AN-764
Application Note

A FLOPPY DISK CONTROLLER USING THE MC6852 SSDA AND OTHER M6800 MICROPROCESSOR FAMILY PARTS

Prepared by:
Larry A. Parker
Semiconductor Systems Engineering

This application note describes a floppy disk controller based on the M6800 family of parts. It uses the Synchronous Serial Data Adapter (SSDA) as the primary data interface with the MPU and does not require DMA for transfer of data to and from memory. A Peripheral Interface Adapter (PIA) controls all non-date related operations in the controller (including seek, drive selection, etc.).

MOTOROLA Semiconductor Products Inc.

A FLOPPY DISK CONTROLLER USING THE MC6852 SSDA AND OTHER M6800 MICROPROCESSOR FAMILY PARTS

INTRODUCTION

With the introduction of the MC6852 SSDA, the task of interfacing synchronous serial peripherals such as floppy disks, tape cassettes or cartridges, and bi-sync or HDLC data channels, has been reduced significantly.

Described in this application note is an efficient and flexible floppy disk controller design. Various features of this design include:

- Controller operates one to four daisy-chained drives
- Four drive radial configuration possible with additional multiplexing
- Flexible drive interfacing
- MPU controls data transfer allowing:
  - Store only desired data from a sector into memory
  - Search disk for pattern match without transferring data into memory until pattern is found
  - Read or write entire track in one revolution; consecutive tracks on consecutive revolutions
- DMA not required when using host MPU
- Interrupt MPU system operations on address mark match to start operations, allowing increased throughput
- Seek interlaced with R/W when using radial configuration
- Hard or soft sectoring
- IBM or user programmable sync patterns and format
- Write format blank disks
- Cost competitive
- Effective use of MPU leaves time available for additional tasks (see Table 1).
- Low parts count

Controller: (MPU and RAM shared with system)

Formatter	14 TTL SSI, MSI Devices 1 SSDA 1 PIA 1 CRCCG
Data Recovery	5 TTL + filter SSI, MSI Devices
Drive Interface	
Buffers and Receivers	5-10 TTL/CMOS Devices + Termination
MPU System Interface	2-5 TTL/Three-state Devices

The disk controller system consists of four basic blocks as shown in Figure 2. The PIA serves as the interface to the drive controls. There are 16 available PIA lines which allow a wide variety of drive configurations. The remaining four lines are used internal to the controller. The clock is separated from the raw disk data by the phase-locked loop data recovery block. The SSDA has the responsibility of synchronizing read/write operations and serializing/deserializing the data. Error detection and system clock functions are performed by the CRCC and clock control logic block.

The MPU has essentially complete software control over the system. Mechanical drive functions and status such as step, step direction, head load, ready, write enable, etc. are controlled and monitored in software by the MPU via the PIA. SSDA data transfer operations are initialized and supervised with MPU instructions. Due to the PIA, SSDA and system hardware configuration, programming can be kept simple and effective with a minimum of software overhead. Basic driver routines can be

TABLE 1

Function	Conditions IBM Format	Microprocessor Processing Time Available for Non-Floppy Operations
Consecutive Sector R/W on Multiple Tracks	Processing non-floppy operations allowed only between sectors	1 ms between sectors = 25 ms 21.6% 11 ms at Index
Read or Write a Single Sector	Processing non-floppy operations allowed while R/W the sector; a 44 μs R/W loop is assumed for 2 bytes of data	52 μs block available each 192 μs 43.7% 42 * 52 μs = 2.184 ms
Consecutive Sector R/W on Multiple Tracks	Processing non-floppy operations allowed while R/W a sector and between sectors as above	See above 65.3%
Search for Sector	Assume 250 μs to Read and Test ID block for match after Sync Interrupt	1.00 - 3.9 ms/rev 96.1%
Search for Track	Assume 50 μs to process track info for each step	1.00 - 50 μs/ 167 ms 99.97%

FIGURE 1 – M6800 Floppy Disc Controller Schematic

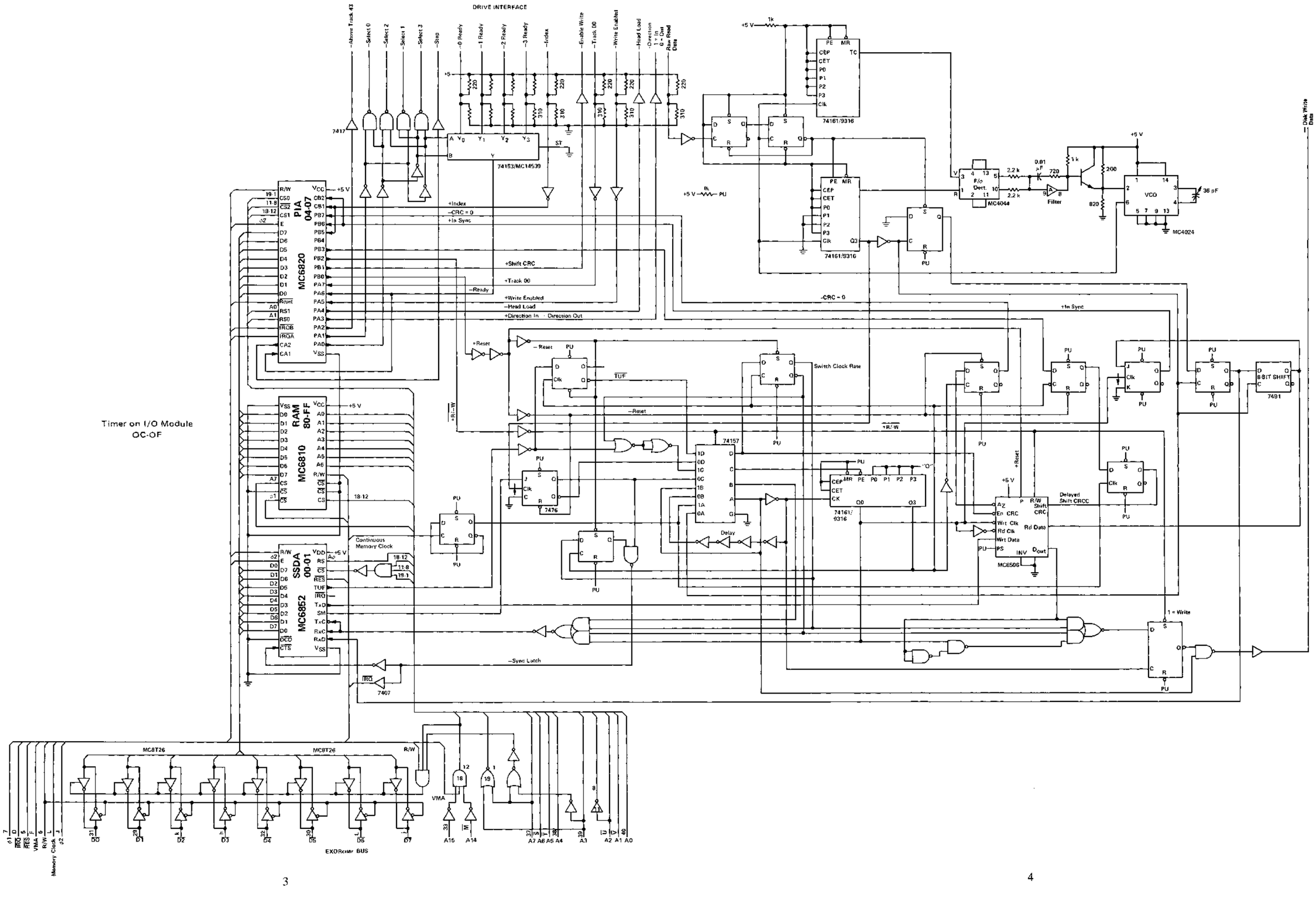
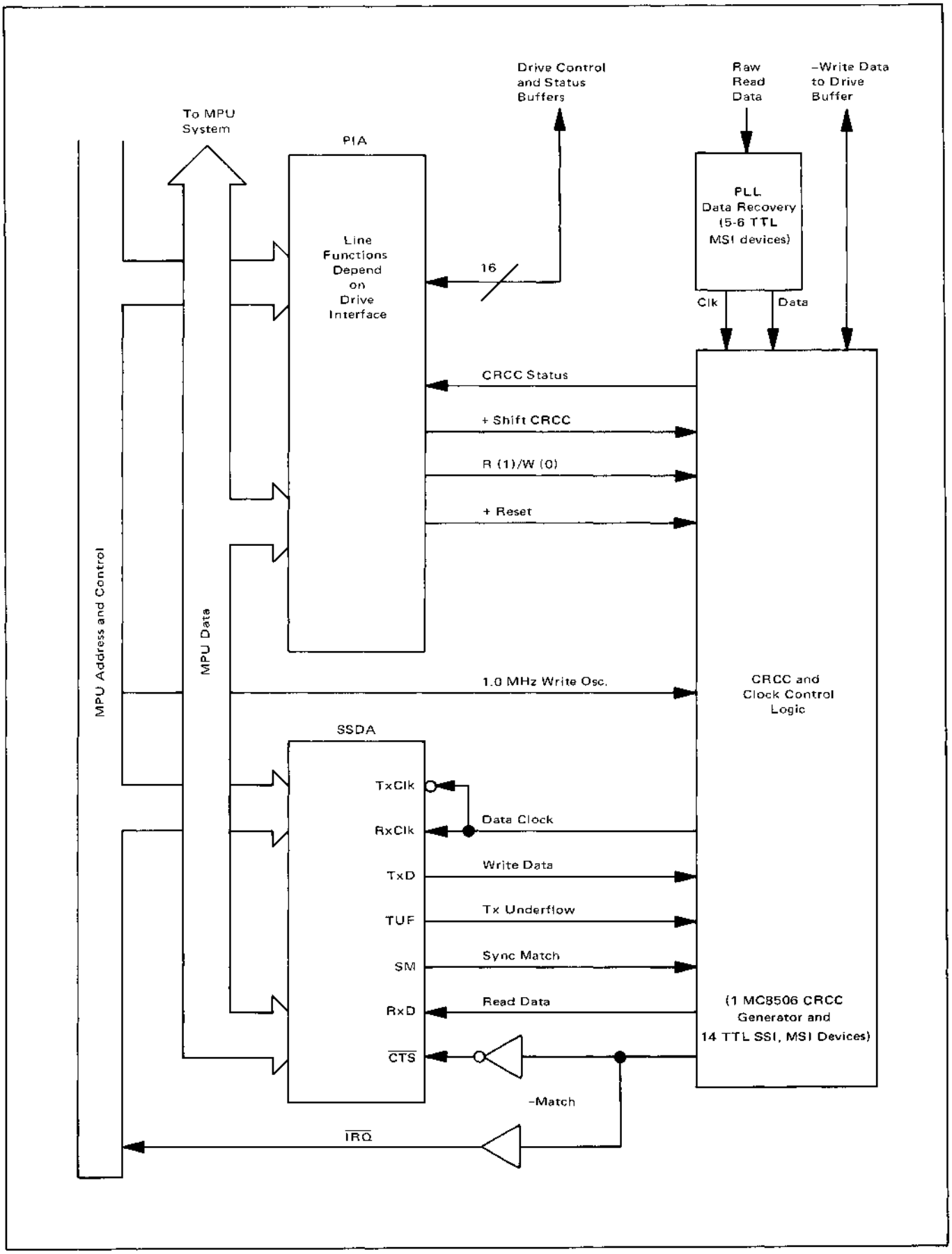


FIGURE 2–Floppy Disk Controller Block Diagram



written with fewer than 600 bytes of code. Operating systems suitable to most user needs can be done within two to four kilobytes.

Specific descriptions of the data recovery circuit and read and write operations are discussed in the following pages. Simplified logic diagrams are used in the circuit descriptions. The actual system schematic is shown in Figure 1.

It is important to be familiar with the operation of the SSDA, PIA and the IBM 3740 format in order to understand the controller design. A review of the MC6820 and MC6852 data sheets is recommended. A discussion of drive interfacing and IBM format can be found in the *M6800 Microprocessor Applications Manual*. A description of the software drivers and the software for the controller is available upon request.

DATA RECOVERY CIRCUIT DESCRIPTION

The raw data from the drive (clock and data) is terminated and buffered before clocking the first D flip-flop (Figure 3-B and Figure 4-B). Flip-Flops 1 and 2 generate a negative pulse 1 VCO period wide (Figure 3-D and Figure 4-D) which is used to load the reference counter with 9 and to set the data flip-flop 3.

IBM 3740 data can have only one consecutive pulse missing in the stream. By loading the reference counter with 9, Q3 will have a positive transition within 15 VCO periods generating a clock edge even if the data pulse is missing (Figure 3-E and Figure 4-E). Carryouts will occur every 2 μs ($16/f_O$), nominally providing a fundamental reference for the frequency/phase detector (Figure 3-F and Figure 4-F). The variable input to the frequency/phase detector is generated by dividing f_O by 16, using the carryout to give a pulse similar in duration to the reference.

Negative transitions on Q3 are inverted and clock flip-flop 3, whose output goes low (Figure 3-G, H and Figure 4-G, H). If a data pulse is present, the flip-flop is set by pulse from flip-flop 2 (Figure 3-D, Figure 4-D). If no data pulse is present, the output of flip-flop 3 remains low until set by a data pulse which must occur within 32 μs of the last one. The output of flip-flop 3 is then clocked one Q3 period later by Q3 to generate the NRZ data required by the formatter circuitry (Figure 3-J, Figure 4-J).

The 8-bit shift register provides 16 μs delayed data which is fed to the CRCCG. The SSDA clocks in 8 bits of data at 500 kHz before sync occurs and the read operation starts; because the sync data is included in the CRCC permutation, this sync data must be included in the CRCC field.

Phased-locked loop design is described in Motorola Application Note AN-535.

READ OPERATION

The sync code register of the SSDA is used to synchronize read operations by testing the incoming data stream, clocked at 500 kHz (2X clock), for the first half clock and data pattern of the desired address mark. When a match is found, the external circuitry is released by the Sync Match (SM) output and the second half of the address mark (clock and data) is read from the SSDA Rx FIFO (when it becomes available) and tested for a match with the desired type. If it does not match the sequence is restarted. If the second half of the address mark matches, the desired data transfer is initiated. The external circuitry switches the SSDA read clock to 250 kHz (1X clock) after the second half of the address mark has been received so that only the data portion of the remaining Rx FIFO information is recovered. The external circuitry also controls the CRCC generator (CRCCG) timing so that only the data portion of the recovered information is clocked into the generator.

After the data block has been transferred, the CRCC status is made available to the MPU for 32 μs at a PIA peripheral line.

READ DATA LOGIC

Figure 5 is a simplified logic diagram of the read data logic. Figure 6 is a timing diagram which shows the signal timing relationship when a read operation is begun.

This explanation of the read data logic assumes that system initialization has been completed. This includes the completion of the seek and head load operation. The enable read line is set and the formatter reset line has been toggled to reset the sync match latch and set the switch clock rate latch. These two previous operations are initiated in software and are executed via the system Peripheral Data Adapter (PIA). Initialization of the Synchronous Serial Data Adapter, SSDA, has been completed as described in the SSDA Read Preparation section.

Raw serial data is processed by the data recovery circuit which provides the separated read data and 500 kHz clock to the read data logic, Figure 5-A, B and Figure 6-A, B.

The 500 kHz clock from the data recovery circuit is inverted, delayed and then fed to the SSDA read clock input (RxD), via combinational AND/OR selector logic controlled by the switch clock rate latch, Figure 5, Figure 6-C. Inverting the clock provides the correctly phased positive transition to load the read data in the SSDA receiver shift register. The delay (4 inverters) is necessary to prevent a possible timing glitch which will be discussed later. The 500 kHz, 2X clock rate will load the receiver register with both clock and data bits from the read data line.

When the bit pattern loaded in the receiver shift register is equal to the pattern present in the SSDA sync code register, the SSDA synch match output, SM, will go high for one read clock period, Figure 6-D. When the sync match occurs, the SSDA receive data FIFO is internally enabled and will begin to store the read data, Figure 6-D.

FIGURE 3—Data Recovery Circuit

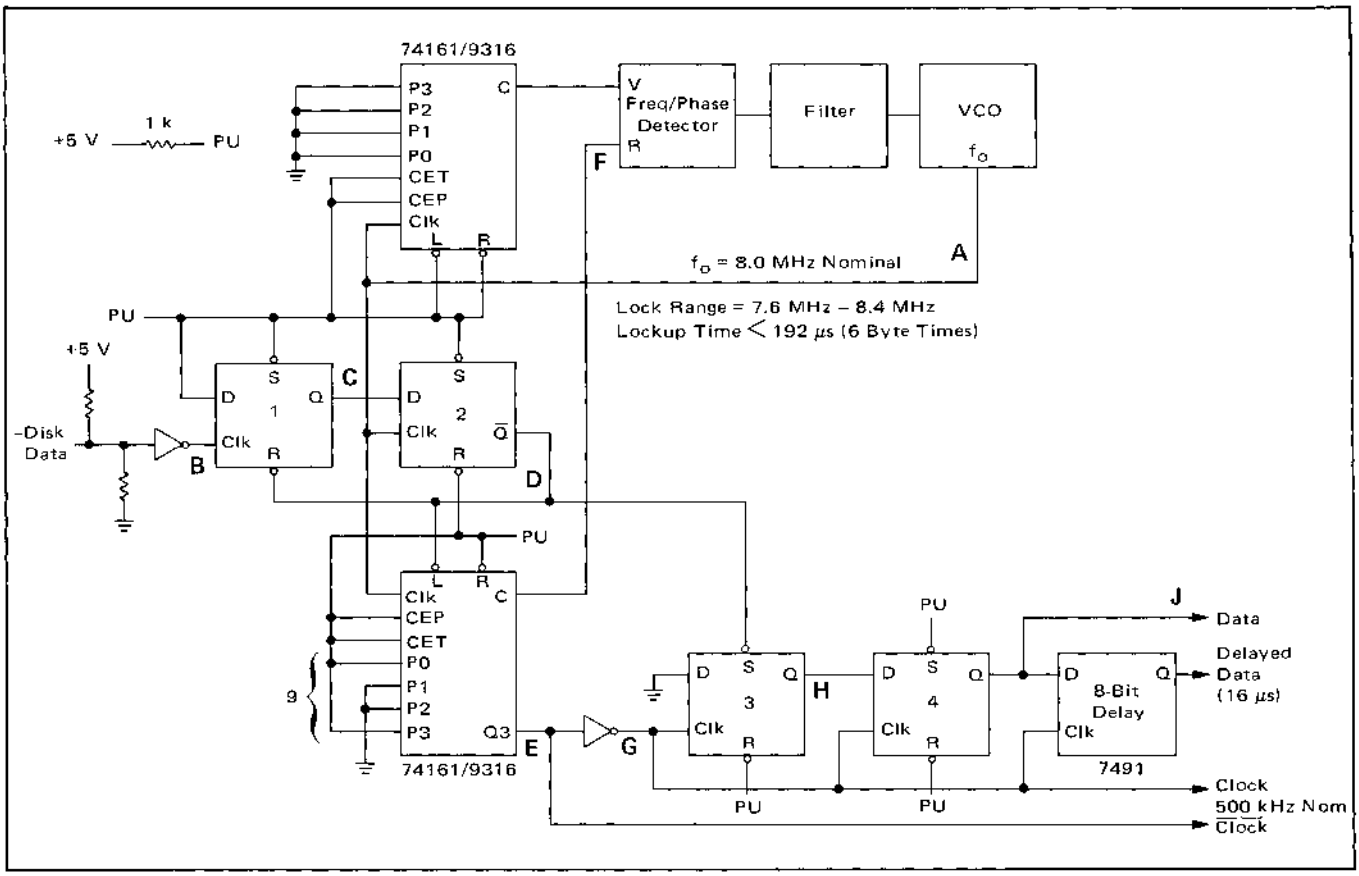


FIGURE 4—Data Recovery Timing

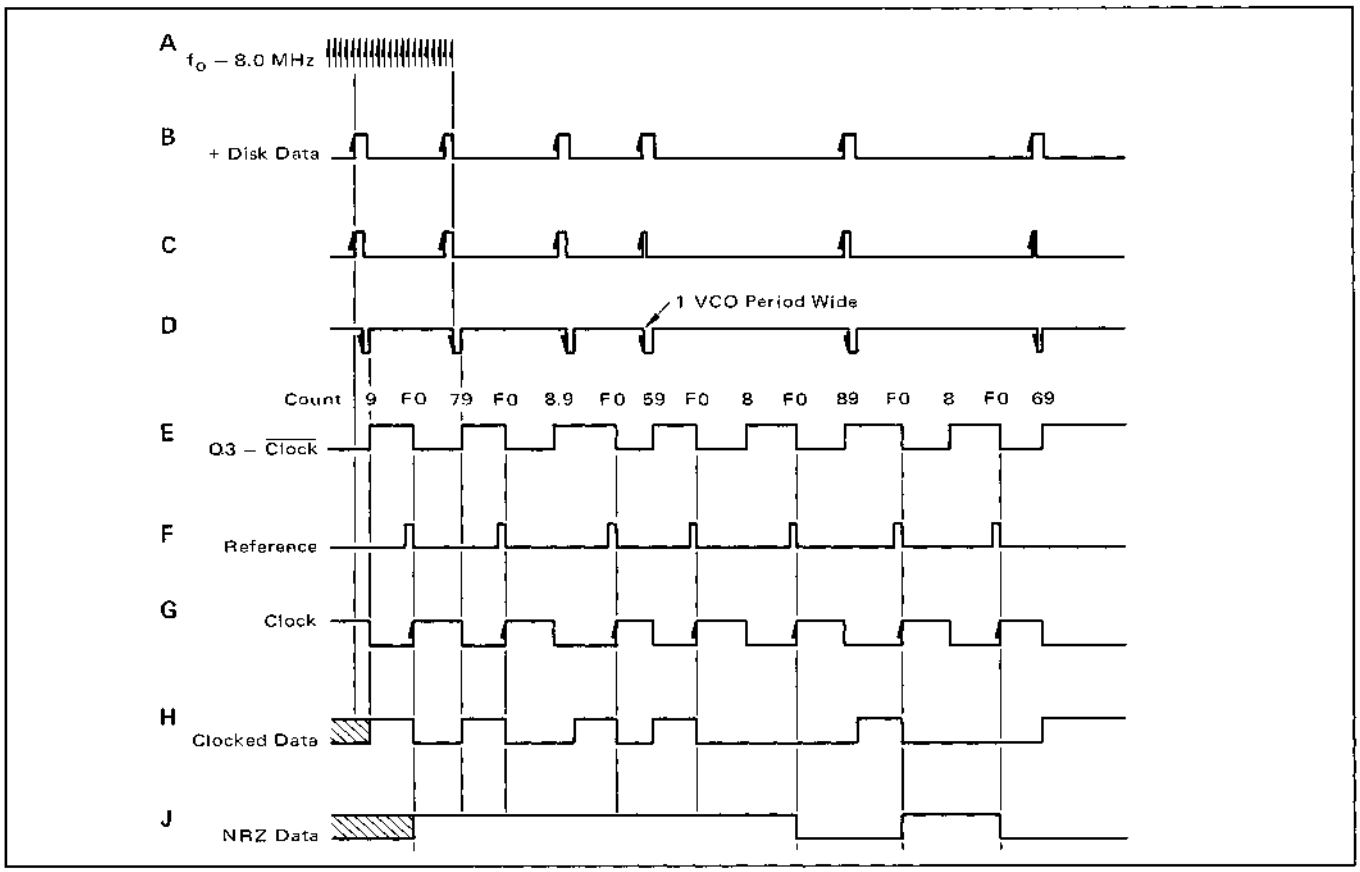


FIGURE 5—Read Data Simplified Logic Diagram

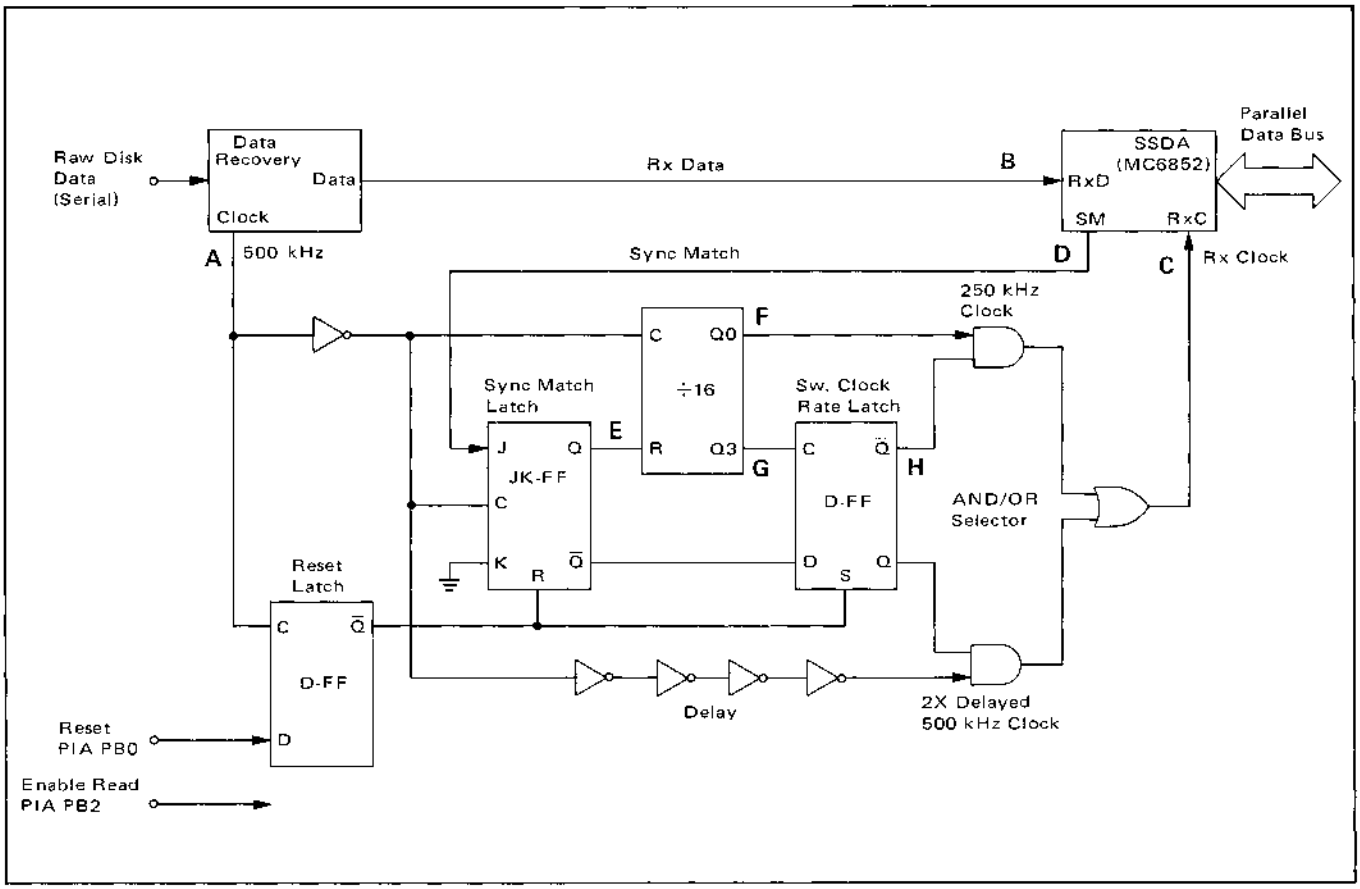
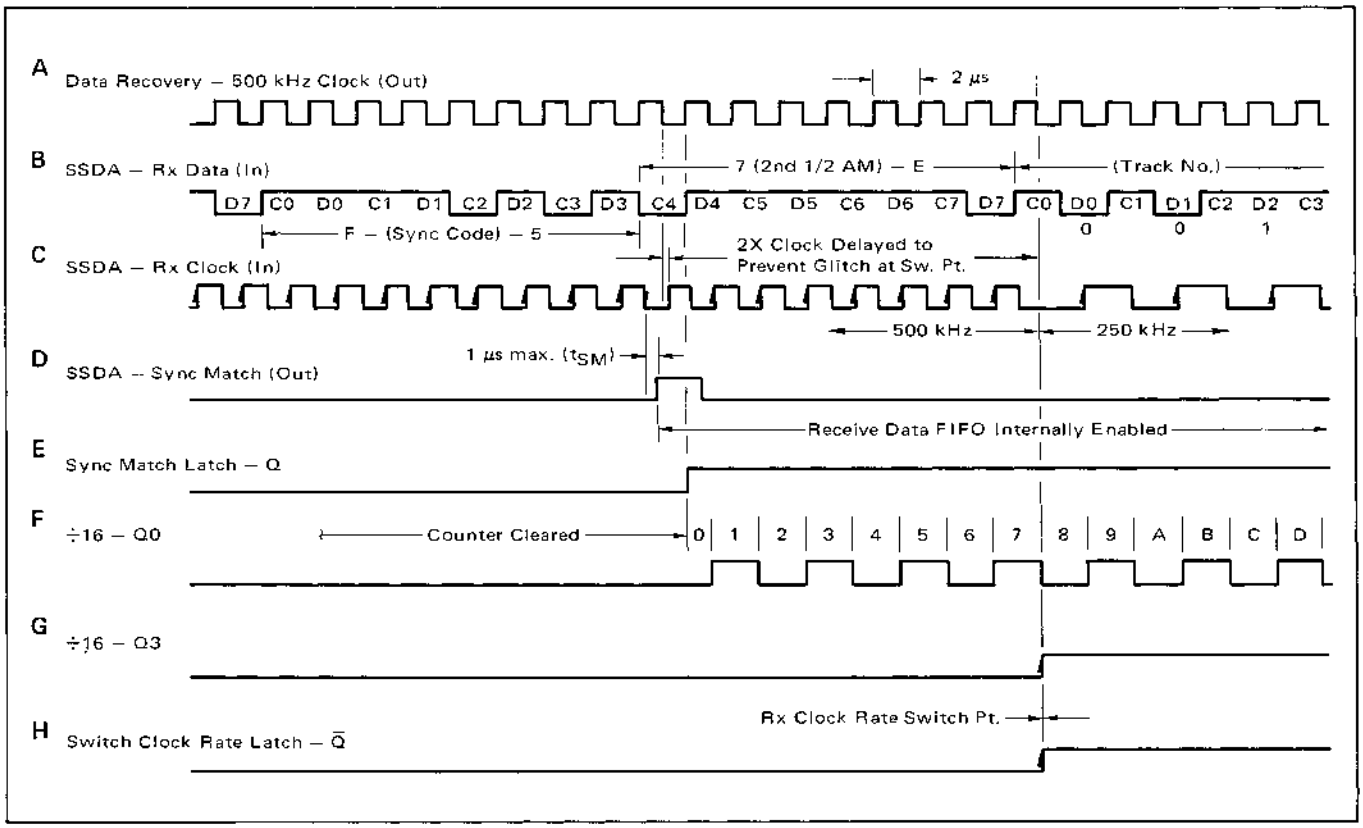


FIGURE 6—Read Data Logic Timing Diagram



The clock and data bit pattern used for sync match with the IBM 3740 format is a hex F5. The example in Figure 6-B shows the ID address mark which has the clock and data bit pattern of Hex F5 7E. Sync match will also occur with the data address mark which is Hex F5 6F. The sync code F5 portion of either address mark will not be stored in the receiver FIFO. The second half of the address mark, 7E or 6F will be the first byte of data stored.

The first positive transition of the 500 kHz clock occurring while the sync match output is high will set the sync match latch, Figure 5-E, Figure 6-E. The Q output of the sync match latch will enable the ÷16 counter. After eight counter counts, Figure 6-F, Q3 of the ÷16 will reset the switch clock rate latch, Figure 6-G, H. As mentioned previously, the clock rate latch controls the AND/OR selector and, at this time, the 250 kHz clock rate, ÷16-Q0, Figure 6-C, is selected and fed to the SSDA read clock. Because the ÷16 counter and clock rate latch are both synchronized with the 500 kHz clock, the delay in the 500 kHz read clock is necessary to guarantee that the AND/OR selector is switched before the next positive transition of the 500 kHz read clock at the clock rate switch point, Figure 6-C. After the clock rate has been switched, the delay is no longer needed. The read data is now being clocked into the SSDA receiver register at a 250 kHz rate so that only the data bits will be loaded. The eight count delay between the sync match and clock rate switch point allows the second half of address mark to be clocked into the SSDA receive data FIFO register at the 2X clock rate.

The receive data FIFO will now continue to fill with data bits clocked in at the 250 kHz read clock rate. As described in the MC6852 SSDA data sheet, the Receiver Data Available (RDA) bit in the SSDA status register will be high when data is available in the last 2 locations of the Rx Data FIFO. The read data can now be read from the SSDA via the system parallel data bus.

The read operation will continue at the 250 kHz read clock rate until terminated or reset with software.

CRC READ ERROR CHECK LOGIC

Figure 7 is a simplified logic diagram of the read data logic. Figure 8 is a timing diagram which shows the signal timing relationship when a CRC read operation is begun.

This explanation of the CRC read logic assumes that the read operation is initialized and is running as described in the Read Data Logic discussion. The reset latch has been toggled which presets the MC8506 CRCCG, and resets the sync match latch. This is done with software via the PIA. Familiarity with the MC8506 Polynomial Generator (Cyclic Redundancy Check Character Generator) data sheet is assumed.

Separated data and the 500 kHz clock are provided to the CRC logic from the data recovery circuit, Figure 7-A, B and Figure 8-A, B. These data and clock signals are the same as those received by the Read Data Logic, Figure 5-A, B and Figure 6-A, B, except that they are delayed 16 μs by the eight-bit shift register.

When SSDA sync code match occurs, the SSDA sync match line goes high, Figure 7-C, Figure 8-C. The first positive edge of the 500 kHz clock during sync match sets the sync match latch which enables both the ÷16 counter and the MC8506 CRCCG, Figure 7-D, Figure 8-D, E, F. The 250 kHz clock ÷16 $\overline{Q0}$, is now fed to the CRCCG, Figure 8-E.

At the 250 kHz clock rate, only the data bits from the read data are loaded into the CRCCG. The data presented to the CRCCG is delayed eight bits (four data bits), behind the read data, Figure 8-B, G. This allows the CRCCG to receive the first half of the address mark which occurs just before the sync match and before the CRCCG is enabled. The first half of address is included in the cyclic permutation of data bits which generate the two CRC bytes. Two CRC bytes append every ID and data field.

If the complete address mark and ID or data field has been read correctly, the CRCCG $\overline{\text{All Zero}}$ line will go low after the last CRCC byte for that field has been read, Figure 8-H, G. The positive transition of the ÷16 $\overline{Q3}$ output will reset the CRCC = 0 latch, Figure 8-F, H, I. The CRCC = 0 latch Q output will remain low until clocked again one byte time later by ÷16 $\overline{Q3}$, Figure 8-F, I. The software test for a CRCC error must be made during that one byte time which immediately follows the last CRCC byte. (The CRCC = 0 latch Q output is read by the software through the PIA). If a detectable read error occurs, the $\overline{\text{All Zero}}$ line will not go low and the CRCC = 0 latch will remain high during the one byte test time.

After completing a CRC check of a single ID field or data field, the CRC read error logic must be reinitialized before reading the next field by pulsing the Formatter Reset line.

FIGURE 7–CRC Read Error Check–Simplified Circuit Diagram

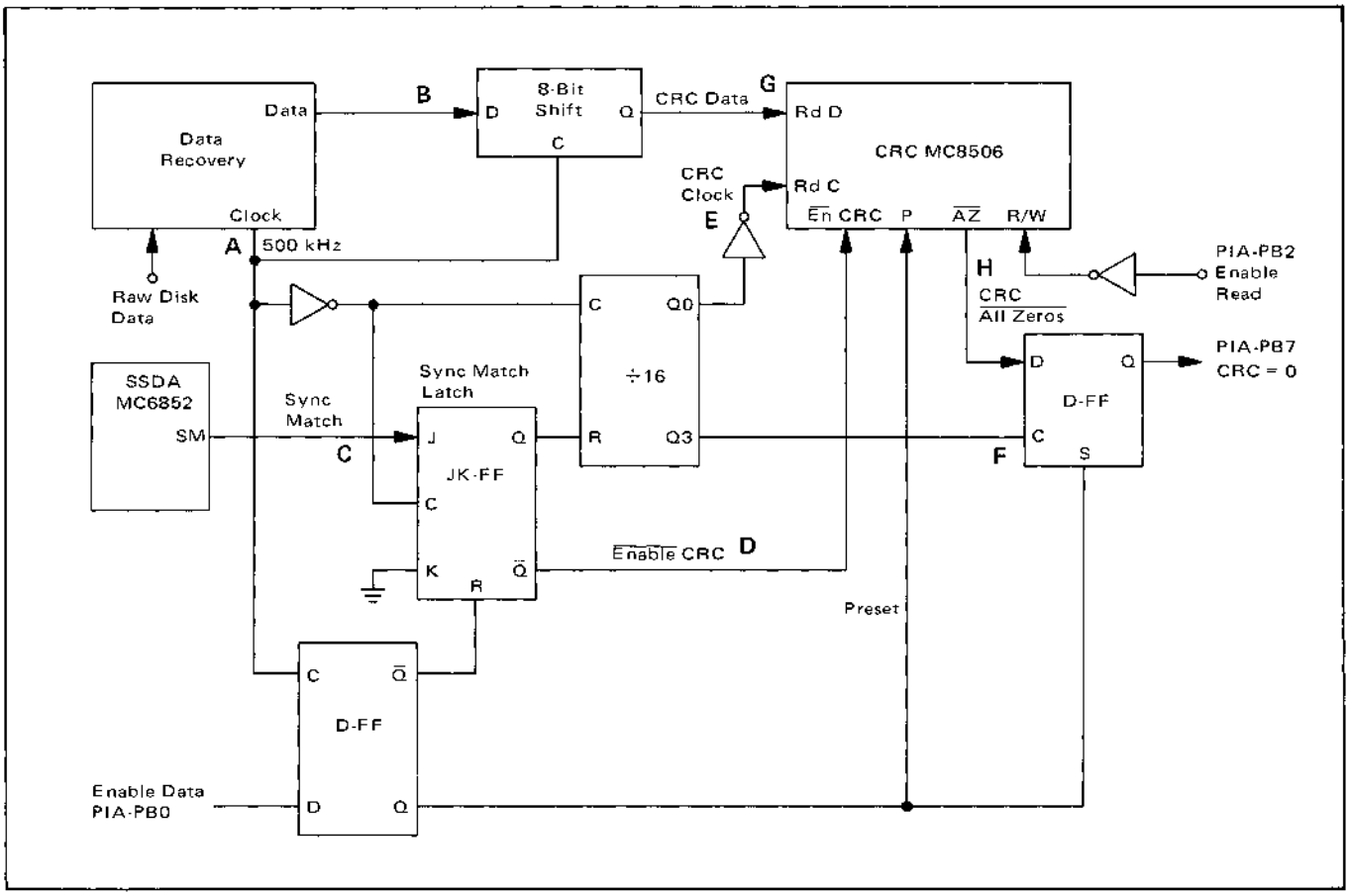
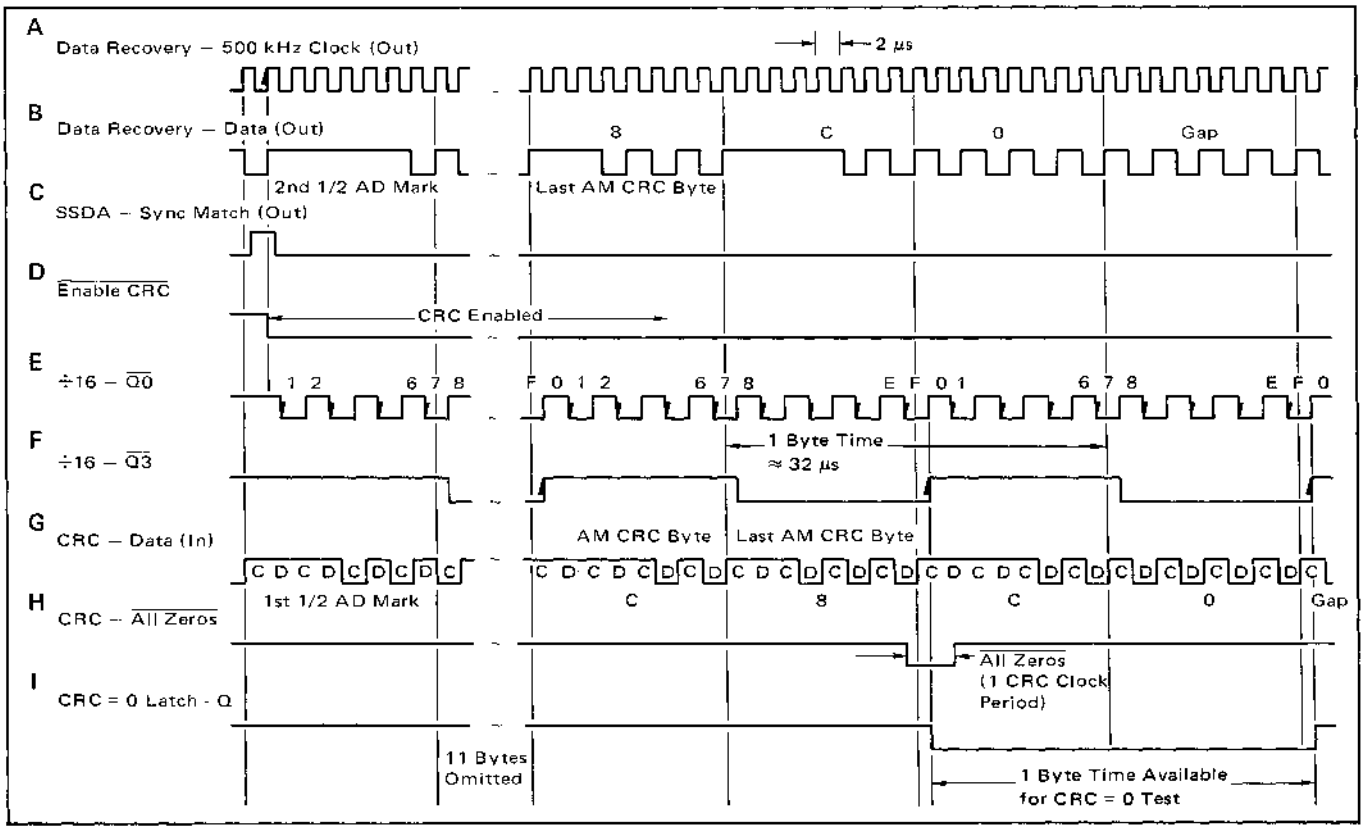


FIGURE 8–CRC Read Error Check–Timing Diagram



SSDA PREPARATION FOR READ

Table 2 summarizes the necessary sequence of SSDA register programming steps for a read operation. A further explanation of SSDA Register programming is summarized in Table 1 of the SSDA data sheet.

In this particular system, hardware chip selects with direct addressing are used to access the SSDA. Specifically, writing into hex address 00 will select Control Register 1, CR1. Writing into the next address, hex 01, will access the CR2, CR3 or the Sync Code Register as selected with CR1. The SSDA Status Register is read by reading from hex address 00. Data is read from hex 01. As described in the SSDA data sheet, the Sync and Control Registers are write only. Status and Data Registers are read only.

In Table 2, Step 1, SSDA Control Register 1, CR1, is addressed and set to inhibit Receive, Transmit, and Sync, and CR3 is selected. Step 2 loads CR3 to prepare the SSDA for the one character internal sync mode. Step 3 returns to CR1 and selects the Sync Code Register. The sync code hex F5 is loaded into the Sync Code Register in Step 4. These first four steps are required only once per read operation.

Steps 5 through 11 must be carried out before each new field is read. Step 5 sets CR1 to select CR2. Step 6 then loads CR2 to prepare the SSDA for eight bit word transfer, two byte RDA, and to inhibit the sync match output. Step 7 enables the receiver. In Steps 8 and 9, PIA Data Register B is addressed and set to enable read and toggle the read logic reset latch. Step 10 enables the internal sync and selects CR2. The sync match output is enabled in Step 11.

In Step 12, the SSDA Status Register is read and the RDA bit is tested. A high RDA indicates two bytes of data are ready and can be read from the Data Register as in Step 13.

The SSDA must be programmed in the proper sequence to avoid several non-obvious errors. A combination of the receiver reset mode and reading gap can cause a false sync match if the receiver is not enabled before the sync. The Receiver Shift Register, when reset, is actually set to all ones or hex FF. Gap read at the 2X rate will appear as alternating zeros and ones or hex 55. If a half byte of gap is clocked into the Receiver Shift Register, the contents of the register will be hex F5, the sync code. Enabling the receiver before enabling the sync allows hex FF to be clocked out of the register while sync is inactive. The reset latch must also be toggled before enabling the sync. This switches the read logic back to the 2X read clock and prevents a possible false sync match with data at the 1X rate.

WRITE OPERATION

The transmitter underflow (TUF) output is used to synchronize write operations by resetting the external ÷16 bit counter while writing the pre-address mark gap from the sync code register at 500 kHz (2X clock). After counting 11 TUF's, 5-1/2 bytes of gap, the first half of the desired AM is stored in the Tx FIFO.

When the first half of the address mark (C & D) enters the Tx Shift register, no TUF output will occur releasing the external hardware sequence. The second half of the address mark (C & D) is then stored in the Tx FIFO, followed by the data to be transferred to the disk. The external hardware switches the SSDA Tx Clock to 250 kHz (1X clock) after the address mark is written and clocks the data portion of the information into the CRCCG. When the data transfer is complete, two dummy bytes are stored in the FIFO while the Frame Check Sequence (FCS) is appended by the CRCCG on command from the MPU via a PIA peripheral line. The Sync Code Register is loaded with the postamble which will be written at 1X clock after the FCS has been appended and the first TUF has occurred. The sync code register is then loaded with the 2X gap clock and data pattern which will be written after the second TUF which switches the SSDA clock back to the 2X mode. Write current to the drive is controlled by the MPU via a PIA peripheral line and is initialed at the start of the pre-address mark gap and terminated after the postamble byte for all but write format operations where it is held on for the entire track.

The Formatter Reset line must be pulsed prior to the next formatter operation to initialize the sequencer logic. During write format operations, this will not cause a gap glitch since the formatter has already switched back to the 2X clock.

WRITE DATA LOGIC

The synchronization of the write logic with the SSDA is accomplished by the transmitter underflow (TUF) output of the SSDA. This line is inverted and fed to the ÷16 bit time counter and to the enable CRCC flip flop (Figure 9). The first TUF resets the ÷16 bit counter. The Q3 output of the bit counter, which is clocked by the inverted write oscillator (500 kHz), is used to clock the enable CRCC flip-flop. As long as TUF's are present, the ÷16 bit counter will be reset prior to its Q3 going high, preventing the enable CRC flip-flop from being clocked from its reset state (Figure 9-E, Figure 10-E). The first TUF missing at bit 7 time is clocked through the enable CRCC flip-flop, enabling the CRCC generator CRCCG (Figure 9-E, Figure 10-E) and allowing the switch clock rate flop to be clocked from its reset state by the next positive transition of Q3 (Figure 9-E, Figure 11-E). The switch clock rate flip-flop's outputs (Figure 9-H, Figure 11-H) toggle after the 16 bits of address mark clock and data information have been transmitted by the SSDA at 500 kHz. The SSDA Tx Clock (Figure 9-I, Figure 11-I) is then switched from the 500 kHz write oscillator to the 250 kHz Q0 output of the ÷16 bit counter by the switch clock flip-flop's outputs combined with the AND/OR

TABLE 2 – SSDA PREPARATION FOR READ

Step	Register Address	R/W	Data D7	D6	D5	D4	D3	D2	D1	D0	Function & Comments
1	SSDACR CR1		Rx Rs	Tx Rs	Strip Sync	Clear Sync	TIE	RIE	AC1	AC2	Inhibit: Sync, Tx, Rx
	00	W	1	1	0	1	0	0	1	0	Select CR3
2	SSDADR CR3		Intrn Sync	1 Sync Char	Clear $\overline{CTS}$	Clear TUF	X	X	X	X	1 Sync Character, Internal Sync, Clear: $\overline{TUF}$, $\overline{CTS}$
	01	W	0	1	1	1	0	0	0	0	
3	SSDACR CR1		Rx Rs	Tx Rs	Strip Sync	Clear Sync	TIE	RIE	AC1	AC2	Select Sync Code Register
	00	W	1	1	0	1	0	0	0	1	
4	SSDADR Sync Code Reg		D7	D6	D5	D4	D3	D2	D1	D0	Set Sync Code to hex F5
	01	W	1	1	1	1	0	1	0	1	
5	SSDACR CR1		Rx Rs	Tx Rs	Strip Sync	Clear Sync	TIE	RIE	AC1	AC2	Select CR2
	00	W	1	1	0	1	0	0	0	0	
6	SSDADR CR2		PC1	PC2	2 Byte RDA	WS1	WS2	WS3	Tx Syn	EIE	Inhibit SM, 2-Byte RDA, 8-Bit Word
	01	W	1	1	0	1	1	0	0	0	
7	SSDACR CR1		Rx	Tx Rs	Strip Sync	Clear Sync	TIE	RIE	AC1	AC2	Enable Read
	00	W	0	1	0	1	0	0	0	0	
8	PIADRB		CRC = 0 (In)	Int Sync (In)	Shift CRC (Out)	Index Drive (In)	X	Enable Read (Out)	Enable Write (Drive)	Reset (Out)	Enable Read Logic, Toggle Reset High
	05	W	0	0	0	0	0	1	1	1	
9	PIADRB										Toggle Reset Low
	05	W	0	0	0	0	0	1	1	0	
10	SSDACR CR1		Rx	Tx Rs	Strip Sync	Clear Sync	TIE	RIE	AC1	AC2	Enable Sync, Select CR2
	00	W	0	1	0	0	0	0	0	0	
11	SSDADR CR2		PC1	PC2	2 Byte RDA	WS1	WS2	WS3	Tx Sync	EIE	Enable SM Output
	01	W	1	0	0	1	1	0	0	0	
12	SSDASR Status Reg		RDA	TDRA	DCD	$\overline{CTS}$	TUF	Rx DVR	PE	IRQ	Test RDA for 2 Bytes Ready
	00	R	0→1	X	X	X	X	X	X	X	
13	SSDADR 01	R	0	1	1	1	1	1	1	0	Read Data, Exp: $7E_{16}$ = 2nd ½ IDAM

logic. The switch clock rate flip-flop's outputs also control the selection of 2X write data and clock or 1X write data and clock being fed to the write data formatting circuit (Figure 9-R).

The CRCCG has been clocked from the first missing TUF by the ÷16 Q0 output so that only the data portion of the transferred information is accumulated during the write data operation, including the data portion of the address mark (Figure 9-D, Figure 10-D).

Once the last data byte has been transferred from the SSDA's Tx FIFO into the Tx Shifter, the MPU enables the shift CRCC line via the PIA (Figure 13-K and 14-K). This signal is then clocked by the next positive transition of Q3 at the end of the last data byte. The shift CRCC command to the CRCCG is then delayed 1 μs (Figure 13-P, S, A; Figure 14-P, S, A; Figure 15-P, S, A) by the write oscillator clock to allow for the last data bit to be transferred into the CRCCG registers before switching into the shift mode.

Two dummy data bytes are written from the SSDA while FCS is being appended to the data field. This allows the MPU to keep track of the FCS status and disable the Shift CRCC command at the proper time, i.e., while the last dummy byte is to be shifted out of the SSDA. The disable shift CRCC command is clocked by the positive transition of Q3 at the end of the second dummy byte and is again delayed 1 μs before reaching the CRCCG. This switches the CRCCG data out back to the SSDA TxD.

When the last bit of the second dummy byte is being transmitted, a TUF occurs indicating that the Tx FIFO is empty and the content of the sync code register will be transmitted next. The sync code register was previously loaded with the 1X postamble data field and it will immediately follow the FCS field. The first TUF following the FCS is clocked by the positive transition Q3 to disable the CRCCG (Figure 13-C, E, F; Figure 16-C, E, F) starting the write termination sequence. The next positive transition of Q3 restores the clock to the 2X mode which allows the second TUF to reset the ÷16 bit counter (Figure 13-F, H, N; Figure 16-F, H, N). While the postamble is being written, the sync code register is loaded with the 2X gap clock and data pattern which will be written until the FIFO is loaded with the next address mark restarting the operation or the transmitter section is reset by software. Using this technique, the entire track may be formatted without write current to the drive being shut off and without any glitches at the switchover points.

TUF's may be counted to determine gap sizes when write formatting the disk.

The write data format logic takes the NRZ clock and data information (Figure 9-R and Figure 12-R) and generates the raw unseparated clock and data format required by the drive electronics (Figure 9-M and Figure 12-M). The NRZ data is clocked and delayed for one-half a write oscillator period (Figure 9-L and Figure 12-L) to remove any delays generated in the previous logic. The inverted and delayed NRZ data is NANDed with the 500 kHz write oscillator (Figure 9-A and Figure 12-A) to generate the -Write Data for the drive electronics.

ACKNOWLEDGEMENT

Special appreciation is given to Mark Eidson who is responsible for the original text of this article. Also, credit and thanks for initial circuit design are given to Mark Eidson and Tom Daly.

FIGURE 9—Write Synchronization Simplified Circuit Diagram

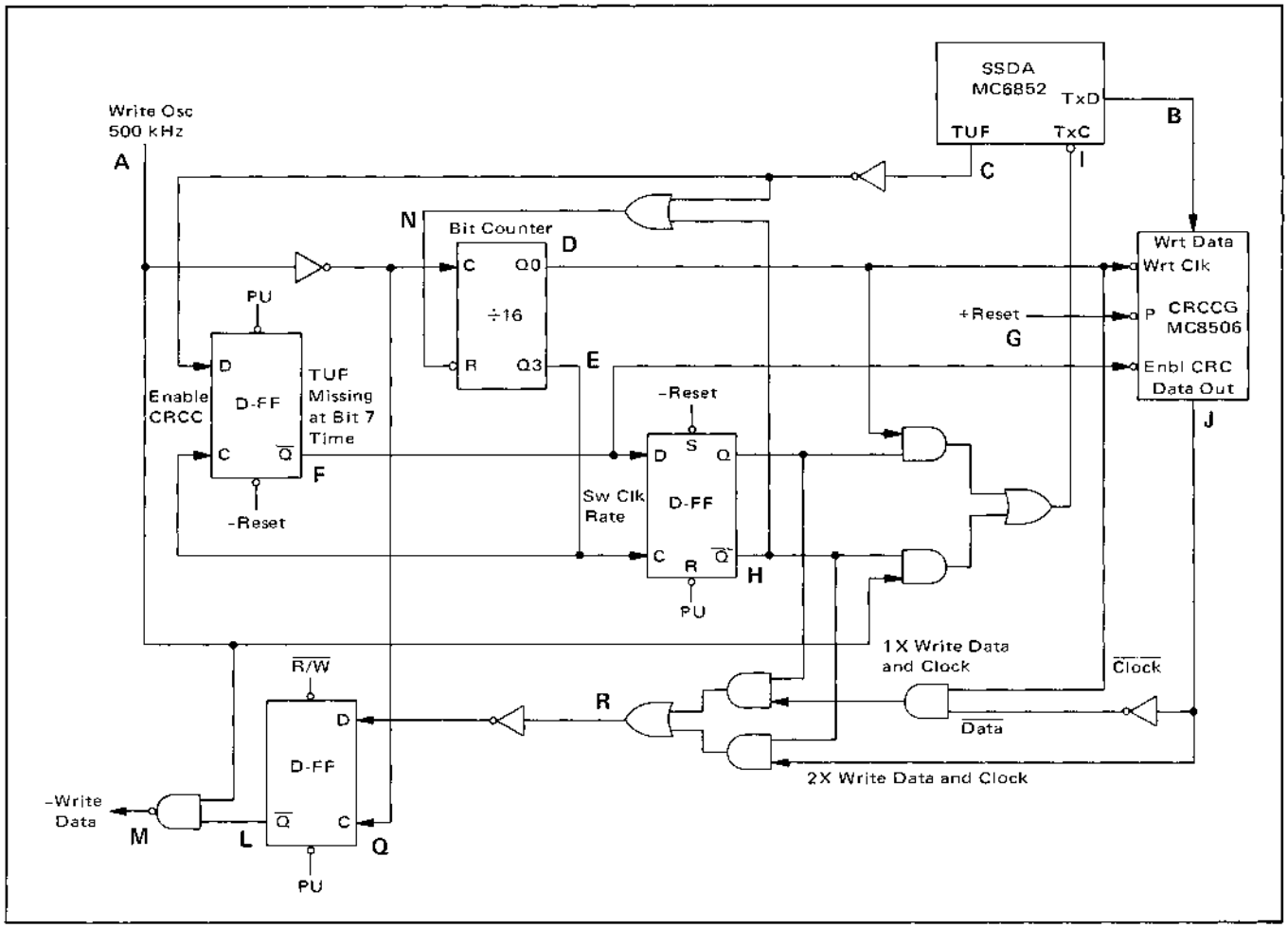


FIGURE 10—Write Synchronization Timing Diagram

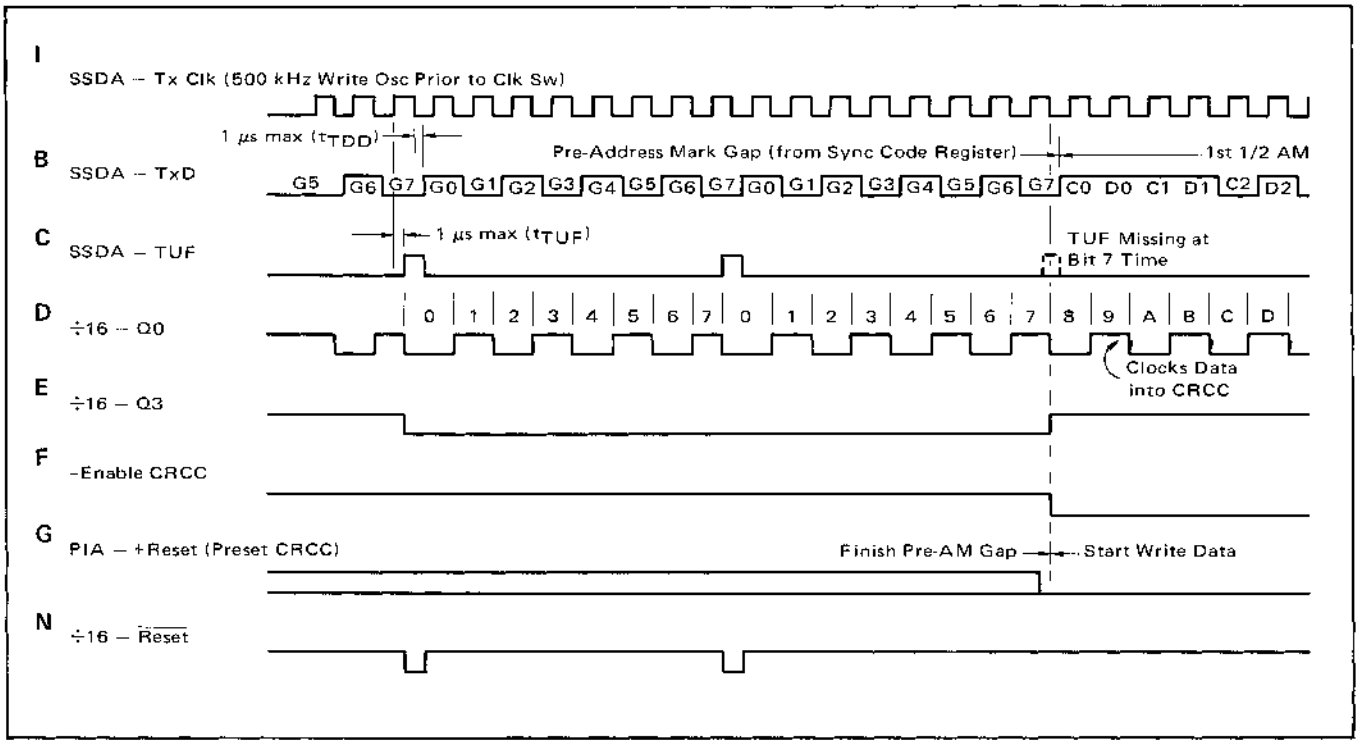


FIGURE 11--Write Switch Clock Rate Timing Diagram

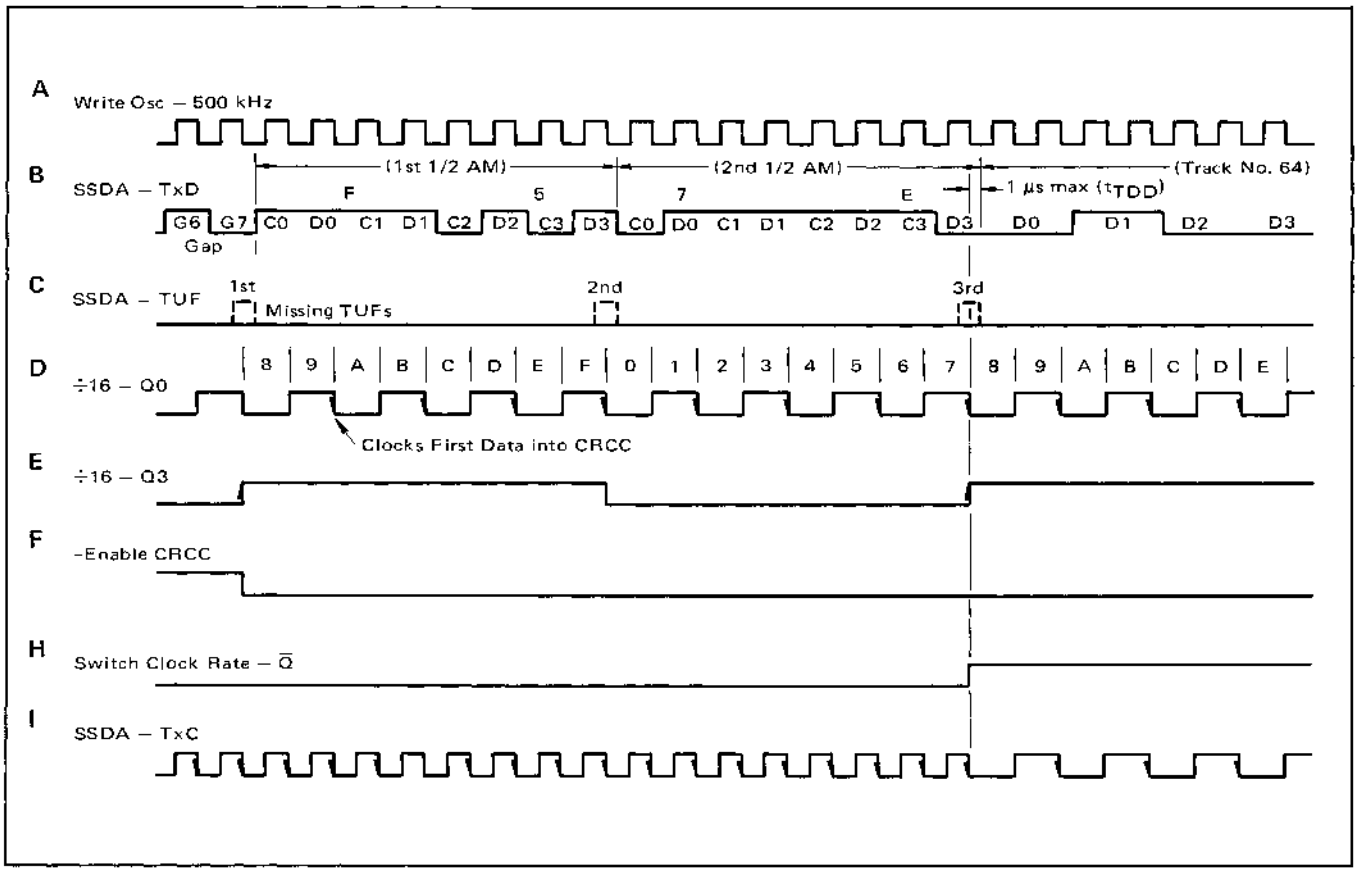


FIGURE 12–Write Data Format Timing

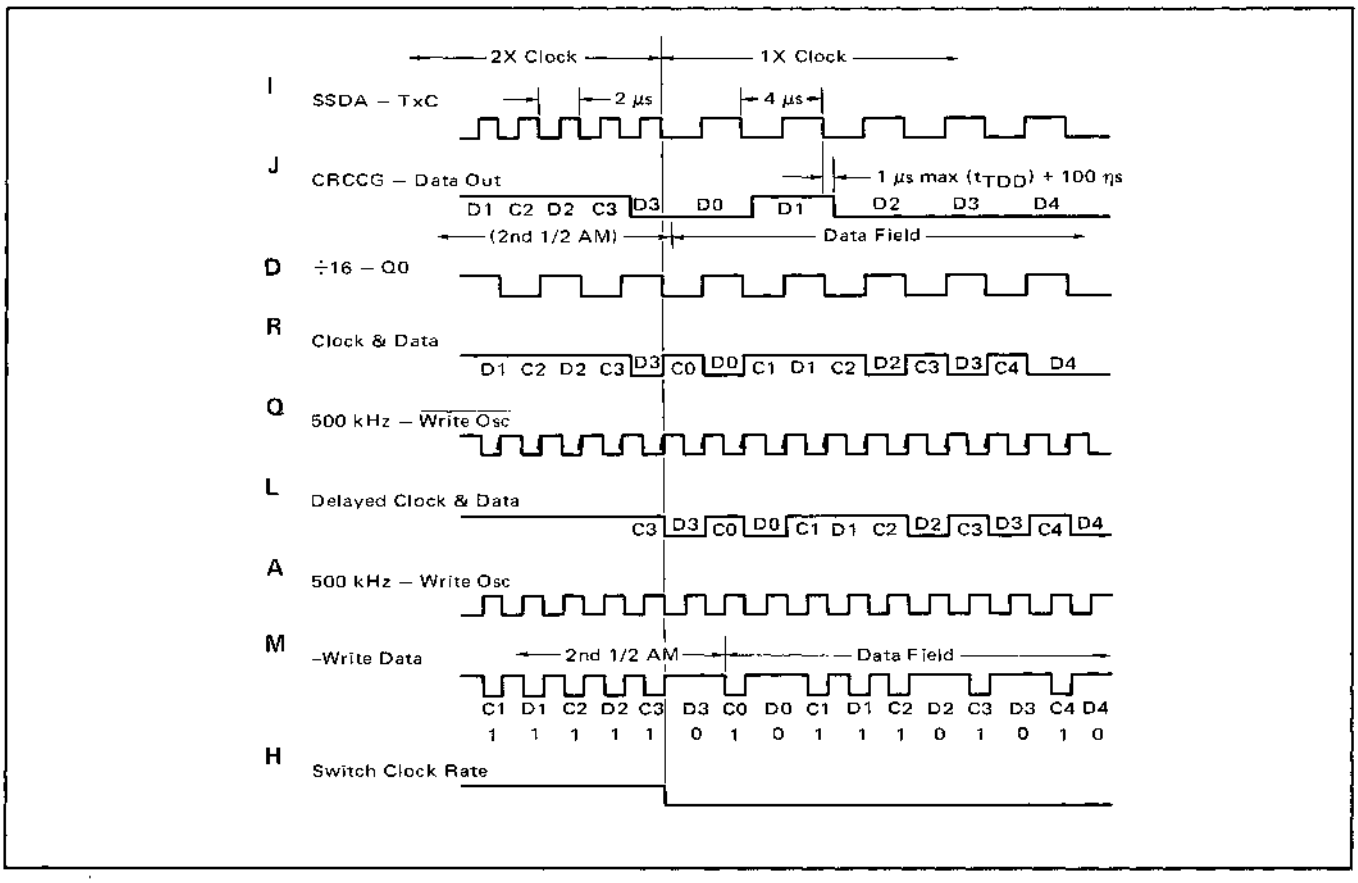


FIGURE 13—Write CRCC Simplified Block Diagram

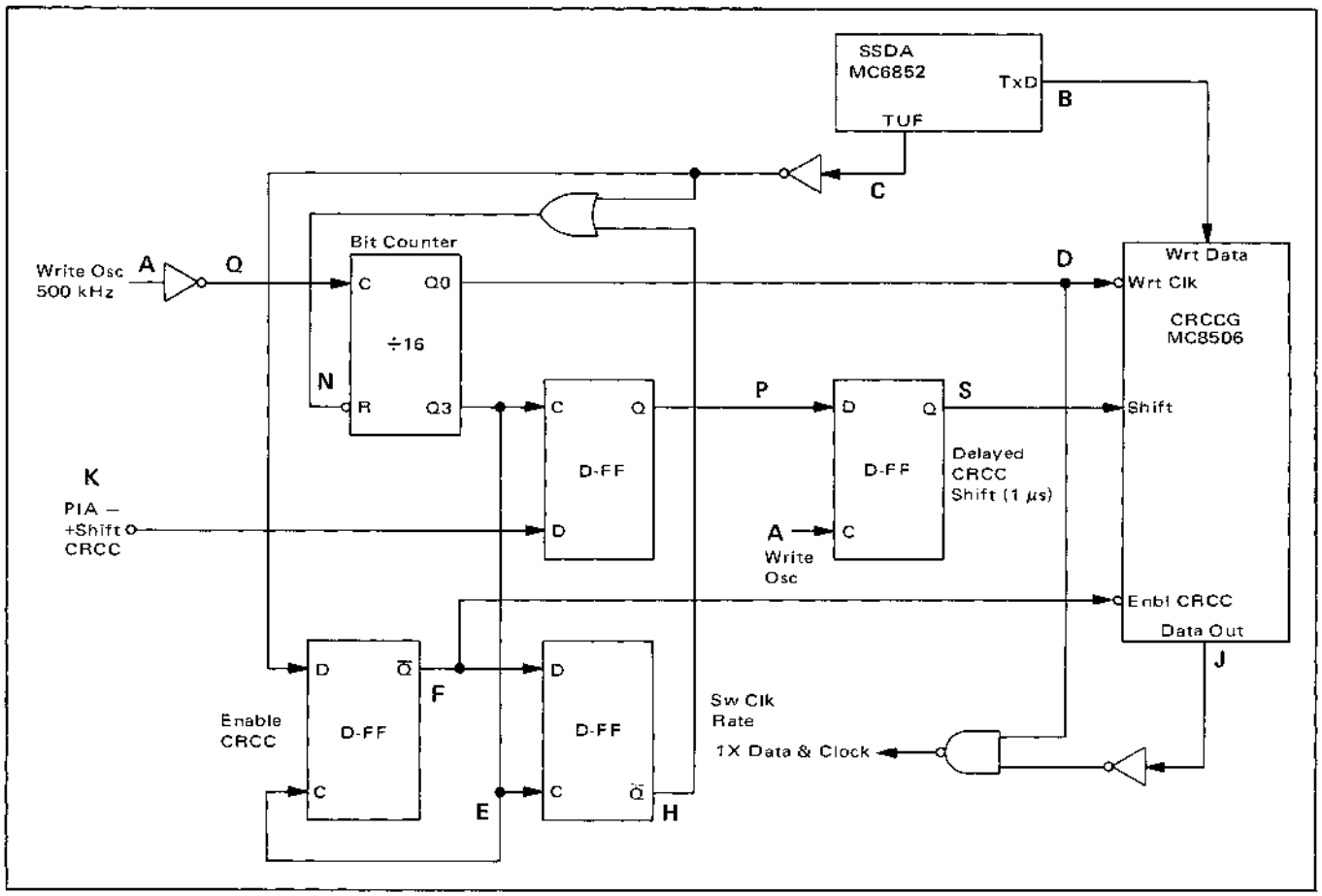


FIGURE 14—Write CRCC Timing Diagram

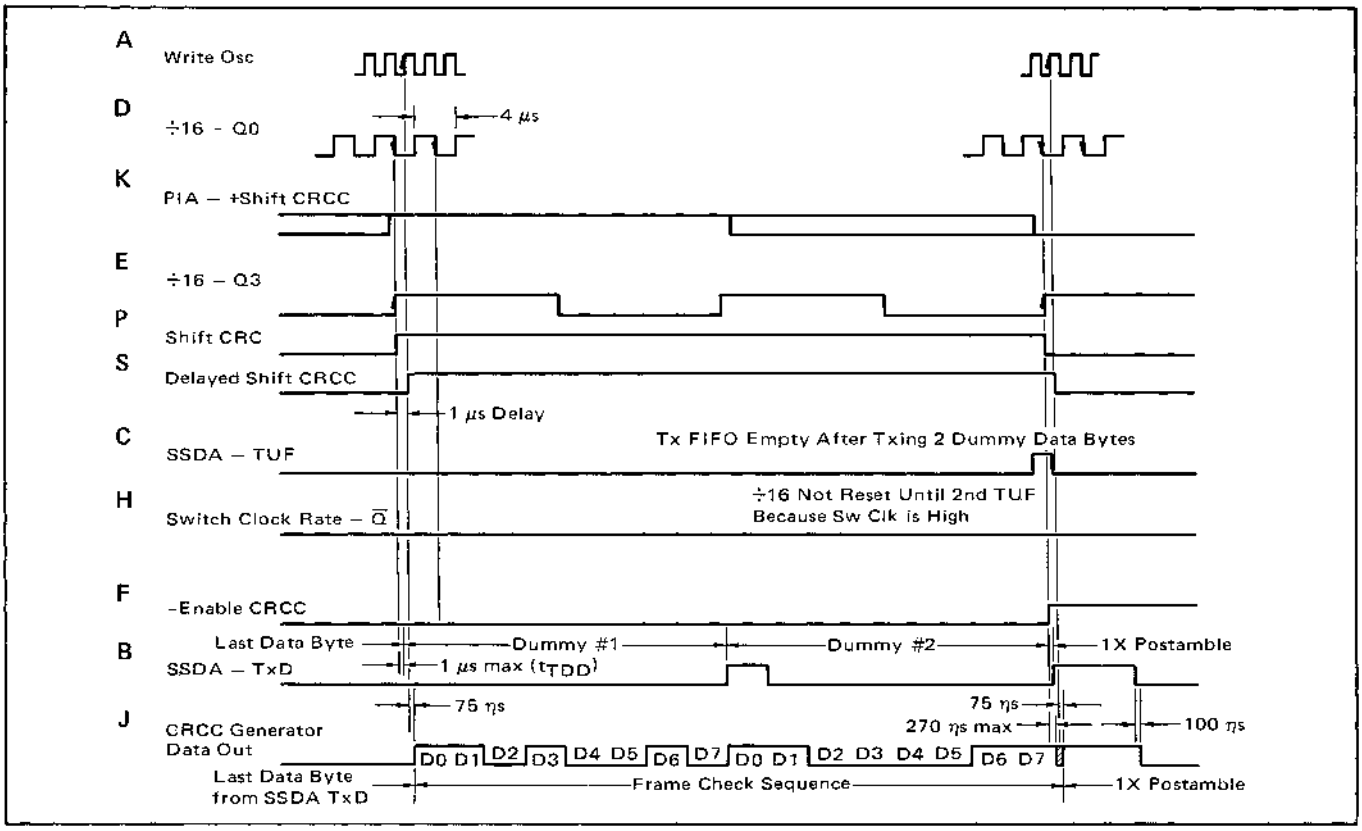


FIGURE 15–Write CRCC Detailed Timing

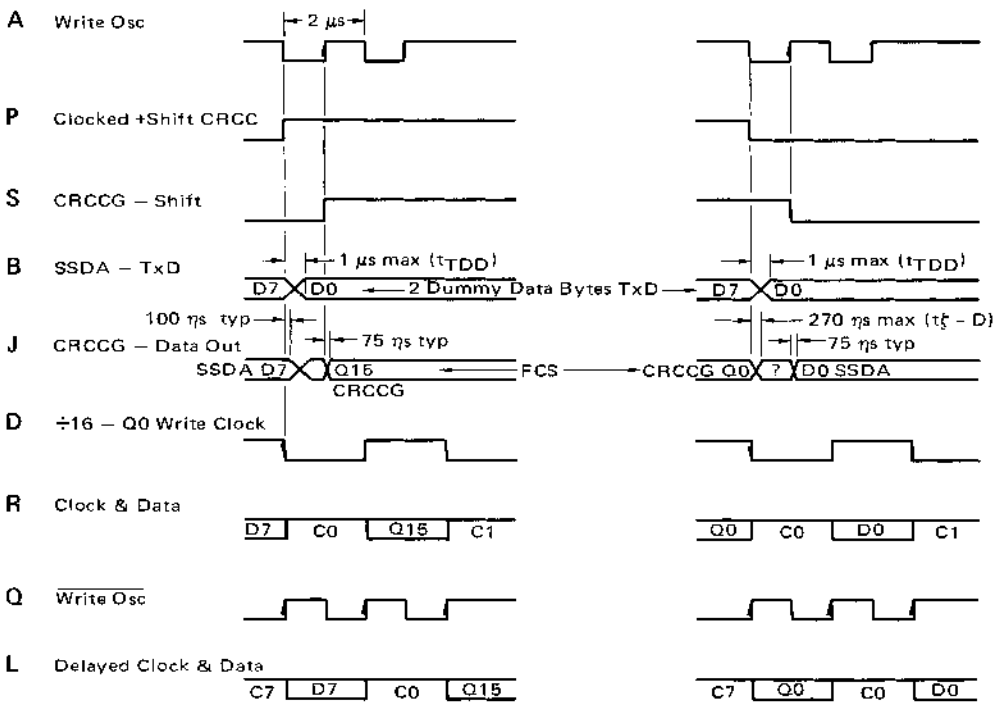
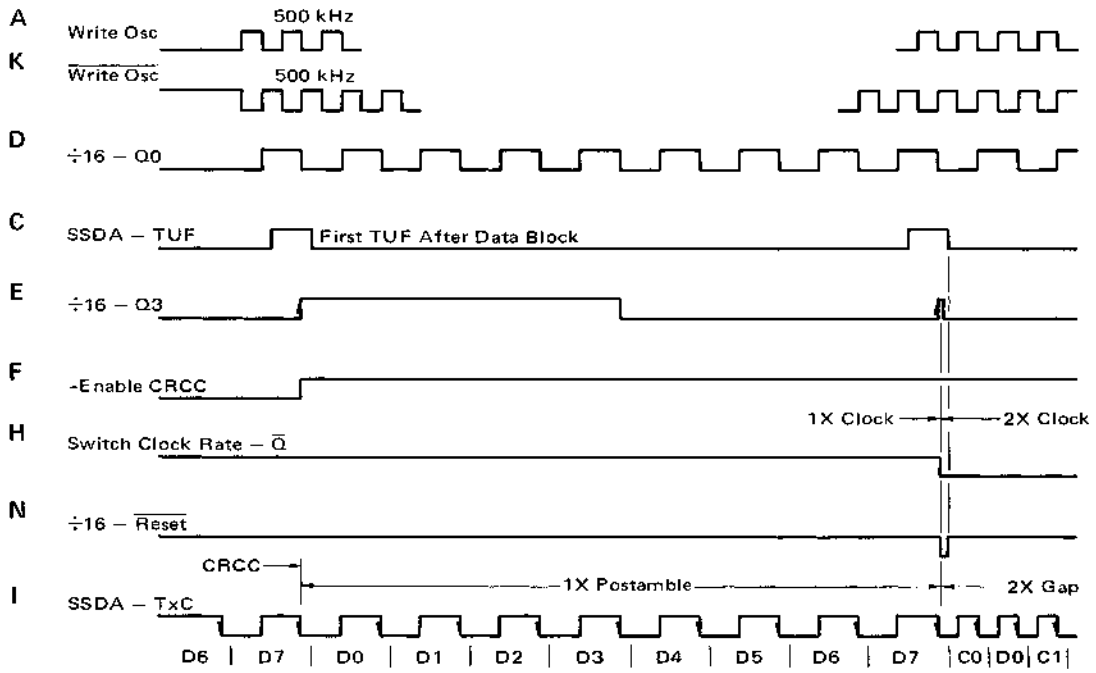


FIGURE 16–Write Termination Timing Diagram



MOTOROLA Semiconductor Products Inc.

BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

10692 PRINTED IN USA 10-76 IMPERIAL LITHO 86C066 12M AN-764