixed or removable drive with capacities from 10 to 300 megabytes. Initially, there was no need for interface standards. IBM Corporation was the predominant leader in the marketplace, and anyone else who decided to build drives were IBM compatible units. But as competition increased, more and more companies began producing lower cost units with increased capacity. Minicomputer companies were being formed, offering complete systems that were non-IBM compatible. The disk drive race was on.

In order to standardize a common interface and to prevent product obsolescence, Control Data Corporation developed an intelligent interface called the Storage Module Device or SMD. This interface allowed a variety of drives to use the same hardware signals, even though their capacities and physical sizes differed. Variations of the SMD were also introduced. Some of these are the CMD (Cartridge Module Drive) and the MMD (Memory Module Drive). The SMD interface began to gain acceptance in the marketplace as competitive manufacturers offered "SMD-compatible" drives as well. The SMD was well on its way to becoming a defacto standard in the industry. Its longevity has been proved by over 10 years worth of product based on this "intelligent" interface.

With today's smaller diameter low cost drives, where does SMD stand? Oddly enough, the higher capacity 5¼ and 8 inch Winchesters are reviving the SMD protocol. Because the SMD interface offers several advantages over the ST506 type interface in the high capacity arena (such as parallel seek instead of serial step pulses), several manufacturers are planning to offer the SMD on their traditional small system disk drives. The SMD, however, is not a trivial interface when it comes down to designing a controller.

A LOOK AT THE SMD

Figure 1 illustrates the electrical signals of the SMD. Two separate cables are used: one for control and one for data. The control cable (commonly referred to as the "A" cable) is responsible for all head movement, status reportings and issuing commands. The data cable (or "B" cable) is used for reading and writing NRZ data to a particular sector on the drive. Note that all lines on both cables are differential signals; they require a differential driver/receiver at both ends.

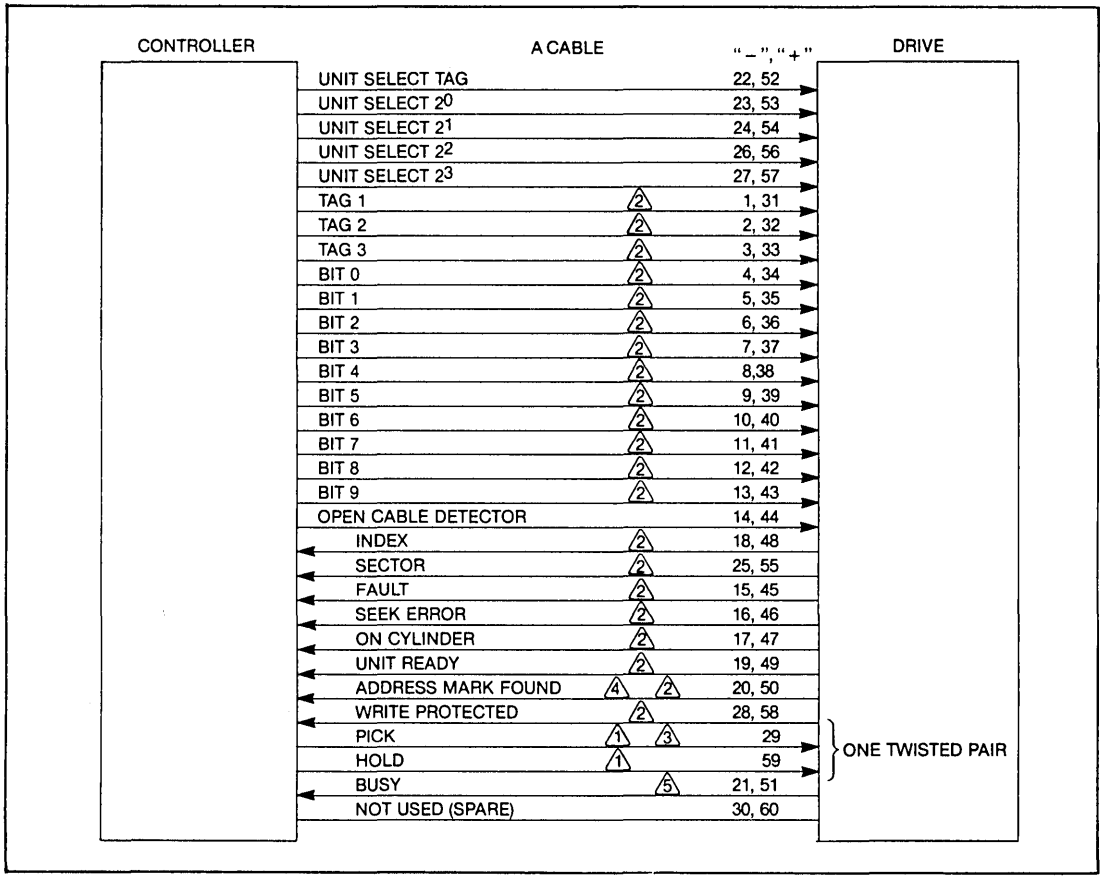
Primary control over the "A" cable is based upon a 10 bit bus called the Tag Bus. These 10 lines send particular information to the driver and initiate a command. Three Tag lines (Tag 1-3) are used to tell the drive what the bus contains during the strobing of the Tags. For example, Tag 1 tells the drive that the Tag bus contains a cylinder number that the head assemblies should be moved to for reading or writing. Tag 2 tells the drive the Head/Volume to select, while Tag 3 is used to initiate read or write commands and to perform special recovery routines.

Drives are selected by separate UNIT SELECT lines on the "A" cable, which have their own strobe line called Unit Select Tag. Other signals on the "A" cable serve status reporting type functions. SEEK ERROR and ON CYLINDER are examples of status lines.

The "B" cable is used to transmit serial, NRZ data to and from the drive. Associated with the R/W lines are clocks: Write Clock for write recovery and Read Clock for read recovery. Additional signals aid in determining the status of each drive on the bus.

In a multiple drive configuration, the two cables are connected as shown in Figure 2. The "A" cable is daisy-chained; each drive is tied together in parallel with termination resistors on the last drive. The "B" cable is radial-connected; a separate cable from each drive connects to the controller.

It is probably obvious by now that a great deal of control is necessary to perform even a simple Read or Write operation on the SMD Bus. The drive controller must perform simultaneous operations on both cables, as well as monitoring status signals to determine successful execution of operation. A typical SMD controller can consist of 150 SSI/MSI Integrated Circuits and a local microprocessor or bit-slice to perform the necessary functions. SMD controller designers of today can take advantage of a new LSI chip that will reduce the number of I.C.'s to well under 40.



NOTE: 60 Position
30 Twisted pair-straight flat cable
Maximum Length-100 ft. (30.48 meters)

1 Special signal, not a balanced transmission signal
2 Gated by unit selected
3 Not interpreted, is Daisy chained, no driver connection within the LMD
4 Not activated, is Daisy chained, always a logic zero output if unit is selected
5 Not generated, is Daisy chained, no driver connection within the LMD

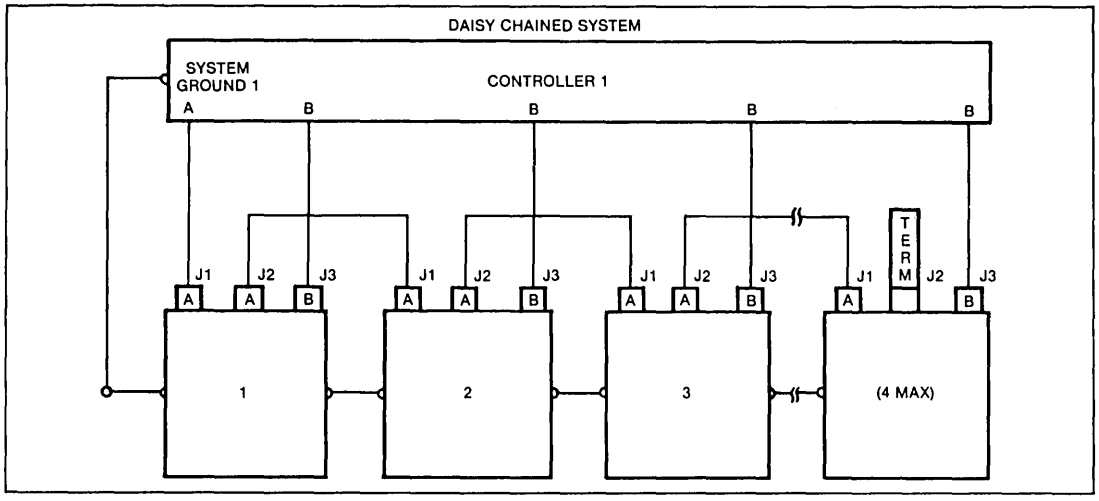
Figure 1(A). Tag Bus I / O Interface ("A" Cable)

CONTROLLER	"B" CABLE	"−","+"	DRIVE
	WRITE DATA	8, 20	
	GROUND	7	
	WRITE CLOCK	6, 19	
	GROUND	18	
	SERVO CLOCK	2, 14	
	GROUND	1	
	READ DATA	3, 16	
	GROUND	15	
	READ CLOCK	5, 17	
	GROUND	4	
	SEEK END	10, 23	
	UNIT SELECTED	22, 9	
	GROUND	21	
	INDEX	12, 24	
	GROUND	11	
	SECTOR	13, 26	
	GROUND	25	

NOTES:
1. 26 conductor flat cable.
   Maximum Length–50 ft. (15.24 meters)
2. No signals gated by "A" cable unit select

Figure 1 (B). "B" Cable Interface

NOTES:
1. Maximum individual A cable lenghts = 100 feet (30.48 meters)
2. Maximum individual B cable lenghts = 50 feet (15.24 meters)



NOTES:
1. Termination of "A" cable lines are required at controller and the last unit of the Daisy chain or each unit in a radial configuration.
2. Termination of "B" cable receiver lines are required at the controller and are on the unit' receiver cards.
3. Maximum cumulative "A" cable length per controller = 100 feet (30.48 meters) maximum individual "B" cable lenght = 50 feet (15.24 meters).

Figure 2. Daisy Chained System

WD1050 SMD CONTROLLER CHIP

Western Digital Corporation offers an LSI controller chip for the SMD protocol. This device, called the WD1050, has been designed to interface an SMD rigid disk drive to a 16-bit Host processor. A set of macrocommands allows the Host to request a specific operation such as seek, read, etc., in which all Tag and control lines on the drive interface perform their appropriate signaling. By using this device, the designer is free to concentrate on operating system software intervention, rather than meeting electrical requirements of the drive protocol. Figure 3 shows the Block Diagram of the WD1050. Data or commands are entered in 16-bits through the Data I/O Buffers. This information is stored in the Task File and tells the device parameters about a specific command. This could be a cylinder address, a sector number to search for, a particular drive that should be selected, etc. After this information is loaded, a command is issued. The Control Unit instructs the various pins on the drive interface to generate their proper signals. Upon completion of a command, the WD1050 interrupts the Host and reports via the status register if any errors were encountered. The device is then ready for the next command.

Figure 4 illustrates the Task File and its contents. The Host processor generates the three address lines shown, then performs a read or write operation to the selected 16-bit register. All registers can be read or written to with the exception of the Command/Status Register. Since both of these registers share a common address location, a "write" will cause a command to execute, while a "read" will cause the status to be fetched from the device. This memory mapped architecture allows the Host to randomly access any location in the Task File without disrupting or reloading the data in other registers.

The Instruction Set of the WD1050 is shown in Figure 5. Return to Zero, and Seek Cylinder commands are used for head movement, while the remaining commands are responsible for reading or writing data. Each Read/Write command also contains an "Implied Seek" feature. This allows the Host processor to issue a read or a write function even though the heads are sitting over the wrong cylinder. The WD1050 will perform an automatic seek operation

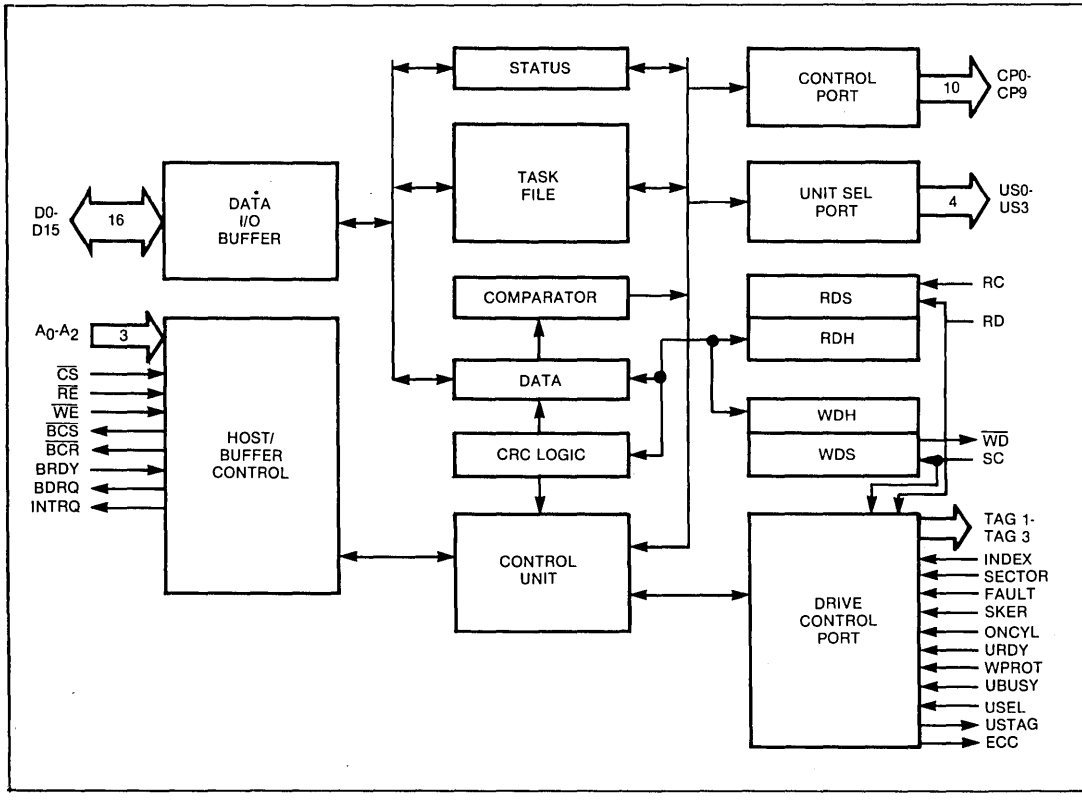


FIGURE 3. WD1050 BLOCK DIAGRAM

before the actual read or write. Because of this, the head movement commands (Return to Zero and Seek Cylinders) are usually restricted for use in overlap seeks. This is the ability to perform seek operations on several drives simultaneously.

After a command has finished execution, the WD1050 will report to the Host through its Status Register (shown in Figure 6) how successful a command execution was. Many commands will not execute if certain conditions are not met. For example, a FAULT condition, shown by status bit 6, will prevent all commands except Fault Clear from executing. Read / Write commands will not execute if the "On Cylinder" bit is false, either. In summary, the Host must examine the various bits to determine what action to take next.

HOST SECTOR BUFFER

Because of the high data rates used on the SMD protocol (9.677 Mbits / sec.), even a fast micro- processor will have trouble keeping up in a Programmed I / O environment. For this reason, the WD1050 has been designed to use a sector buffer.

Figure 7 shows a Host Interface to the device using a low cost Static RAM and a binary counter. Since the WD1050 will be transferring data directly to the RAM, a transceiver will be needed to isolate the Host from the RAM / WD1050 logic. This transceiver, as shown in Figure 7, is disabled by Buffer Chip Select (BCS). Whenever BCS is active, the WD1050 is reading or writing to the RAM. During this condition, the Host cannot read status or any other registers. When the data transfer is over, the device disables BCS and enables Buffer Data Request (BDRQ). This tells the Host that the buffer is now available for use. If a read command had been issued, the sector buffer would have filled the data requested.

During this process, the WD1050 takes control over Write Enable ($\overline{WE}$) by making it an output. It places its first data word on the bus and strobes $\overline{WE}$. This causes a write operation to the RAM and increments the binary counter that is tied to the RAM's address lines. Another $\overline{WE}$ strobe then occurs, increments the counter again, and the process continues until the sector is transferred. If a single sector operation was requested, the WD1050's use of the sector buffer is completed. However, multiple sectors may be transferred as an option within the command. In this case, the Buffer Ready (BRDY) input to the device is examined. If false, the WD1050 assumes there is more RAM available and transfers the next sector of data. The BRDY signal is normally generated by a "carry" or overflow out of the binary counter. If BRDY has gone active but the device still has more sectors to transfer, BDRQ will be made active to allow the Host to unload the data in the RAM, making room for the additional sectors. The WD1050 will then resume its operation of finding a sector and writing the data to the buffer. After all the data has been transferred, the command will terminate. To complete the scheme, a signal called Buffer Counter Reset (BCR) is used to zero the counters before the Host or device starts a transfer. A BCR pulse is generated whenever BCS makes a transition.

By using this buffer scheme, the designer has the ability with one command to transfer the maximum number of sectors specified by the Sector Counter in the SDH Register.

CONCLUSION

Using the WD1050 as the basis for an SMD controller design, can reduce the complexity of the design effort considerably. However, challenges still remain in interfacing the device to maximize the efficiency of the interface. The buffer control signals, for example, can be changed to accomodate a DMA controller for higher throughput. ECC can be appended to the buffer for data correction purposes. A local microprocessor dedicated on the SMD controller board could even be used to emulate existing SMD / Host software routines.

Regardless of the application, the WD1050 signifies a trend in the semiconductor industry to not only replace logic in a discrete design, but to offer complete functions in large scale integration. This device is certainly not the first to offer an LSI functional building block, and will not be the last.

A2	A1	A0	REGISTER SELECTED
0	0	0	Head Number/Sector Address
0	1	0	Sector Count/Length/Unit Address
1	0	0	16 Bit Cylinder Register
1	1	0	Command Register (Write Only)
1	1	1	Status Register (Read Only)

Figure 4. WD1050 Task File

COMMAND	LSB				COMMAND REGISTER BITS											MSB
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fault Clear	1	0	0	0	0	0	0	I	0	0	0	0	U	S	E	L
Return to Zero	1	0	0	1	V	L	O	I	0	0	0	M	U	S	E	L
Seek Cylinder	1	0	1	0	V	L	O	I	Z	C	H	M	U	S	E	L
Read ID Field	1	0	1	1	R	L	O	I	Z	C	H	M	U	S	E	L
Read Sector	1	1	0	0	R	L	O	I	Z	C	H	M	U	S	E	L
Write Sector	1	1	0	1	R	L	O	I	Z	C	H	M	U	S	E	L
Format	1	1	1	0	R	P	O	I	Z	C	H	M	U	S	E	L
Verify	1	1	1	1	R	P	O	I	Z	C	H	M	U	S	E	L

FLAG SUMMARY	
V = Verify	I = Interrupt Enable
R = CRC Enable	Z = Volume/Head change
L = Logical Sectoring	C = Cylinder Address
P = Programmable Sectors	H = Head Selection
O = On Cylinder	M = Marginal Data Recovery
E = Priority Release/Early	U = Unit Sel/Servo Minus
L = Unit Deselect/Late	S = Priority Sel/Servo Plus

Figure 5. WD1050 Instruction Set

BIT	NAME	DESCRIPTION
0	ID Field Not Found (ID/NF)	Set if the sync character preceding the ID Field or ID Field contents read from the disk do not match the respective Task File contents.
1	ID CRC Error (IDCE)	Set if the CRC calculation on the ID Field read from the disk is in error.
2	Data Field Not Found (DFNF)	Set if the Data Field sync pattern following the ID Field does not match the sync character.
3	Data Field CRC Error (DFCE)	Set if the CRC Calculation on the Data Field read from the disk is in error.
4	Not Used	This bit is not used; it is forced to a zero.
5	Buffer Data Request (BDRQ)	Reflects the Buffer Data Request output.
6	Fault (FLT)	Reflects the status of the Fault (FLT) input.
7	Buffer Chip Select (BCS)	This bit is an inverted copy of the Buffer Chip Select ($\overline{BCS}$) output.
8	Seek Error (SKER)	Reflects the status of the Seek Error (SKER) input.
9	On Cylinder (OCYL)	Reflects the status of the On Cylinder (OCYL) input.
10	Unit Ready (URDY)	Reflects the status of the Unit Ready (URDY) input.
11	Write Protect (WPRT)	Reflects the status of the Write Protect (WPRT).
12	Unit Selected (USEL)	Reflects the status of the Unit Selected (USEL) input.
13	Unit Busy (UBSY)	Reflects the status of the Unit Busy (UBSY) input.
14	CIP	Set when a command is in progress.
15	Buffer Chip Select (BCS)	This bit is an inverted copy of the Buffer Chip Select ($\overline{BCS}$) output. This bit also appears in STATUS Bit 7.

Figure 6. WD1050 Status Register

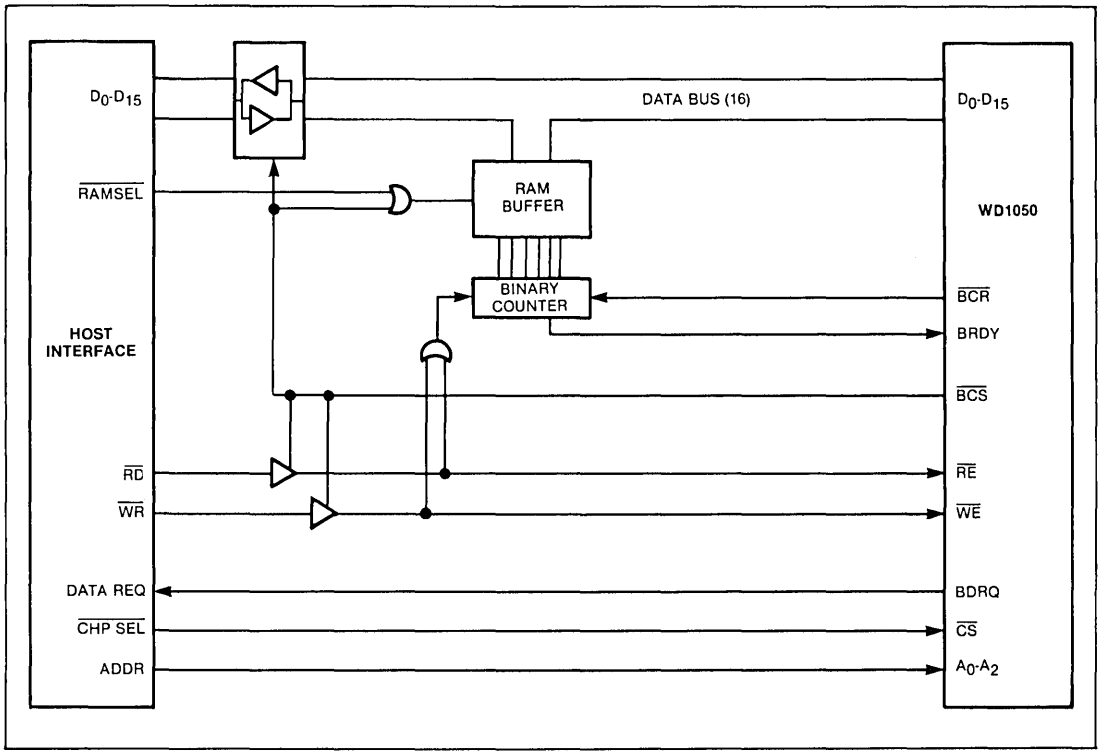


Figure 7. WD1050 Host Interface