SX1602A

IPI-2 DISK DRIVE SERDES / FORMATTER CIRCUIT

Features

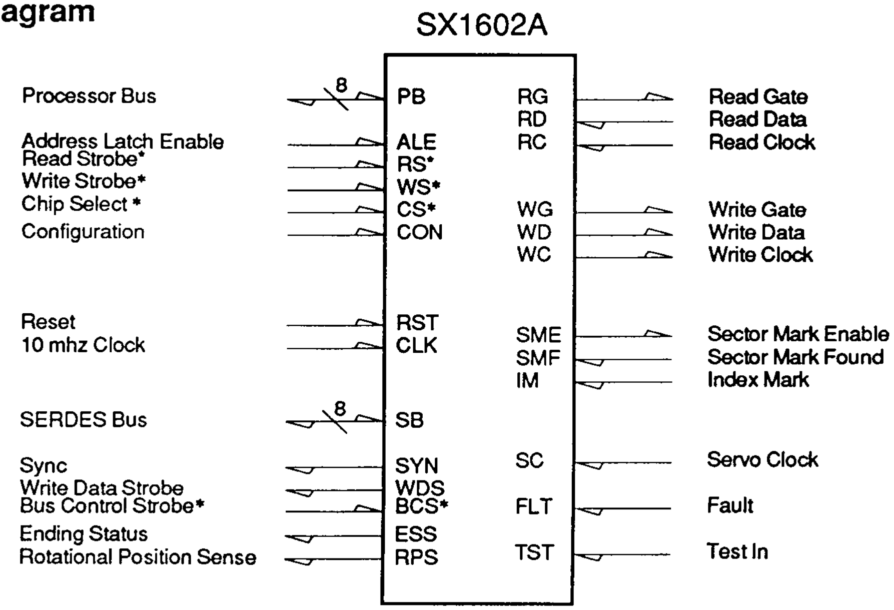
- Interprets IPI-2 Sector and Field type Data Bus Controls
- Performs serial-parallel conversion of disk data
- Performs header verification on disk writes
- Executes surface verification read/write primitives
- Functions in Hard Sector, Soft Sector and Embedded Servo Modes
- Supports disk transfer rates up to 40Mhz
- Partitions a disk track into sectors
- Partitions disk sectors into 1, 2 or 3 fields
- Generates Read Gate, Write Gate and Sector Enable signals
- Acquires PLO and byte synchronization on disk reads
- Generates PLO and byte sync fields on disk writes
- 68 pin PLCC package

Description

The IPI-2 Serdes/Formatter Circuit (SFC) is a 68-pin VLSI integrated circuit fabricated with semi-custom CMOS technology. The SFC is intended to provide the serial-parallel conversion and format control functions for a magnetic disk drive implementing the IPI-2 interface. The SFC, in conjunction with the Interface Protocol Circuit (IPC), interprets all IPI-2 Fixed Sector Data Controls and schedules the timing of Write Gate, Read Gate, and/or Sector Mark Enable to record or recover data to / from a magnetic disk drive.

The SFC interfaces to a drive processor as a slave peripheral device. The SFC is initialized and programmed by reading and writing to an internal register array. The format of data to be recorded or recovered from the disk drive is established through the values stored in these registers.

Logic Diagram



Rev. 1.0 9/22/87

1.0 APPLICABLE DOCUMENTS

The following documents are considered a part of this specification:

1) Simulex Corporation Product Specification, SX1601 Interface Protocol Circuit (IPC)
2) ANSI Standard X3.129, IPI Physical
3) ANSI Standard X3.130, IPI Command Set, Device Specific for Magnetic Disk

1.1 Block Diagram

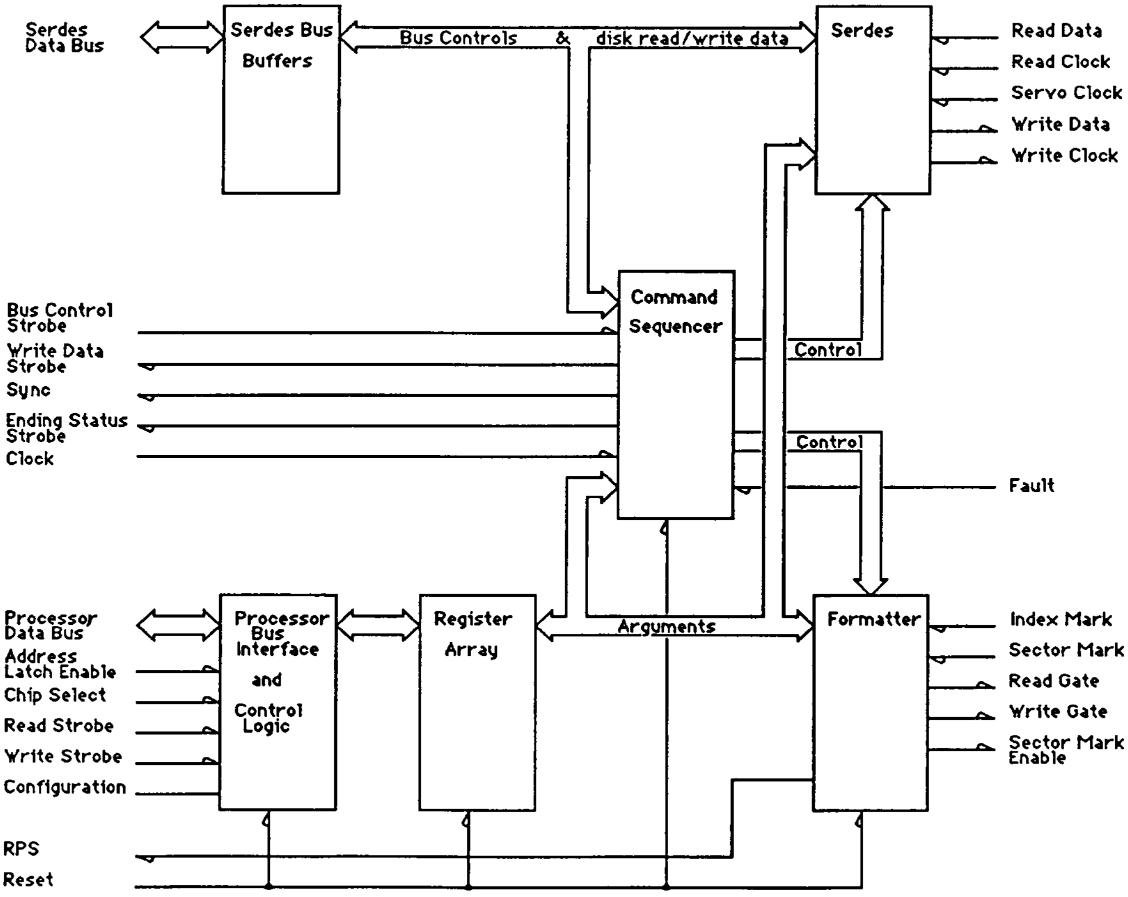


Figure 1
Serdes/Formatter Circuit
Block Diagram

2.0 SIGNAL DESCRIPTIONS

The signal description table provides a brief description of the function of each signal and its mnemonic. Signal mnemonics ending in an " * " are active low, all other signals are active high.

2.1 Processor Interface Signals

Signal Mnemonic	Input/ Output	Signal Description
PB 0-7	I/O	**Processor Address/Data Bus** The Processor Bus lines are TTL compatible bi-directional 3-state active high signals for data exchange between the drive processor and the SFC. The bus address is captured at the beginning of a read or write cycle when Address Latch Enable goes inactive. The outputs are enabled during a read cycle to allow the contents of the SFC internal registers to be examined by the drive processor. The outputs are disabled and the internal inputs are enabled during a write cycle to allow the contents of the internal registers to be altered.
RS*(DS*)	I	**Read Strobe (Data Strobe)** This TTL compatible high impedance active low input performs one of two functions depending on the state of the CON input. When CON is inactive, this input becomes a Read Strobe signal and enables the contents of the addressed register to be output to the Processor Bus while the RS* signal is active. When CON is active, this input becomes a Data Strobe and enables the contents of the Processor Bus to be latched into the addressed register if R/W is active, or allows the contents of the addressed to be output to the Processor Bus if R/W is inactive.
WS*(R/W)	I	**Write Strobe (Read/Write)** This TTL compatible high impedance active low input performs one of two functions depending on the state of the CON input. When CON is inactive, this input becomes a Write Strobe and causes the contents of the Processor Bus to be latched by the addressed register on the active to inactive going edge of the WS* signal. When CON is active, this input becomes a Read/Write signal and determines whether a Processor Bus read or write cycle will be performed when DS* is active. If R/W is active, the active going edge of DS* starts are read cycle. If R/W is inactive, the active going edge of DS* starts are write cycle.
CS*	I	**Chip Select** This TTL compatible high impedance active low input enables the SFC to respond to the RS and WS signals.

ALE I **Address Latch Enable**

This TTL compatible high impedance input indicates when a valid address is present on the Processor Bus. The address is internally latched on the active to inactive going edge of the ALE signal. When CON is inactive this input is active high. When CON is active this input is active low.

RST I **Reset**

This TTL compatible high impedance input resets and initializes the SFC circuit when an active level input is detected for five or more clock cycles. When the CON signal is inactive the RST signal is active high. When CON is active this input is active low.

CON I **Configuration**

This TTL compatible high impedance active high input determines the processor interface mode of operation.

When low, the processor bus control signals, are configured as RD* and WR* and behave in a manner compatiable with the Intel 8051.

When this signal is high, the processor bus control signals, are configured as R/W and DS* and behave in a manner compatiable with the Zilog Z8 and Motorola 6801 processors.

CLK I **10 Megahertz Clock**

This TTL compatible high impedance active high signal is the circuit clock. Frequency must be 10 MHz ±1%. Symmetry must be 50/50 ±5%.

2.2 Serdes Bus Interface Signals

SB 0-7 I/O **Serdes Data Bus**

The Serdes Data Bus lines are CMOS bi-directional 3-state active high signals for data exchange between the Serdes/Formatter Circuit and the IPC. Bus Controls, read/write disk data, and an Ending Status Byte are passed over this bus for each Data Control executed by the SFC.

BCS* I **Bus Control Strobe**

This CMOS high impedance active low input signals the receipt by the IPC of a Data Bus Control from the IPI interface.

The active going edge of this signal indicates that a Data Bus Control is being placed on the Serdes Bus data lines.

This signal goes inactive when a Master Termination of the data transfer is recognized by the IPC, or when the SFC initiates a Slave Termination of the data transfer by asserting Ending Status Strobe.

WDS O **Write Data Strobe**

This CMOS active high output is used during disk write operations to indicate when disk write data is to be place on the Serdes Bus data lines from the IPC data FIFO. The FIFO is advanced and the next byte of write data is placed on the Serdes Bus on the inactive going edge of this signal. During disk read operations the Write Data Strobe signal is inactive.

SYN O **Sync**

This CMOS active high output provides a pulse stream during the execution of disk read/write Data Controls. This pulse stream is used by the IPC to generate IPI Sync In pulses. During the execution of disk read Data Controls this signal also indicates the presence of valid disk data on the Serdes Bus.

ESS O **Ending Status Strobe**

This CMOS active high output indicates the end of a disk read/write Data Control operation, and the presences of an Ending Status Octet on the Serdes Bus data lines.

When this signal goes active, the IPC is required to drive BCS to the inactive state if it not in that condition, and to release the BCS signal and Serdes Bus data lines in preparation for the transfer of ending status.

When this signal goes inactive, an Ending Status Octet is being placed on the Serdes Bus data lines by the SFC.

RPS O **Rotation Position Sensing**

This CMOS active high output, when active, indicates that the disk data heads are over the target area defined in the register set.

2.3 Disk Interface Signals

RD I **Read Data**

This TTL compatible high impedance active high input receives the serial NRZ disk data, synchronized to Read Clock, recovered from disk. Read data must be valid on the rising edge of Read Clock.

WD O **Write Data**

This TTL compatible active high output presents serial NRZ data, synchronized to Write Clock, to be recorded on disk. Data is valid on the active going edge and changed on the inactive going edge of Write Clock.

SC I **Servo Clock**

This TTL compatible high impedance active high input receives the disk drive servo clock signal. This clock signal is used to determine the starting location of sectors and fields, as well as generating most output signals, and therefore must be present any time the drive spindle is at speed (i.e. during actuator seeks).

RC I **Read Clock**

This TTL compatible high impedance active high input receives the disk drive read clock signal. The active going edge of this signal is used to strobe the state of the Read Data signal into the Serdes. Read Clocks must be present for a minimum of 2 bytes after the fall of Read Gate.

WC O **Write Clock**

This TTL compatible active high output transmits the Write Clock signal, which is synchronized with the serial NRZ Write Data signal. The Write Clock is the received Servo Clock re-transmitted to the drive and is used to strobe Write Data from the Serdes. Write data is valid on the active going edge and changed on the inactive going edge of Write Clock.

IM I **Index Mark**

This TTL compatible high impedance active high input marks the beginning of a disk track. The active going edge of this signal must be synchronized with Servo Clock.

RG O **Read Gate**

This TTL compatible active high output, when activated, indicates that the Serdes/Formatter Circuit is beginning a disk read operation. The drive read chain is expected to lock to the PLO sync segment and output Read Data in synchronization with Read Clock. Read Gate is de-activated after the last data byte from a given field has been read.

WG O **Write Gate**

This TTL compatible active high output, when activated, indicates that the Serdes/ Formatter Circuit is beginning a disk write operation.The drive write chain is expected to record the data present on the Write Data line in synchronization with the Write Clock. Write Gate is negated after the last byte of the PAD segment has been output.

SME O **Sector Mark Enable**
This TTL compatible active high output, when active in conjunction with Read Gate, indicates that the drive is required to search for a sector mark (Soft Sector Mode only). When active in conjunction with Write Gate, it is an indication that the drive is required to record a sector mark on the disk. This signal is synchronized with Write Clock. Sector Mark Enable is also used in other modes to provide external sync signals for testing, see section 4.1.4 for a complete description of its operation. Sector Mark Enable is only allowed to go active when it is enabled with bit 4 of Register 25 (Last Field Reg).

SMF I **Sector Mark Found**
This TTL compatible high impedance active high input is asserted by the drive when a sector mark has been detected (Soft Sector and Embedded Servo Modes only). This signal must be synchronized with Servo Clock.

FLT I **Fault**
This TTL compatible high impedance active high input indicates the occurrence of a disk read/write fault. When active, the SFC will abort any executing read/write operation or reject any further Bus Controls with an Operation Exception Slave Status Byte.

2.4 Device Test

TST I **Test In**
This TTL compatible high impedance active high input is used during the manufacturing test of the chip and must be grounded during normal operation.

2.5 Power

VCC1	Positive Five Volt Power
VCC2	Positive Five Volt Power
VCC3	Positive Five Volt Power
VCC4	Positive Five Volt Power
VCC5	Positive Five Volt Power
GND1	Power Ground and Signal Reference
GND2	Power Ground and Signal Reference
GND3	Power Ground and Signal Reference
GND4	Power Ground and Signal Reference
GND5	Power Ground and Signal Reference

2.6 Signal Characteristics

		Output		Input		
Signal Mnemonic	Input/ Output	I_{OL} (ma)	I_{OH} (ma)	Threshold	Setup (ns)	Hold (ns)
RS*	INPUT			TTL	Note 1	
WS*	INPUT			TTL	Note 1	
CS*	INPUT			TTL	10.0	10.0
ALE	INPUT			TTL	Note 1	
RST	INPUT			TTL	Note 1	
CON	INPUT			TTL	Note 1	
CLK	INPUT			TTL	Note 1	
BCS*	INPUT			CMOS	15.0	5.0
RD	INPUT			TTL	15.0	5.0
RC	INPUT			TTL	Note 1	
IM	INPUT			TTL	15.0	5.0
SMF	INPUT			TTL	15.0	5.0
SC	INPUT			TTL	Note 1	
FLT	INPUT			TTL	Note 1	
TST	INPUT			TTL	Note 1	
WDS	OUTPUT	4.0	4.0			
SYN	OUTPUT	4.0	4.0			
ESS	OUTPUT	4.0	4.0			
RPS	OUTPUT	4.0	4.0			
WD	OUTPUT	8.0	8.0			
WC	OUTPUT	8.0	8.0			
RG	OUTPUT	8.0	8.0			
WG	OUTPUT	8.0	8.0			
SME	OUTPUT	8.0	8.0			
SB(0-7)	I/O	4.0	4.0	CMOS		
PB(0-7)	I/O	12.0	12.0	TTL		

Note 1 - No Setup or Hold requirements.

2.7 Signal Pin Assignments

Signal	Pin	Signal	Pin
SB0	40	SB1	39
SB2	38	SB3	37
SB4	34	SB5	33
SB6	32	SB7	31
PB0	16	PB1	15
PB2	14	PB3	12
PB4	7	PB5	5
PB6	4	PB7	3
SYN	29	ESS	41
RPS	67	BCS*	42
RS*	19	WS*	20
CS*	21	ALE	22
CLK	68	RST	23
TST	48	CON	24
FLT	47	IM	58
RC	66	RD	65
RG	50	SC	51
SME	56	SMF	57
WC	54	WD	55
WDS	30	WG	49
VCC1	36	VCC2	1
VCC3	8	VCC4	17
VCC5	53	GND1	2
GND2	6	GND3	13
GND4	18	GND5	35
GND6	52		

2.8 Package Pin Assgnments

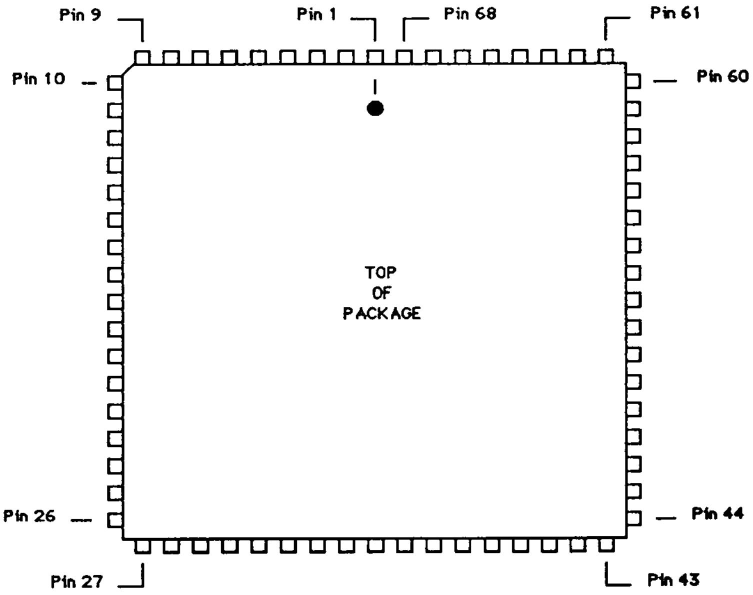


Figure 2
68 Pin PLCC
Pin Assignments

3.0 FUNCTIONAL DESCRIPTION

The Serdes/Formatter Circuit is a single-chip implementation of the Formatter and Serdes circuitry for an IPI level-2 disk drive. It provides the means to record, recover, and format user data on a magnetic disk drive. The disk data format and functional configuration of the SFC is programmed by the disk drive processor. After the operating mode and the data format have been established, the SFC is capable of executing all IPI-2 Data Bus Controls without further assistance from the drive processor.

There are six major components to the circuit: the Processor Bus Interface, Register Array, Serdes Bus Interface, Command Sequencer, Serdes, and Formatter. The following sections describe the various components of the circuit.

3.1 Processor Bus Interface

The Processor Bus Interface, in conjunction with the Register Array, allows the drive processor to setup the mode of operation (Hard Sector, Soft Sector, or Embedded Servo) and to specify the format under which data will be recorded and recovered. The drive processor configures and programs the SFC through the values stored in the Register Array.

The SFC acts as a processor bus slave. The Processor Bus consists of an eight-bit address/data bus, and the Address Latch Enable, Chip Select, and two bus control signals. The function of the bus control signals is determined by the state of the SFC Configuration signal (CON). If CON is inactive, the two bus control signals function as Read Strobe and Write Strobe, in a manner compatible with the Intel 8051 family of micro-controllers. If CON is active, the two bus control signals function as Read/Write and Data Strobe, in a manner compatible with the Zilog Z8 and Motorola 6801 micro-controllers.

The active state of ALE indicates that a valid address is on the data bus. The SFC latches this address and the state of the Chip Select signal in an internal address register. If the Chip Select signal was active and the address falls within the address map of the SFC, the circuit becomes selected.

When the CON signal is inactive: If Read Strobe goes active and the circuit is selected, the contents of the addressed register are output to the data bus lines. If Write Strobe goes active and the circuit is selected, the eight-bit data value present on the data bus is stored in the addressed register.

When the CON signal is active: If Data Strobe goes active when Read/Write is true and the circuit is selected, the contents of the addressed register are output to the data bus lines. If Data Strobe goes active when Read/Write is not true and the circuit is selected, the eight-bit data value present on the data bus is stored in the addressed register.

The Processor Bus Interface is directly timing compatible with the Intel 8051, Zilog Z8, or Motorola 6801 family of microprocessors. With the appropriate logic to multiplex the address onto the data bus and to generate the proper bus control signals, most eight-bit microprocessors can be adapted for use with the SFC.

3.2 Register Array

The register array occupies the second block of 32 addresses within the 256 byte addressing space. All register can be read or written by the drive processor unless otherwise specified.

Table 1 lists the registers and their hexadecimal addresses.

Table 1 List of Registers

Address	Description
20	Control Register
21	Status Register
22	Target Sector Register
23	Sector Mark Search Window Length Register
24	Last Sector Register
25	Last Field Register
26	Sector Length Register (high byte)
27	Sector Length Register (low byte)
28	Sync Pattern Register
29	Current Sector Register
2A	Verify Skip Length Register
2B	Start RPS Register
2C	not used, all zero's returned when read
2D	Write only - Diagnostic Strobe
2E	Write only - Write Interlock
2F	Write only - Error Reset
30	Read Gate Delay Register
31	PLO Sync Field Length Register
32	Data Field 0 Length Register/Start Byte RPS (high byte)
33	Data Field 0 Length Register/Start Byte RPS (low byte)
34	Data Field 1 Length Register/End Byte RPS (high byte)
35	Data Field 1 Length Register/End Byte RPS (low byte)
36	Data Field 2 Length Register (high byte)
37	Data Field 2 Length Register (low byte)
38	Field 0 Length Register (high byte)
39	Field 0 Length Register (low byte)
3A	Field 1 Length Register (high byte)
3B	Field 1 Length Register (low byte)
3C	Field 2 Length Register (high byte)
3D	Field 2 Length Register (low byte)
3E	Segment Byte Counter (high byte)
3F	Segment Byte Counter (low byte)

3.2.1 Control Register (20)

This eight-bit read/write register controls the operation of the Serdes/Formatter Circuit. The Circuit can be directed to execute the surface verification read/write primitives, have the sector mode established, or put into the diagnostics mode through the control bits contained in this register. All control bits are reset by the external Reset input. The definition of each bit is outlined below:

Bit	Description
7	Execute Write Primitive The function of this bit is determined by the formatter Mode established in bits 5 and 4 of this register. In Track Verify mode (bits 5 and 4 cleared) this bit, when set, arms

the Verify Write primitive. When a zero is subsequently written to bit 7 of register address 2E (as an additional interlock before writing) the SFC initiates the Verify Write Primitive. When the Write operation is complete the SFC resets this bit. See section 4.3 for a description of the Verify function.

In Soft Sector mode (bit 5 cleared, bit 4 set) this bit, when set, causes the SFC to format a track with sector marks. The operation begins with a search for the Index Mark. When Index is found the SFC begins writing address marks at the start of each sector. Address marks are written following the internally generated Sector Mark pulse, after a delay in bytes equal to the contents of the Sector Mark Search Window Length register. When Index is encountered a second time, the SFC stops writing address marks and resets this bit.

In the Hard Sector and Embedded Servo modes this bit has no meaning and cannot be set.

6 Execute Read Primitive
The function of this bit is determined by the formatter Mode established in bits 5 and 4 of this register.

In the Track Verify mode (bits 5 and 4 cleared) this bit, when set, causes the SFC to initiate the Verify Primitive. When the verify primitive is complete the SFC resets this bit. See section 4.3 for a description of the Verify function.

In the Hard Sector, Soft Sector, and Embedded Servo modes this bit has no meaning and cannot be set.

5,4 Formatter Mode
Bits 5 and 4 determine the formatter mode as defined below:

Bit 5	Bit 4	Mode
0	0	Track Verify
0	1	Hard Sector
1	0	Soft Sector
1	1	Embedded Servo

3 Run
When this bit is set, the SFC is enabled and Data Bus Controls can be accepted. When the SFC is enabled Registers 30 thru 3D can not be accessed from the Processor Bus (Registers 32 thru 35 can be accessed but have a different definition).

When this bit is reset, the SFC is reset and disabled and Data Bus Controls will be rejected.

2,1 Diagnostic Mode

Bits 2 and 1 determine the diagnostic mode as defined below:

Bit2	Bit1	Diagnostic Mode
0	0	Normal operation
0	1	Load Counters
1	0	Decrement Counters
1	1	not used

See section 4.4 for a description of the diagnostic mode operation.

0 Not used, returns the value written.

3.2.2 Status Register (21)

This eight-bit read only register contains the current state of the Serdes/Formatter Circuit. The drive processor can determine through this register the reason for a command rejection, or the state of the Diagnostics output. All status bits are reset by the external Reset input. The definition of each bit is outlined below:

Bit Description

7,6 Data Control Reject Type

Bits 7 and 6 report three catagories of Data Control rejects as defined below:

Bit 7	Bit 6	Reject Type
0	1	Invalid Data Control The last Data Control received was invalid or not supported.
1	0	Data Control Context A The last Data Control was received while the SFC was disabled (Control Register bit 3 was zero).
1	1	Data Control Context B The last Data Control received was a Target type and the RPS function was disabled (Target Register = FF) or it was a Sector type Data Control that attempted to operate on more fields that were programmed in the Last Field Register.

5 Data Control Late

When this bit is set, the last Data Control received was rejected as late (required orientation not established).

4 Operation Fault

When this bit is set, the last Data Control received incurred an operation fault via the Fault input signal.

3 Track Overrun Fault
When this bit is set, a track overrun condition has occurred. This condition normally occurs when the SFC has been programmed for a track length in excess of the actual track length or a false Index Mark was received.

2 Sector Overrun Fault
When this bit is set, a sector overrun condition has occurred. This condition normally occurs when the SFC has been programmed for a sector length in excess of the actual sector length.

1 Field Overrun Fault
When this bit is set, a field overrun condition has occurred. This condition is normally caused by the SFC being programmed for a field length in excess of the actual field length or by a field being read late (displaced by head scatter) so that the SFC is still reading at the end of the programmed field.

0 Diagnostic Mux Output
This bit reflects the status of internal circuitry. Its meaning depends on the state of the two Diagnostic Control Bits in Control Register bits 3 and 2 respectively:

Bit 3	Bit 2	Bit 0 Output
0	0	Output of Verify Skip comparator
0	1	Output of RPS and Target comparators
1	0	"And" of zero detect for all counters
1	1	not used

For a description of the use of this bit see section 4.4

3.2.3 Target Sector Register (22)

This eight-bit read/write register contains the address of the Target Sector. The address is formed by subtracting the Target Sector number from value contained in the Last Sector register. The Target Sector is the designated sector upon which certain "target type" Data Controls operate.

When the Enable Byte RPS feature is disabled (bit 6 of the Last Field Register is zero), the RPS signal will be raised at the beginning of the target sector (if not already activated by the Start RPS Register) and dropped at the end of the target sector. In this mode, if the Target Register is set to "FF" the RPS function is disabled.

When the Enable Byte RPS feature is enabled (bit 6 of the Last Field Register is a one), the RPS signal will be raised at the byte location on the track specified in the Start Byte RPS Register and dropped at the byte location specified in the End Byte RPS Register.

The Target Register also contains the "seed pattern" used during a Write Primitive when in the Surface Verification mode. See section 4.3 for a description of the Verify function.

3.2.4 Sector Mark Search Window Length Register (23)

This seven-bit read/write register contains the value in bytes of the sector mark search window. In Soft Sector mode, the Sector Mark Enable signal will go active from either the end of the previous sector or the rising edge of Sector Mark, for a period of time equal to the Sector Mark Search Window Length.

3.2.5 Last Sector Register (24)

This eight-bit read/write register contains the address of the last sector on a track. The contents of this register are loaded into the Sector Counter when the Index input goes active.

3.2.6 Last Field Register (25)

This five-bit read/write register contains the number of fields in a sector, the verify pattern modulo, the enable continuous RPS bit and the SME enable bit.

Bit — Description

7 — not used, returns zero when read

6 — Enable Byte RPS Mode
When this bit is set, the SFC will determine the start and end of the generated RPS signal from the Start Byte RPS and End Byte RPS Registers (32 through 35).

When this bit is reset, the SFC will determine the start and end of the generated RPS signal from the Start RPS Register (2B) and Target Sector Register (22).

5 — Enable Continuous RPS
When this bit is set, the SFC will continue to generate the RPS signal after the receipt of a Data Bus Control.

When this bit is reset, the SFC will discontinue generation of the RPS signal after the receipt of a Data Bus Control.

4 — Enable SME
When this bit is set, the SFC will generate the Sector Mark Enable signal. See section 4.1.4 for a description of SME generation.

When this bit is reset, the SFC will discontinue generation of the SME signal.

3,2 Verify Pattern Length

Bits 3 and 2 specify the length (in bits) of the repeating verify pattern as defined below:

Bit 3	Bit 2	Repeating Pattern
0	0	8 bits - Target Register bits 7 thru 0
0	1	7 bits - Target Register bits 6 thru 0
1	0	6 bits - Target Register bits 5 thru 0
1	1	5 bits - Target Register bits 4 thru 0

1,0 Last Field

Bits 1 and 0 specify the number of Fields in a Sector as defined below:

Bit 1	Bit 0	Last Field (number of fields per sector)	
0	0	0	(1)
0	1	1	(2)
1	0	2	(3)
1	1	not used, invalid selection	

3.2.7 Sector Length Register (26, 27)

This 16-bit read/write register contains the sector length in bytes. The contents of this register are loaded into the Sector Byte Counter whenever that counter reaches a count of zero.

3.2.8 Sync Pattern Register (28)

This eight-bit read/write register holds the Sync Byte pattern. During a Write type operation this pattern is written following the PLO Segment and preceeding the Data Segment. On a Read type operation this pattern is compared against the incoming serial Read Data to establish byte sync for the transfer.

3.2.9 Current Sector Register (29)

This eight-bit read only register contains the address of the data sector over which the read/write heads are currently positioned.

3.2.10 Verify Skip Length Register (2A)

This eight-bit read/write register contains the header Verify skip length, in bytes, to be performed during the excution of a Verify operation (whether executed via a Bus Control or the Processor Bus).

3.2.11 RPS Start Register (2B)

This eight-bit register contains the sector address at which the RPS signal will be asserted when the SFC is not is the Byte RPS mode (see Last Sector Register bit 6). The RPS signal will be deasserted at the end of the sector address by the Target Register. If this register is set to the same value as the Target Register the RPS signal will be one sector long.

During Write or Verify operations executed from the Processor Bus during Surface Verification this register is instead used to hold the most significant byte of the Verify skip length value.

3.2.12 Read Gate Delay Register (30)

This eight-bit read/write register holds the delay byte count from the start of a field until Read Gate is asserted when reading a field. During a read or write to disk, the contents of this register are loaded into the Segment Byte Counter at the start of the each field. The number in the register is one less than the actual byte count.

3.2.13 PLO Length Register (31)

This eight-bit read/write register holds the length, in bytes, of the PLO Segment. During a read or write to disk, the contents of this register are loaded into the Segment Byte Counter after the Read Gate Delay count has been exhausted. The number in the register is one less than the actual byte count.

3.2.14 Data Length Registers (32 thru 37 when Control Register bit 3 is zero)

These three 16-bit read/write registers hold the data length of the three fields in a sector. The contents of the Data Length Register selected by the Field Counter, are loaded into the Segment Byte Counter at the start of the Data Segment of a field. The number in the register is one less than the actual byte count.

3.2.15 Start Byte RPS Register (32 and 33 when Control Register bit 3 is a one)

This 16-bit read/write register holds the byte location of the beginning of the RPS pulse when the SFC is in the Byte RPS mode (see Last Field Register bit 6).

Note: Writing to the Start Byte RPS Register suppresses RPS until the Target Register (22) is subsequently written into to prevent glitches on the RPS signal while setting new RPS values.

3.2.16 End Byte RPS Register (34 and 35 when Control Register bit 3 is a one)

This 16-bit read/write register holds the byte location of the end of the RPS pulse when the SFC is in the Byte RPS mode (see Last Field Register bit 6).

3.2.17 Field Length Registers (38 thru 3D)

These three 16-bit read/write registers contain the length, in bytes, of the three fields in a sector. The contents of the Field Length Register selected by the Field Counter are loaded into the Field Byte Counter every time its count goes to zero. The number in the register is one less than the actual byte count.

3.3 SERDES BUS INTERFACE

The Serdes Bus Interface is the data path over which Data Bus Controls, Ending Status, and disk read/write data are passed during the execution of IPI read/write operations. The interface consists of an eight-bit data bus and four control signals: Bus Control Strobe, Write Data Strobe, Sync, and Ending Status Strobe. These four control signals establish the nature and direction of information exchanged across the data bus. Examples of the Serdes Bus Interface operation for disk reads and disk writes are given in Figures 3 and 4 respectivly.

3.4 COMMAND SEQUENCER

The Command Sequencer receives, decodes, and executes Data Bus Controls from the Serdes Bus Interface, and generates an Ending Status byte on completion of a disk read/write operation.

The Sequencer controls the operation of the Serdes, enables the Formatter signals at the proper time, and directs the exchange of information across the Serdes Bus Interface, during the execution of a disk read/write operation.

3.5 SERDES

The Serdes is responsible for recording and recovering serial disk data under the direction of the Command Sequencer. During disk read operations, the Serdes acquires byte synchronization and converts disk read data from bit serial to bit parallel. During disk write operations it writes the PLO Sync segment and Sync Byte, and converts write data from bit parallel to bit serial during write operations.

The Serdes consists of an eight-bit, parallel input/output shift register, data holding register, and comparator. The serial input to the shift register is from the Read Data signal. The serial output is to the Write Data signal. The shift register is clocked by the Servo Clock signal during disk write operations, and by the Read Clock signal during disk read operations.

During a disk read, the parallel output of the shift register is compared against the value in the Sync Byte register until the Sync Byte is detected. After byte sync is obtained, the Serdes outputs a byte of disk read data to the holding register on every eighth Read Clock pulse. The data in the holding register is then transferred across the Serdes Bus Interface to the Interface Protocol Circuit.

If the disk read operation is part of a verify/write Data Control, the output of the holding register is compared against the data present on the Serdes Bus Interface. Any miscompare generates a signal to the Command Sequencer which supplies the proper status at the end of the transfer.

During disk write operations, the Serdes outputs a PLO sync segment of the length specified by the value contained in the PLO Length register, and outputs the value contained in the Sync Byte Register before shifting out the disk write data obtained from the Serdes Bus Interface.

3.6 FORMATTER

The Formatter partitions the disk drive data tracks into user defined data sectors through the scheduling of Read Gate, Write Gate, and Sector Mark Enable signals, based on the values contained in the Register Array.

The Formatter divides the data tracks into sectors, sectors into fields, and fields into segments. Each sector may contain an Address Mark (if Soft Sector Mode is selected) and up to three fields. Each field is further divided into a Read Gate Delay segment, PLO Sync segment, Sync Byte segment, Data Field segment, Pad segment, and Decision Gap and/or Write to Read Recover Gap segment. All sectors on a track will be identical. Each field within a sector will contain identical Read Gate Delay, PLO Sync, Sync Byte, and Pad segments. Each field may have a different Data field and Decision Gap and/or Write to Read Recover Gap lengths. All sector field segment lengths and the Sync Byte value are programmable via the SFC register set.

The Formatter consists of a number of counters. The counters are used to establish the "turn on" and "turn off" times of Read Gate, Write Gate, and Sector Mark Enable, as well to mark the beginning and end of Sectors, Fields, and Segments. Table 2 is a tabulation of the Formatter counters.

Table 2 List of Formatter counters

Sector Counter

Sector Byte Counter
Field Counter
Field Byte Counter
Segment Counter
Segment Byte Counter
Window Byte Counter

3.6.1 Sector Counter

This eight-bit counter counts the sector pulses generated by the Sector Byte Counter. It is loaded with the contents of the Last Sector Register when the Index input occurs and is decremented with each sector. When the counter reaches a count of zero, the Sector Byte Counter is disabled until the next Index input. Reading this register gives the current rotational position of the disk.

3.6.2 Sector Byte Counter

This 16-bit counter is used to determine the starting location of sectors in the Hard Sector Mode and the starting location of sector mark searches in the Soft Sector Mode. The counter is loaded with the value contained in the Sector Length Register when Index Mark goes active. The counter generates an internal Sector Mark pulse, and is reloaded, every time it reaches a count of zero. The internal Sector Mark pulse decrements the Sector Counter. When the Sector Counter reaches a count of zero, the Sector Byte Counter is inhibited from further counting until the next Index Mark.

3.6.3 Field Counter

This two-bit counter counts the field mark pulses generated by the Field Byte Counter. It is loaded with the contents of the Last Field Register when the sector mark pulse occurs and is decremented with each field. When the counter reaches a count of zero, the Field Byte Counter is disabled until the next sector mark pulse.

3.6.4 Field Byte Counter

This 16-bit counter determines the length of the current field, it is loaded with the contents of the Field Length register selected by the Field Counter at beginning of a Sector and decremented by the Servo Clock signal (divided by eight). When the counter reaches a count of zero a Field Mark pulse is generated, incrementing the Field Counter and reloading the Field Byte Counter with the next field length value.

3.6.5 Segment Counter

This 2-bit counter is used to divide a field into its elements; Read Gate Delay segment, PLO segment,Sync Byte segment and Data segment.

3.6.6 Segment Byte Counter

This 16-bit counter determines the length of the various field elements. At the beginning of a field, it is loaded with the value contained in the Read Gate Delay register and decremented at the Byte Clock rate. The point at which the counter reaches zero marks the beginning of the PLO segment, and defines the Read Gate "turn on" time during read operations. The counter is then reloaded with the contents of the PLO Length register and begins decrementing again. The point at which the counter reaches zero marks the beginning of the Sync Byte. The counter is reloaded with the value contained in the appropriate Data Segment Length register and begins decrementing. The point at which the counter reaches zero is the end of the Data Segment and defines the Read Gate "turn off" time. After the Data Segment, the counter is inhibited from further counting until the next Field Mark.

3.6.7 Window Byte Counter

This eight-bit counter is reset at every sector or index pulse and establishes the window in which the SMF pulse must occur. In Soft Sector or Embedded Sector modes a "No Sector Mark Found" exception will be reported if a Data Control is active and no SMF pulse occus within the window.

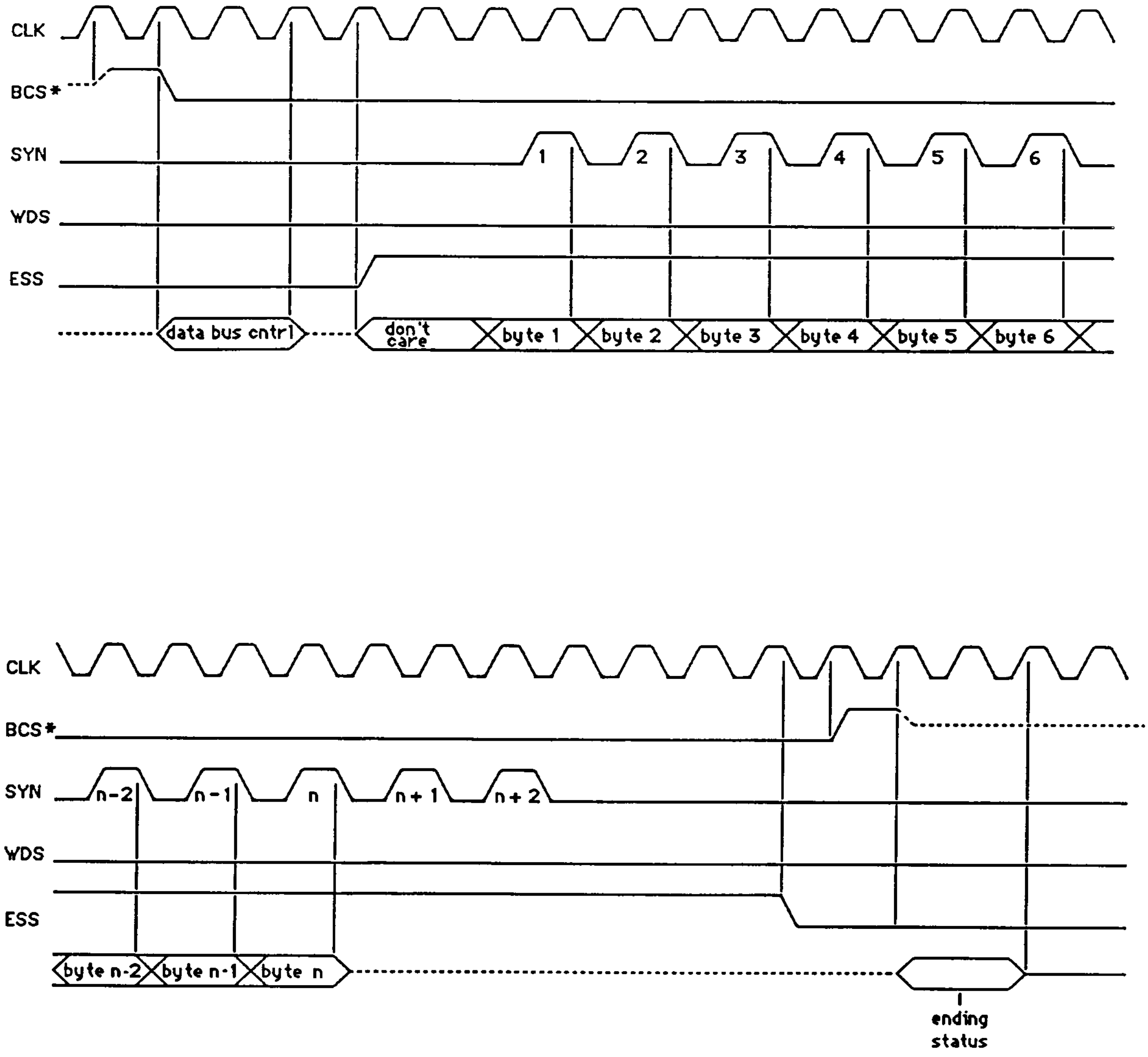


Figure 3
Serdes Bus Interface
Read Operation Example

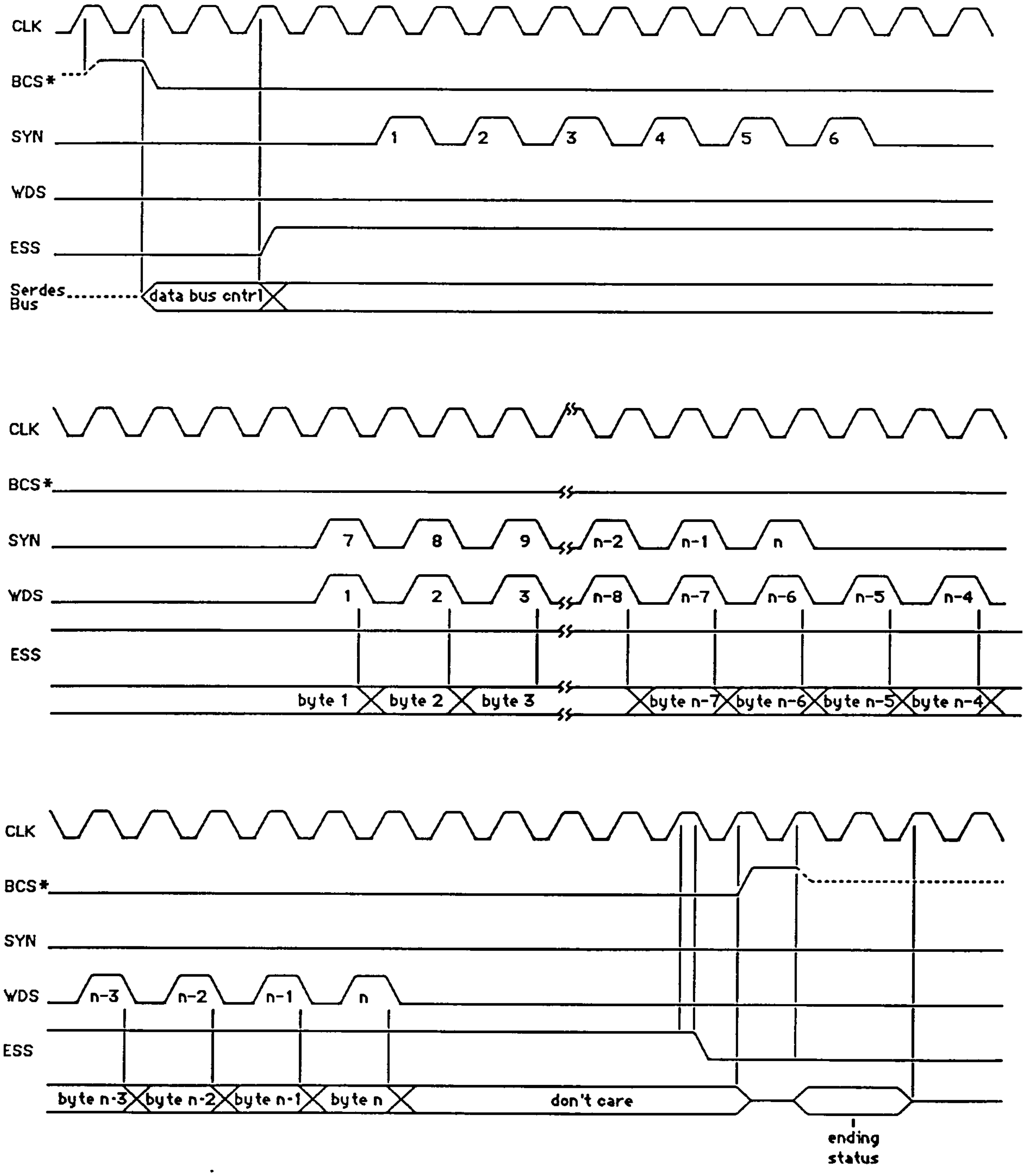


Figure 4
Serdes Bus Interface
Write Operation Example

4.0 OPERATIONAL DESCRIPTION

The Serdes/Formatter Circuit is responsible for interpreting and executing all Data Bus Controls received from the Interface Protocol Circuit. The drive processor is not involved in the execution of the Data Controls, and is only required to setup the mode of operation and initialize to the Formatter registers. The following sections provide a brief description of track format and the action taken in response to Data Bus Controls, under the three modes of operation.

4.1 Track Format

The surface of the magnetic media contained within a disk drive is divided into a fixed number of cylinders, depending on the drive type, manufacturer, model, etc. The individual cylinders are accessed through the movement of the head actuator assembly via seek commands. A cylinder is further divided into a fixed number of tracks, equal to the number of data read/write heads contained on the actuator assembly.

It is the responsibility of the Serdes/Formatter Circuit to divide these tracks into regions usable for the recording and subsequent recovery of user data. The SFC accomplishes this function by dividing the track into a programmable number of identical sectors. The start of each sector is established by either an internally or externally generated Sector Mark signal, depending on the operating mode established in the Control Register.

In Soft Sector or Embedded Servo Mode, the external Sector Mark signal marks the beginning of a sector. In Hard Sector Mode, the Sector Mark signal is internally generated by the Sector Byte Counter. Following the Index Mark Gap, the Sector Counter is loaded with the value contained in the Last Sector Register; the Sector Byte Counter is loaded with the value contained in the Sector Length register. The Sector Length Counter is then decremented by the internal Byte Clock signal; Byte Clock is simply the external Servo Clock signal divided by eight, re-synchronized at every Index pulse.

When the Sector Byte Counter reaches a count of zero, an internal Sector Mark Pulse is generated and the Sector Counter is decremented. If the Sector Counter has not reached a value of zero, then the Sector Byte Counter is reloaded and the process continues, generating internal Sector Mark pulses until the Sector Counter reaches a count of zero. Runt Sectors are therefore prevented.

The SFC further divides sectors into Data Fields (or simply Fields). The beginning of the first Field within a Sector is determined by the Sector Mark pulse; the beginning of the second and third Fields, if they exist, are determined by an internally developed Field Mark pulse generated by the Field Counter and Field Byte Counter. At the beginning of a sector, the Field Counter is loaded with the contents of the Last Field register, and the Field Byte Counter is loaded with the contents of the Field 0 Length register. The Field Byte Counter is then decremented by the Byte Clock signal.

When the counter reaches a value of zero a Field Mark pulse is generated, marking the beginning of Field 1, and the Field Counter is decremented. If the Field Counter has not reached a count of zero, then the Field Byte Counter is loaded with the contents of the Field Length register and the process is repeated. Up to three Fields can be established in this manner, each of a different length.

A Field is composed of six field segments and a field gap. Segments are of fixed length, or are programmable in length through the values contained in various registers of the SFC.
The first field segment is the Read Gate Delay segment and can be up to 256 bytes in length as determined by the value in the Read Gate Delay register. This segment is the same length for all fields. The purpose of this field is to separate the Write Gate "turn on" point, at the beginning of the field, from the Read Gate "turn on" point. This is necessary to prevent the Write Splice at the beginning of the field from drifting forward of the Read Gate "turn on" point, due to head scatter. The correct value for this segment is:

Read Gate Delay = (2 * Header Scatter) + Write Splice Length in bytes.

The next segment is the PLO Sync segment. This segment determines the number of bytes of zeros that will be output prior to the Sync Byte from the Read Gate "turn on" point. This segment is the same length for all fields. The purpose of this segment is to give the drive Read PLO a data pattern of known minimum length in which to acquire bit synchronization. The correct value to be programmed into the PLO Sync Length register is:

PLO Sync Length = 2 * Head Scatter + Minimum PLO sync time in bytes

The third segment in a field is the Sync Byte segment. This segment is one byte in length and is the same for all fields. The purpose of the Sync Byte is to provide a data byte of known value with which the Serdes can acquire byte synchronization. The value of the Sync Byte is programmable and is contained in the Sync Byte register.

The fourth segment in a Field is the Data Segment. This segment is individually programmable for each Field. User data is recorded and recovered from this segment. The segment length is determined by the values contained in the respective Field Data Length registers. The Read Gate "turn off" point is at the end of the Data Segment.

The fifth segment in a Field is the Pad Segment. This segment is of fixed value (all zeros) and length (two bytes). This segment serves to separate the last byte of User Data from the Write Gate "turn off" point, which is at the end of this segment. This is necessary to prevent the inactive going edge of Write Gate from truncating the User Data remaining in the drive write chain, and/or to prevent the ending write splice from effecting a flux transition shift in the last user data byte.

The sixth and last segment in a Field is the Gap Segment. This segment serves to separate a Field from the next Field in a sector. This segment is individually programmable, indirectly, for each Field within a Sector. The Gap Segment length is equal to the Field length less the Read Gate Segment length, the PLO Sync Segment length, the Sync Byte Segment length (fixed at one byte), the Data Segment length, and the Pad Segment length (fixed at two bytes).

This segment, at the very least, must be of a length sufficient to prevent the write splice following the Pad Segment from drifting forward into the next field. Therefore, the minimum Gap Segment length is:

Gap Length (min) = (2 * Header Scatter) + Write Splice Length in bytes

This Gap Segment may be required to have a length sufficient to allow for the write-to-read recovery time of the drive read chain, between the last Field in a Sector and the First Field in the next Sector; or it may be used to provide a decision period between any two Fields.

The maximum and minimum format parameter values are summarized below:

PARAMETER DESCRIPTION	MINIMUM	MAXIMUM
Sectors per Track	1	256
Fields per Sector	1	3
Bytes per Field	9	65536
Bytes per Read Delay Segment	1	256
Bytes per PLO Sync Segment	1	256
Bytes per Data Segment	6	65530

Note:
The Sync Byte Segment and Pad Segment are fixed at one and two bytes, respectively.

4.1.1 Hard Sector Mode

In this mode of operation, the start of each sector is determined by the internal Sector Mark signal generated by the Sector Byte Counter. Writing Address Marks is inhibited in this Mode. The Sector Mark input is ignored. See Figure 5.

4.1.2 Soft Sector Mode

The purpose of this mode is to minimize the impact of removable media disk drive head offset on storage efficiency by fixing the start of a sector to the media.

In this mode of operation, the start of a sector is determined by the active going edge of the external Sector Mark Found signal. The drive circuitry is expected to search for an address mark when the SFC asserts the Sector Mark Enable signal. When an address mark is detected, the drive circuitry asserts the Sector Mark Found signal. An address mark search begins at the end of the previous sector, if an operation was performed on that sector. Otherwise, the search begins from the leading edge of the internally generated Sector Pulse. The search ends when the Sector Mark Found signal goes active or when the Search Window Count is exhausted.

The Sector Counter is decremented by the internal Sector Mark pulse from the Sector Byte Counter. As a result, this internal Sector Mark pulse is always available and RPS remains functional during seeks in this mode. The internal Sector Mark signal is also used to "window" address mark searches when an operation was not performed in the previous sector. Writing Address Marks is only allowed in this Mode. Other than these differences, the Soft Sector mode is identical to the Hard Sector mode. See Figure 6.

4.1.3 Embedded Servo Mode

In this mode, the start of a sector is determined by the active going edge of the Sector Mark Found input signal. Writing Address Marks is inhibited in this Mode. The Sector Counter is decremented by the external Sector Mark signal, and the Sector Byte Counter is disabled. Other than these differences, the Embedded Servo mode is identical to the Hard Sector mode. See Figure 7.

4.1.4 SME Operation

Sector Mark Enable is used differently for the four Format Modes as defined in the following sections.

4.1.4.1 Mode 0 - Track Verify Mode

SME is not asserted during a Write Primitive. During a Verify Primitive it is asserted when the Sync Byte is found and deasserted when a mis-compare is detected or the operation ends successfully. This can be useful in scoping verify miscompares.

4.1.4.2 Mode 1 - Hard Sector Mode

In this mode the internal sector pulse is asserted on the SME output to enable viewing the track sectoring.

4.1.4.3 Mode 2 - Soft Sector Mode

In this mode, when a write primitive is not active, SME is active from the end of the previous sector until Sector Mark Found is detected for the current sector. It is used to enable sector mark (address mark) search.

When a Write primitive is active SME is active for three byte times each sector. The three byte long pulse is displaced from the internal sector pulses by the number of bytes in the Window Length Register (23). This is used to format a soft sector track with Address Marks.

4.1.4.4 Mode 3 - Embedded Sector Mode

In this mode, SME is active from the end of the previous sector until a Sector Mark is found for the current sector. It can be used, if necessary, to enable a search for the embedded sector mark.

4.2 DATA BUS CONTROL EXECUTION

A disk read/write operation begins when a Data Bus Control is received from the Interface Protocol Circuit. The SFC Command Sequencer latches and decodes the Bus Control in an internal register and initiates action on the appropriate sector fields.

4.2.1 Data Bus Controls

Data Controls specify the type and direction of a disk transfer, as well as the number of fields involved. A Data Control may also specify a qualification on the sector or field location, and whether a data head advance is to be performed upon successful execution.

There are two major categories of Data Controls: Sector Controls and Field Controls.

Sector Controls specify an operation on a sector having up to three data fields. Normally the first data field is considered a header.

Field Controls specify an operation on a single field or a pair of fields without regard for sector boundaries.

Data Bus Controls are IPI level-2 Bus Controls with bit 7 set, indicating a disk data transfer operation. Table 3 summarizes the coding of a Data Control.

Table 3 Data Bus Control Coding

Bit	Description
7	1 (indicates Data Control)
6	1 = Information In (Read) 0 = Information Out (Write)
5	0 (indicates Fixed Block Data Control)
4	Step Head Control
3	Header Field Operation
2	Modifier Bit Selects Target Sector for Read/Write Header operations Selects Verify Header for write operation Selects Skip Header for read operation
1	Data Field 2 operation
0	Data Field 1 operation

The above coding results in eight groupings of Data Controls plus one special control, as shown in Table 4 below.

Table 4 Data Bus Control Groups

Data Control	Description
80-83 91-93	Skip/Write Data Field
84-87 94-97	Verify Header, Write Data
88-8B 98-9B	Write Header, Write Data
8C-8F 9C-9F	Write Header, Write Data At Target
C0-C3 D1-D3	Skip/Read Data Field
C4-C7 D4-D7	Skip Header, Read Data
C8-CB D8-DB	Read Header, Read Data
CC-CF DC-DF	Read Header, Read Data At Target
90 Step Head	

A group of read and write Data Controls operate on the "target sector". The target sector is the sector addressed in the Target Sector Register.

4.2.2 Orientation With the Disk

It is necessary that the master's issuing of Data Controls stays oriented with the disk, and that no sectors or fields that are intended to be operated on, are missed. A Data Control that is not received in time for the slave to act on the next sector or field, following the previous Data Control, causes orientation to be lost and is rejected with a "Data Control Late" indication. To prevent the first Data Control of a series from being rejected when there is no orientation, orientation must be restarted. The restart is achieved by starting with a target type Data Control, or issuing a header type Data Control following a Command/Response Control.

Loss of orientation is not caused by a header mis-compare, or by a master-initiated termination of a transfer.

4.3 SURFACE VERIFICATION

The purpose of the track read/write primitives is to allow the drive processor to perform a simple surface verification at the time of manufacture. The surface verification process is directed by the drive processor, and not through the receipt of Data Bus Controls.

The drive processor verifies each track by executing Write and Verify Primitives available through the Control Register (see section 3.2.1). Any 5,6,7 or 8 bit repeating pattern can be written and then read back and compared using these primitives. If the readback data miscompares the operation is stopped and the displacement of the flaw can be determined.

Before verification is initiated, the drive processor should program the connected Interface Protocol Circuit(s) to reject all Data Bus Controls with a busy indication (Control Register bit 4) to prevent interference from a Controller. The Serdes/Formatter Circuit must be placed in the Surface Verification Mode (SFC Control Register bits 4 and 5 equal to zero), and a suitable format specified in the register set before verification is started.

The verification primitives operate on the same data format as normal Write or Verify Field operations issued via Data Bus Controls. The verification takes place on the Data Segment of the first field of the second sector on the track. This Data Segment can be close a full track in length and can start at any position on the track. The starting point is at the beginning of the second sector so that it can be moved to any position on the track by changing the Sector Length Register. The Sector Length Register can be programmed to any value between 1 and 65,535. The SFC suppresses Sector and Track Overrun conditions in this mode so the field being verified can exceed the sector length and can extend past Index. See Figure 8.

The Write Primitive writes zeros for the length of the Read Gate Delay and PLO segments, Writes the Sync Byte specified in the Sync Byte Register and then writes a Data Segment of the programmed length. The data pattern written can be any 5,6,7, or 8 bit pattern as dictated by the two modulo bits (2 and 3 in the Last Field Register) and the "seed pattern" in the Target Register.

During the Verify operation any miscompare will stop the operation and the residule count in the Segment Byte Counter (address 3D and 3F) will point to the byte in error. A residule count of zero indicates no miscompares. A residual count equal to the Data Segment length indicates no sync byte found.

When the processor has setup all the conditions in the drive required in preparation for verification pattern writing (offsets, strobes, etc.), the processor sets the Execute Write Primitive bit (SFC Control Register bit 7) and then writes zeros to Register Location 2E (as an additional interlock before allowing a write). The SFC will wait for the first field of the second sector and then begin a write operation. Read Gate Delay, PLO, Sync Byte and Data Segments are written on the disk. The data for the Data Segment is supplied from a pattern generator in the SFC. When the segment counts are exhausted the primitive bit is reset and test pattern writing ends.

To perform a verification of the track test pattern, the processor sets the required conditions in the drive (offsets, strobes, etc.), and sets the Execute Verify Primitive bit (SFC Control Register bit 6). The SFC will begin searching for the end of the first sector. When it is found, Read Gate will be asserted after a delay equal to the byte count in the Read Gate Delay register. The data recovered from disk will be compared with the contents of the pattern generator until the Data Segment count is zero or until a mismatch occurs. When the end of the sector is encountered, Read Gate is negated and the Execute Verify Primitive bit is cleared. Verification can be suppressed to start the process after a previously detected flaw by programming the Verify Skip Length Register.

To verify the unverified region between the last sector and Index, it is necessary for the drive logic to shift the offset (first sector length) and run the verification routine again.

4.4 Internal Diagnostics

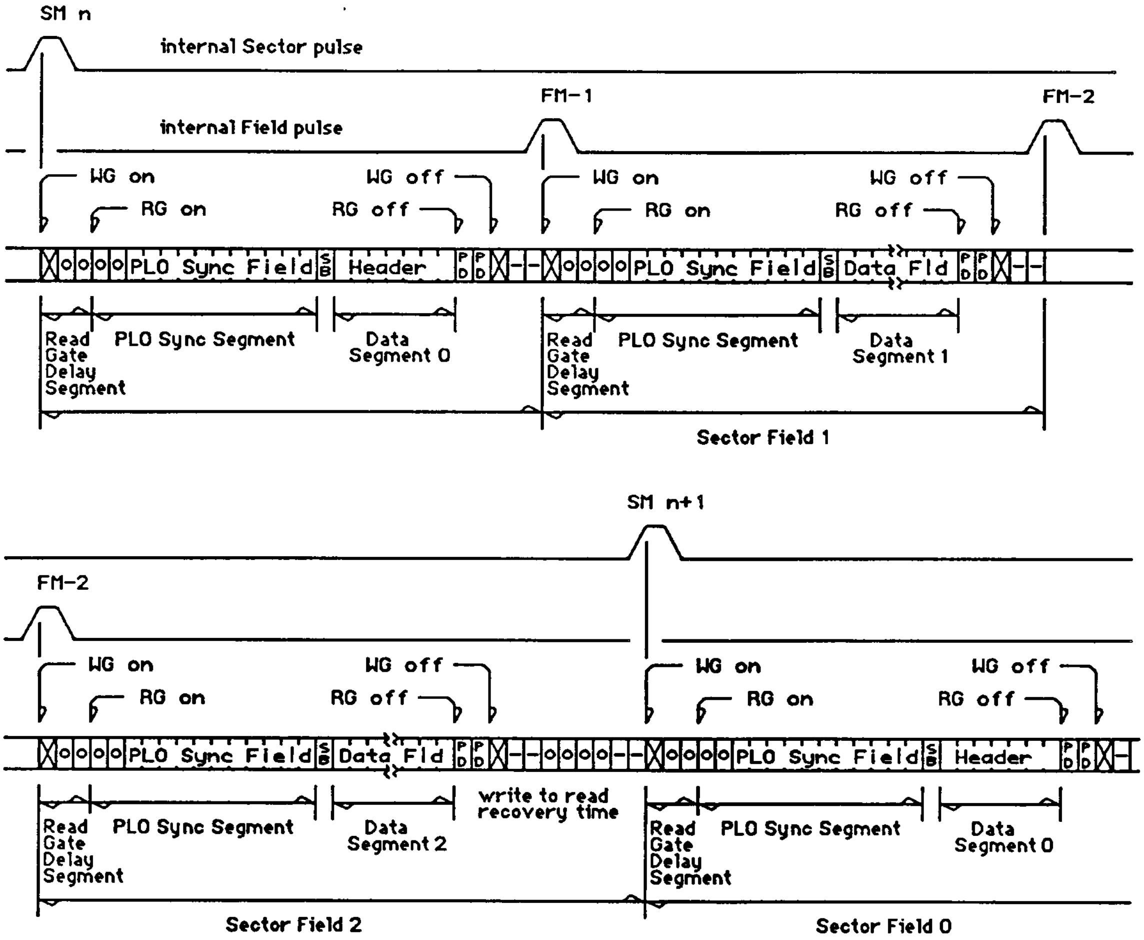
In order to provide a check of the SFC operation and to reduce the number of test vectors required during manufacturing test, circuitry has been added to enable testing of all of the SFC counters from the microprocessor port.

Diagnostic Modes 1 and 2 provide a means of loading and decrementing the Formatters counters. The "and" of the zero indicaters for all counters can then be read back in the Status Register to insure all counters are operational. In addition the comparitors for RPS and Verify Skip as well as the Verify Skip subtractor are checked during this test. The sequence of operation necessary to accomplish this is detailed in Table 5 below:

Table 5 Diagnostic Sequence

1) Target Register (22) = 00
2) Start RPS Register (2B) = 00
3) Last Sector Register (24) = 00

4) Field 0 Length Register (38) = 00
Field 0 Length Register (39) = 00
5) Field 0 Data Seg. Register (32) = 00
Field 0 Data Seg. Register (33) = 00
6) Window Length Register (23) = 00
7) Sector Length Register (27) = 00
8) Control Register (20) = 02
9) Write to 2D (Diagnostic Load)
10) Read Status Register (Bit 0 S/B 1)
11) Control Register (20) = 04
12) Read Status Register (Bit 0 S/B 1)
13) Write to 2D (Diagnostic Decrement)
14) Read Status Register (Bit 0 S/B 0)
15) Control Register (20) = 02
16) Read Status Register (Bit 0 S/B 0)
17) Start RPS Register (2B) = FF
18) Target Register (22) = FF
19) Read Status Register (Bit 0 S/B 1)
20) Control Register (20) = 14
21) Write to 2D (Diagnostic Decrement) 255 times
22) Read Status Register (Bit 0 S/B 1)
23) Skip Length Register (2A) = 00
24) Start RPS Register (2B) = 00
25) Control Register (20) = 02
26) Write to 2D (Diagnostic Load)
27) Control Register (20) = 00
28) Read Status Register (Bit 0 S/B 1)
29) Read Seq. Byte Counter (3E and 3F S/B 0000)
30) Control Register (20) = 02
31) Skip Length Register (2A) = 01
32) Write to 2D (Diagnostic Load)
33) Control Register (20) = 00
34) Read Status Register (Bit 0 S/B 0)
35) Control Register (20) = 02
36) Skip Length Register (2A) = 00
37) Field 0 Data Seg. Register (32) = FF
Field 0 Data Seg. Register (33) = FF
38) Write to 2D (Diagnostic Load)
39) Control Register (20) = 00
40) Read Status Register (Bit 0 S/B 1)
41) Read Seq. Byte Counter (3E and 3F S/B FFFF)
42) Control Register (20) = 02
43) Skip Length Register (2A) = FF
44) Start RPS Register (2B) = FF
45) Write to 2D (Diagnostic Load)
46) Control Register (20) = 00
47) Read Status Register (Bit 0 S/B 0)
48) Control Register (20) = 00
49) Read Status Register (Bit 0 S/B 0)



☒ Denotes a write splice, nominal position

⊟ Denotes a write splice, displaced by scatter

o Denotes a guaranteed zero byte

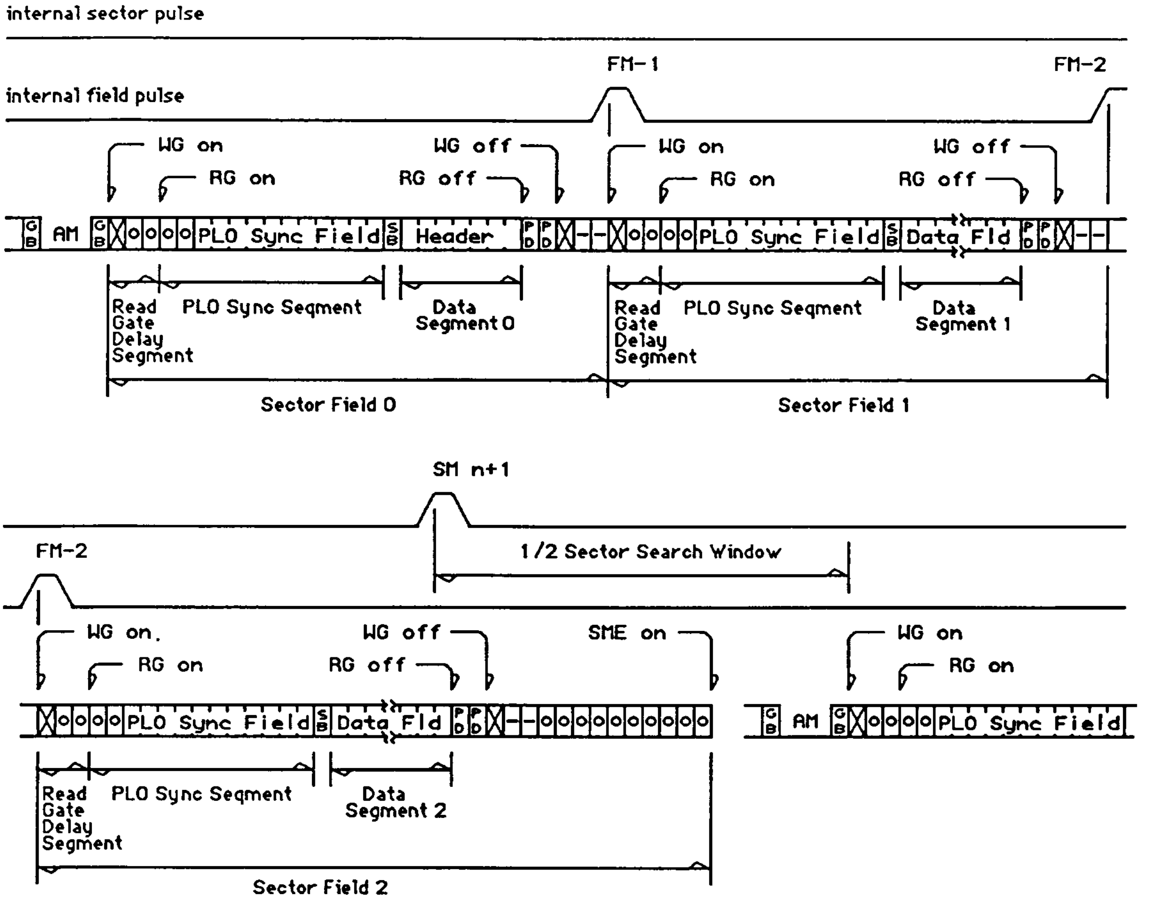
SB Denotes a Sync Byte

PD Denotes a Pad Byte (zero)

Note: Format illustrated for a drive with the following characteristics;

Write Splice Length	equals	1 byte
Head Scatter	equals	±1 byte

Figure 5
Hard Sector Format



☒ Denotes a write splice, nominal position

⊟ Denotes a write splice, displaced by scatter

0 Denotes a guaranteed zero byte

SB Denotes a Sync Byte

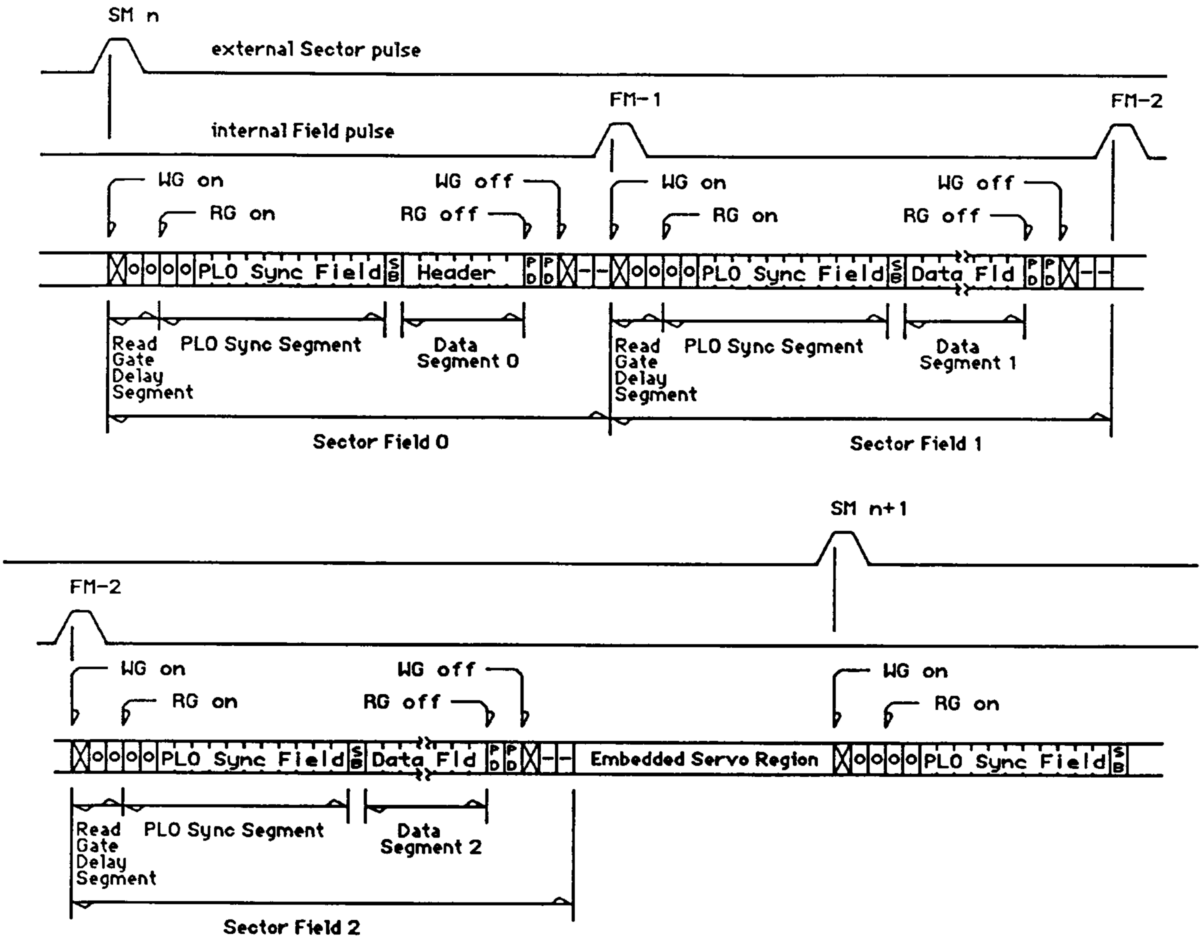
PD Denotes a Pad Byte (zero)

GB Denotes a Guard Band Byte (zero)

Note: Format illustrated for a drive with the following characteristics;

Write Splice Length	equals	1 byte
Head Scatter	equals	±1 byte
Head Offset	equals	±8 bytes

Figure 6
Soft Sector Format



☒ Denotes a write splice, nominal position

- Denotes a write splice, displaced by scatter

o Denotes a guaranteed zero byte

SB Denotes a Sync Byte

PD Denotes a Pad Byte (zero)

Note: Format illustrated for a drive with the following characteristics;

Write Splice Length	equals	1 byte
Head Scatter	equals	±1 byte

Figure 7
Embedded Servo Format

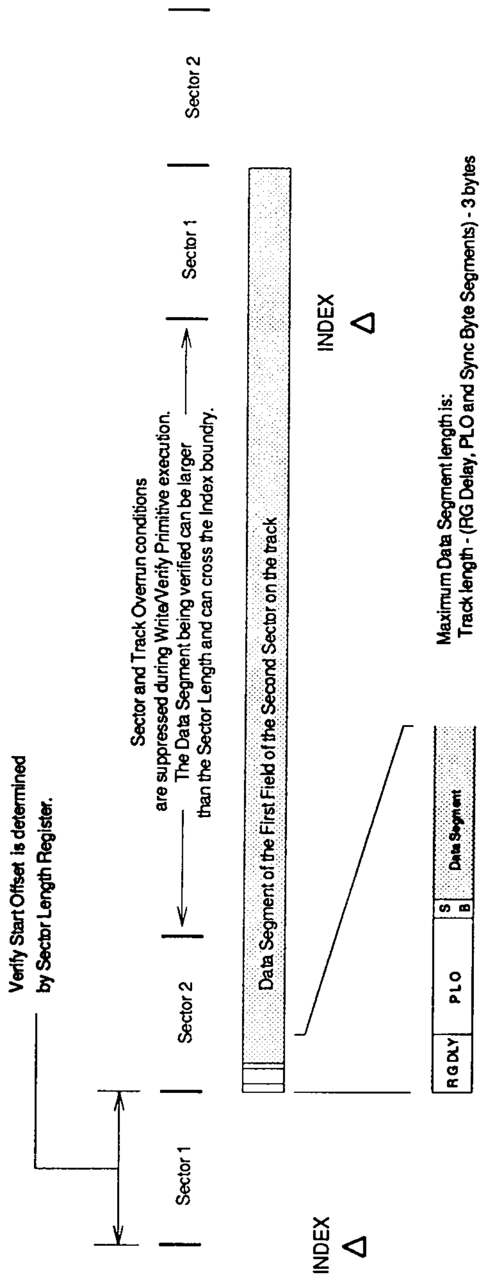

Verify Start Offset is determined
by Sector Length Register.
Sector and Track Overrun conditions
are suppressed during Write/Verify Primitive execution.
The Data Segment being verified can be larger
than the Sector Length and can cross the Index boundry.
Sector 1
Sector 2
Sector 1
Sector 2
Data Segment of the First Field of the Second Sector on the track
INDEX
INDEX
Maximum Data Segment length is:
Track length - (RG Delay, PLO and Sync Byte Segments) - 3 bytes
RGDLY
PLO
S
B
Data Segment
Primitives use normal data format

Figure 8
Serface Verification Example