# Preliminary
# Small File Server

# Theory of Operation

Michael Dhuey

8 – 11 – 84

# Table of Contents

## Introduction

The Small File Server (SFS) is designed to be a shared 20 M byte disk for Macintosh. It contains a 20 Mbyte 'Nisha' Winchester technology hard disk, 128 K bytes of RAM, 64 K bytes of ROM, a 8 MHz 68000 CPU, two serial ports, and a SCSI compatible port. It can be directly connected to a Macintosh through the serial port clocked at 650 K baud or can be shared on an AppleBus at 230.4 K baud. The Small Computer Standard System Interface (SCSI) is compatible with industry standard data storage devices such as Winchester disks, tape drives, and printers. It provides both expansion and media compatibility to devices not directly supported by Apple.

The SFS design can be divided into five logical components: the memory state machine, the disk state machine, the serial interface, the disk servo/clock interface, and the SCSI interface.

The memory state machine interfaces the 68000 CPU to the RAM, ROM, VIA, SCC, DIsk command latch, and SCSI interface. It generates a series of system clocks from the 15.6672 MHz crystal by simple division and a little cycle stretching. These clocks are combined with signals from the 68000 CPU to generate the RAM, ROM and SCC access timing. The memory state machine gives priority access to RAM to the disk state machine at the expense of the 68000 CPU. This is transparent to the 68000 as it is merely held off while the Direct Memory Access (DMA) is being made.

The disk state machine interfaces the 16 bit words of the RAM to the bit stream on the disk. It contains a 16 bit shift register, a RAM address counter, a 4 bit counter, a state sequencer, an Error Correction Chip (ECC), and an address decoder. All transfers of data to or from the disk are through DMA accesses to the top 1 K bytes of RAM. This area is accessed by the disk state machine to bring sectors of 532 bytes of data to or from the disk. The sectors are transferred with one of four state sequences: format, read—ID, read, and write. Format is only used to write the initial headers and data on the disk when it is new. Read—ID reads the next sector that comes under the head into memory. This avoids header compares and allows the fastest possible access. Read reads the sector if the header in RAM matches the one on the disk. Write writes the sector from RAM if the header in RAM matches the one on the disk.

The serial interface is the Serial Communications Controller (SCC) chip and two driver chips and one receiver chip. The SCC is the 4 MHz Z8530 chip from Zilog and the drivers are 26LS30 chips and the receiver is a 26LS32 chip. The SCC is capable of most serial communications protocols such as asynchronous, synchronous, BISYNC, HDLC, and SDLC. It can also FM encode the data stream for self—clocking modes used at high data rates. The 26LS30 and 26LS32 drivers and receivers allow RS— 232C as well as RS—422 modes of connection. One 26LS30 is used to drive the transmitted data while the second can drive a clock generated by the SCC to allow synchronous data transmission useful in direct connection to Macintosh.

The disk servo/clock interface is used to control the movement of the read/write head of the disk and to access the real time clock chip. The interface consists of a Versatile Interface Adapter (VIA), a real time clock chip, a PAL (SERIAL), and a 74LS32. The 74LS32 is purely to hold the address line to the register selects high until an actual access from the 68000. If the address lines are allowed to float the VIA is unreliable. The VIA contains a system timer to generate a 16 ms interrupt similar to the Macintosh. Also, the 1 second interrupt of the real time clock is available. The Serial PAL is used to connect the 8 bit synchronous shift register to the disk servo and the real time clock. Commands to move the disk head are sent to the servo as a sequence of asynchronous characters. The Serial PAL generates the start and stop bits to make the 8 bit shift register into the 10 bit asynchronous character. It also reverses the line to receive status from the servo.

Finally, the SCSI interface is the expansion port to allow high capacity and high performance peripherals to be added to the SFS. It consists of a PAL (SCSI), 3 74LS373s, and 3 DS3662s. The PAL decodes the SCSI address form the memory map and latch the SCSI handshake and control bits. The LS373s latch the 8 bit data bus and status lines. The DS3662 are special bus transceivers which have controlled edges to reduce crosstalk on the SCSI bus as well as reduce EMI.

Additional documents related to this design are:

MC68000 16-BIT MICROPROCESSOR, Motorola Semiconductor, 1983;

ZILOG 1983/84 COMPONENTS DATA BOOK, Zilog, 1983;

SYNERTEK DATA BOOK 1983, Synertek, 1983;

PAL PROGRMMABLE ARRAY LOGIC HANDBOOK, Monilithic Memories Inc., 1983;

PRELIMINARY SMALL FILE SERVER ERS, Michael Dhuey, 7-19-84;

SPECIFICATION FOR WIDGET DRIVE, Apple Part # 068-6001-A, 1-5-84;

SCSI SMALL COMPUTER SYSTEM INTERFACE, ANSI X3T9.2/82-2 - Rev. 14, 5 - 2-84.

## 1.0 Memory State Machine

The memory state machine consists of the PALs Clocks, Ramsm, and Decode. They generate the control signals for the 68000 CPU, the RAM chips, the RAM address multiplexers, the RAM data bus buffers, the ROMs, the VIA, the SCC, and the disk command latch.

The SFS memory map is a 16 Mbyte address space divided into sections for the VIA, SCC, disk command latch, RAM, and ROM. The Decode PAL performs the initial division of memory with the A23–A19 address bits of the 68000 CPU. Based on the decoding of the high order address bits the appropriate device and the timings for that device are selected and executed by the Ramsm PAL. The Clocks PAL generates the time base which the Ramsm PAL uses to generate the appropriate control signals.

The SFS data bus is separated into two 16 bit components as shown in Figure 1.1. The main component connects the 68000 CPU to the RAM buffers, the ROMs, the SCC, the VIA, and the SCSI interface. The RAM chips are on a separate bus with the disk state machine and the RAM buffers. The RAM buffers connect the two buses when the 68000 accesses RAM. This allows the higher priority memory accesses by the disk state machine to occur directly to the RAM while the RAM buffers prevent the 68000 CPU from interfering. In fact the 68000 is free to access anything except the RAM while the DMA from the disk state machine is occurring.

## Important:

The 68000 **Test and Set (TAS)** instruction ***must not be used*** The instruction takes too long to execute to guarantee the DMA memory cycles will occur properly. This is a requirement of Macintosh and Lisa software as well so this is not a serious limitation.

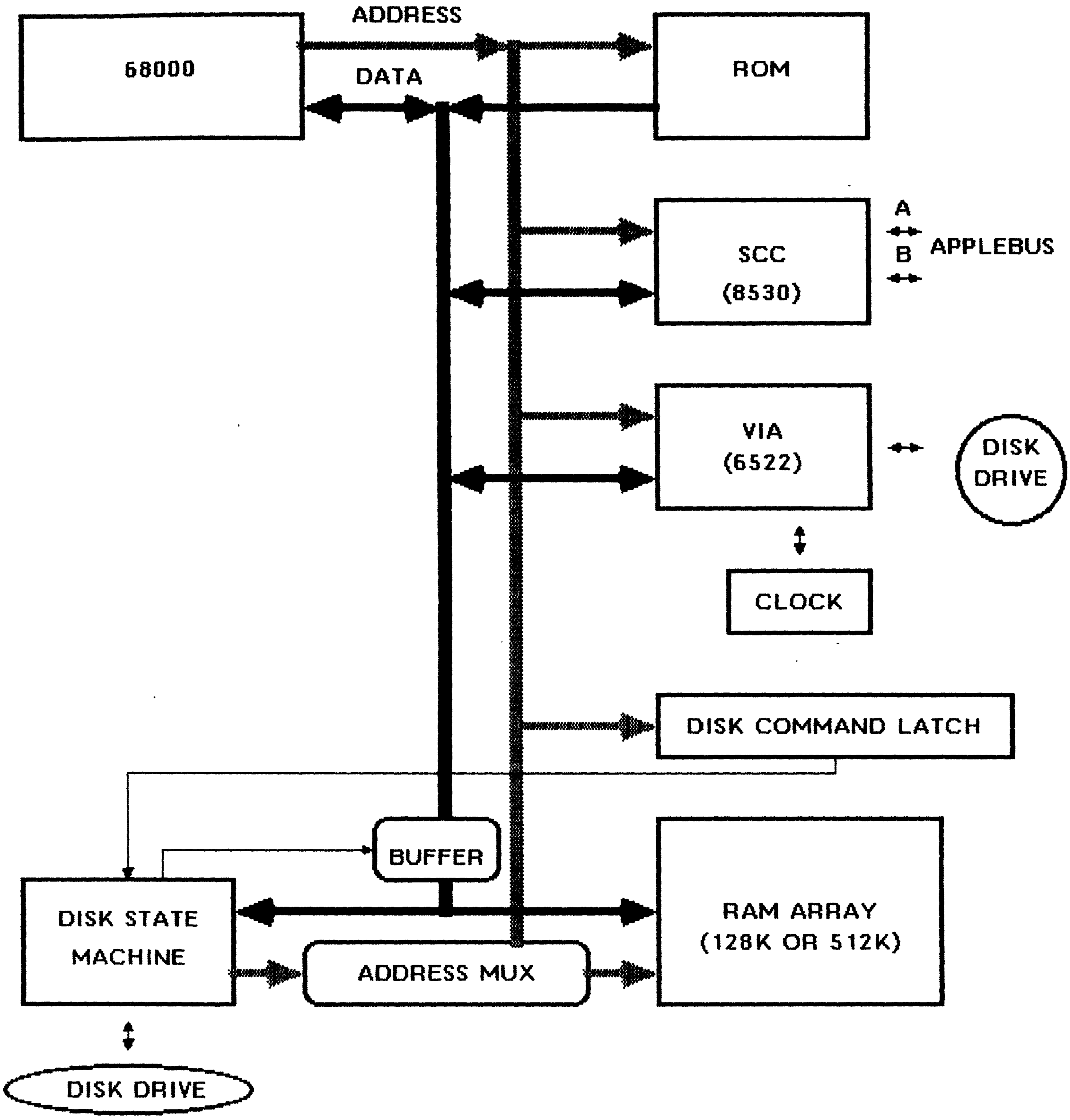

Figure 1.1  Data Bus Block Diagram

## 1.1 Clocks PAL

The Clocks PAL generates all the timing signals used in the memory state machine. It uses the 15.6672 MHz oscillator module to clock the PAL and divides by 2, 4, 8, and 16 to generate the frequencies shown in Table 1.1. It also generates a 3.6864 MHz signal by adding 63.8 ns to the period of /C3M every 16 counts of C16M. Figure 1.2 shows the phase relationship of the clocks which is used by the Ramsm PAL to synchronize the memory state machine with the 68000 CPU timings. This PAL also syncs the DMA request from the disk state machine to the 15.6672 MHz clock of the memory state machine.

**Table 1.1 Clock Frequencies**

| Clock name | Frequency | Period |
|---|---|---|
| C16M | 15.6672 MHz | 63.8276 ns |
| C8M | 7.8336 MHz | 127.6552 ns |
| /C7M | 7.3728 MHz (avg.) | |
| /C4M | 3.9168 MHz | 255.31 ns |
| /C3M | 3.6864 MHz (avg.) | |
| /C2M | 1.9584 MHz | 510.62 ns |
| /C1M | 0.9792 MHz | 1021.24 ns |

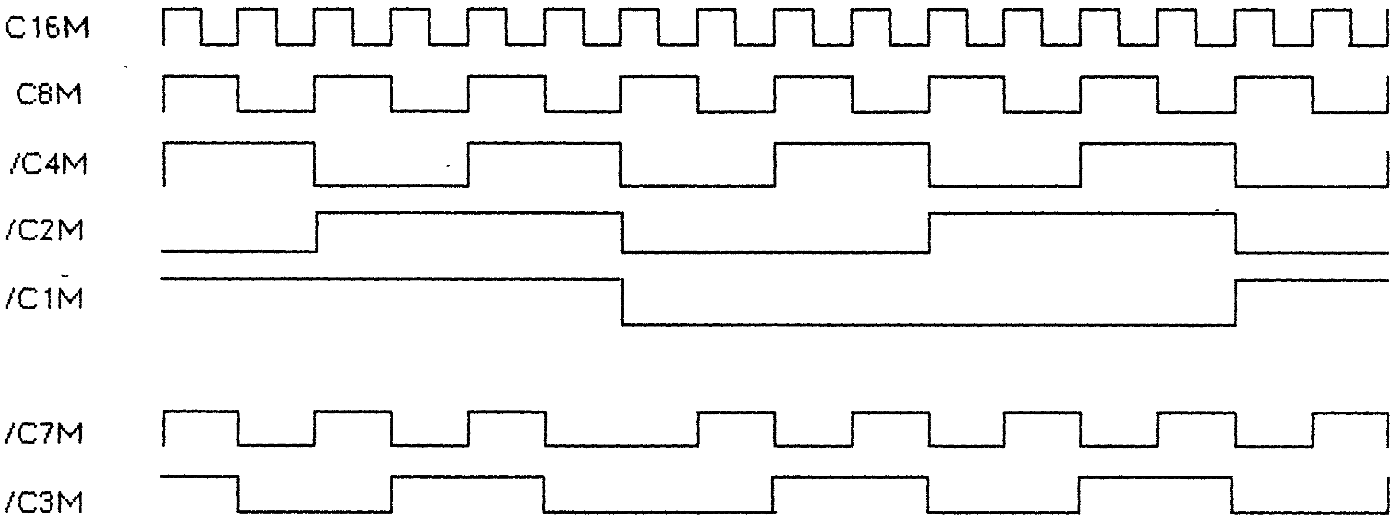

**Figure 1.2 Phase relationship of the clocks**

## 1.2 Decode PAL

The Decode PAL determines what type of memory cycle the 68000 CPU will execute based on the address the 68000 CPU is presenting and whether a DMA cycle for the disk state machine is occurring. Address lines A23–A19 from the 68000 CPU are decoded into the various address spaces indicated in Table 1.2.

**Table 1.2 Address Map**

| Power–Up Map (Overlay = 1) | Address | Normal Map (Overlay = 0) |
|---|---|---|
| | $FF FFFF | |
| VIA (6522) | $EF FFFF | VIA (6522) |
| VIA (6522) | $E8 0000 | VIA (6522) |
| DISK COMMAND LATCH | $DF FFFF | DISK COMMAND LATCH |
| DISK COMMAND LATCH | $D8 0000 | DISK COMMAND LATCH |
| SCSI INTERFACE | $D7 FFFF | SCSI INTERFACE |
| SCSI INTERFACE | $D0 0000 | SCSI INTERFACE |
| SCC WRITE | $BF FFFF | SCC WRITE |
| SCC WRITE | $B0 0000 | SCC WRITE |
| SCC READ | $9F FFFF | SCC READ |
| SCC READ | $90 0000 | SCC READ |
| RAM (512 K BYTES) | $67 FFFF | |
| RAM (128 K BYTES) | $61 FFFF | |
| RAM | $60 0000 | |
| ROM (64 K BYTES) | $40 FFFF | ROM (64 K BYTES) |
| ROM | $40 0000 | ROM |
| | $07 FFFF | RAM (512 K BYTES) |
| | $01 FFFF | RAM (128 K BYTES) |
| DUPLICATE ROM IMAGE | $00 FFFF | RAM |
| DUPLICATE ROM IMAGE | $00 0000 | RAM |

Note that the Overlay bit causes the RAM and ROM address spaces to be mapped differently. This is to put the ROM in the lowest address space to allow the 68000 CPU power–up reset vector to be present at power on. The overlay bit is supplied by the VIA and is high on power–up. The power–up reset vector points to an address in the $40 xxxx address space where the ROM code begins execution. Once the code and vector table have been built in RAM the Overlay bit is set low to form the normal memory map.

The address space is also decoded in five types of memory cycle: RAM, ROM, SCC, VIA, and interrupt. Table 1.3 indicates how the /RAM, /ROM, /VPA, /IPL0, and /IPL1 signals from the Decode PAL indicate which type of cycle is being executed.

**Table 1.3 68000 Memory Cycle Decode**

| Cycle Type | /RAM | /ROM | /VPA | /IPL0 | /IPL1 |
|---|---|---|---|---|---|
| RAM | 0 | 1 | 1 | 1 | 1 |
| ROM | 1 | 0 | 1 | 1 | 1 |
| SCC | 0 | 0 | 1 | 1 | 1 |
| VIA | 1 | 1 | 0 | 1 | 1 |
| Interrupt | 1 | 1 | 0 | 0 | 1 |
| Interrupt | 1 | 1 | 0 | 1 | 0 |

## 1.3 Ramsm PAL

The Ramsm PAL generates the timings for each of the different 68000 memory cycles and interfaces the disk state machine to the RAM. Each cycle type has unique characteristics.

The ROM cycle (Figure 1.3) is not affected by the disk state machine. It always completes in 4 CPU clock cycles (510 ns). This cycle is also used in reference to the disk command latch and the SCSI interface.

The RAM cycles (Figures 1.4, 1.5) are subject to the availability of RAM since the disk state machine has priority access to RAM. If the disk is presently using the RAM then the RAM cycle is stretched by withholding /DTACK to the 68000 CPU and disabling the RAM data buffers. However, if the 68000 CPU has begun a RAM cycle the disk state machine is held off until the 68000 CPU completes the cycle. This is compensated for by the DMA request being made before the RAM is actually needed by the disk state machine to allow the 68000 CPU to complete its cycle.

The SCC cycles (Figures 1.6, 1.7) are a special stretching of the ROM cycle to 765 ns to meet the setup and hold requirements of the SCC chip. This is accomplished by using the /RCMUX signal to indicate the beginning of the cycle as a high and the rest of the cycle as a low. When the disk state machine is accessing memory is uses /RCMUX so SCC cycles are delayed when this occurs.

The VIA cycles (Figures 1.8, 1.9) are generated by asserting the /VPA signal in response to the VIA address space. The /VPA signal causes the 68000 CPU to execute a 6800 equivalent memory cycle with the assertion of /VMA. This cycle can be fairly long (2 us) depending on the relationship between the E clock generated by the 68000 and the assertion of /VPA. Further discussion of the 6800 cycle can be found in the 68000 data sheet from Motorola.

The interrupt cycle is similar to the VIA cycle. It is generated by any of the 3 interrupt lines to the 68000 being asserted. This causes the 68000 to finish execution of the present instruction and to fetch the interrupt vector. The address lines indicate this by all going to one. This causes the /VPA signal to be asserted by the Decode PAL. The 68000 responds by using the vector table in low memory to fetch the address of the interrupt handler for that interrupt.

Each of the cycle timing make assumptions about the 68000 timings. The key is the relationship of the C8M and /C4M clocks. The C8M clock is the clock sent to the 68000 for its CLK. The Ramsm PAL will only start a cycle when the 68000 Address Strobe (/AS) falls and C8M is high and /C4M is low. This allows the Ramsm PAL to make assumptions about the timing of the rest of the cycle. Once the first 68000 cycle is executed with this starting condition all subsequent ones will start with the same conditions automatically based on the internal design of the 68000. Thus no clock cycles are wasted resyncing the 68000 to the system clocks.

Figures 1.3 through 1.9 make certain assumptions. The S0—S7 symbols in the C8M clock indicate the actual internal state of the 68000. States 4 and 5 are repeated by the 68000 until the Data Acknowledge (/DTACK) is asserted by Ramsm or in the case of the 6800 cycles the E clock is in the right state. The gray areas indicate the minimum and maximum time a signal can take to transition. The timing diagrams are only accurate to the half cycle of the C16M clock. Any serious analysis must use the PAL equations and the 68000 data sheet.

The final assumption made by the Figures is that a Direct Memory Access (DMA) is not being made by the disk state machine. If a DMA was occurring the RAM cycles would be stretched by the appropriate number of S4 — S5 states to wait for the DMA to finish and C8M to be high and /C4M to be low. Likewise the SCC cycles would be stretched to wait for /RCMUX to be free.

## Table 1.4 Symbols used in Figures 1.3 — 1.9

| Symbol | Definition |
|---|---|
| A23—A1 | 68000 address lines (/UDS and /LDS form A0) |
| /UDS | 68000 Upper Data Strobe |
| /LDS | 68000 Lower Data Strobe |
| /xDS | 68000 Upper and/or Lower Data Strobes |
| /AS | 68000 Address Strobe |
| R/W | 68000 read/write line |
| /DTACK | Ramsm to 68000 Data Acknowledge |
| ROM | Output enable to the ROMs |
| D15—D0 | 68000 data lines |
| /RAS | Row Address Select for the RAMs |
| /CAS | Column Address Select for the RAMs |
| /RCMUX | Row / column address select for the 68000 to RAM address multiplexers |
| /SCCEN | Chip enable for the SCC |
| /SCCRD | Read select for the SCC |
| /VPA | Valid Peripheral Address from Ramsm to 68000 |
| /VMA | Valid Memory Address from 68000 to VIA |
| E | E clock from 68000 |

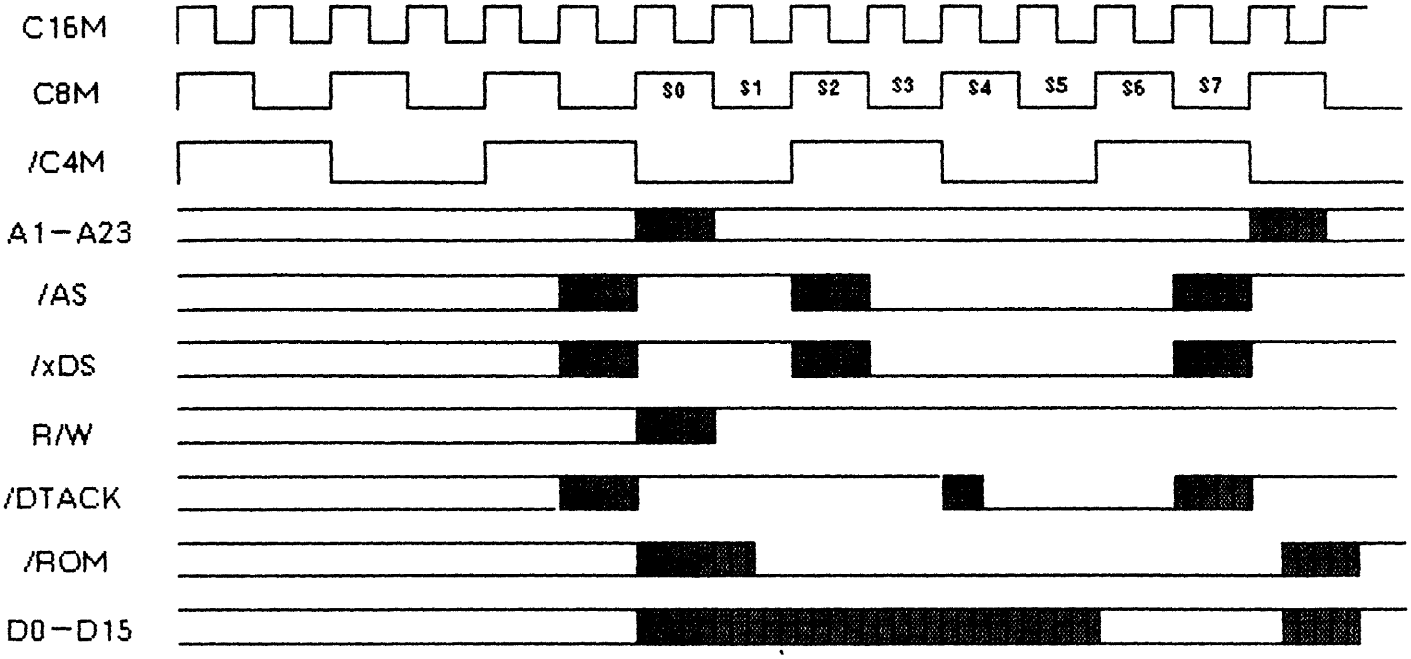

**Figure 1.3 68000 ROM cycle**

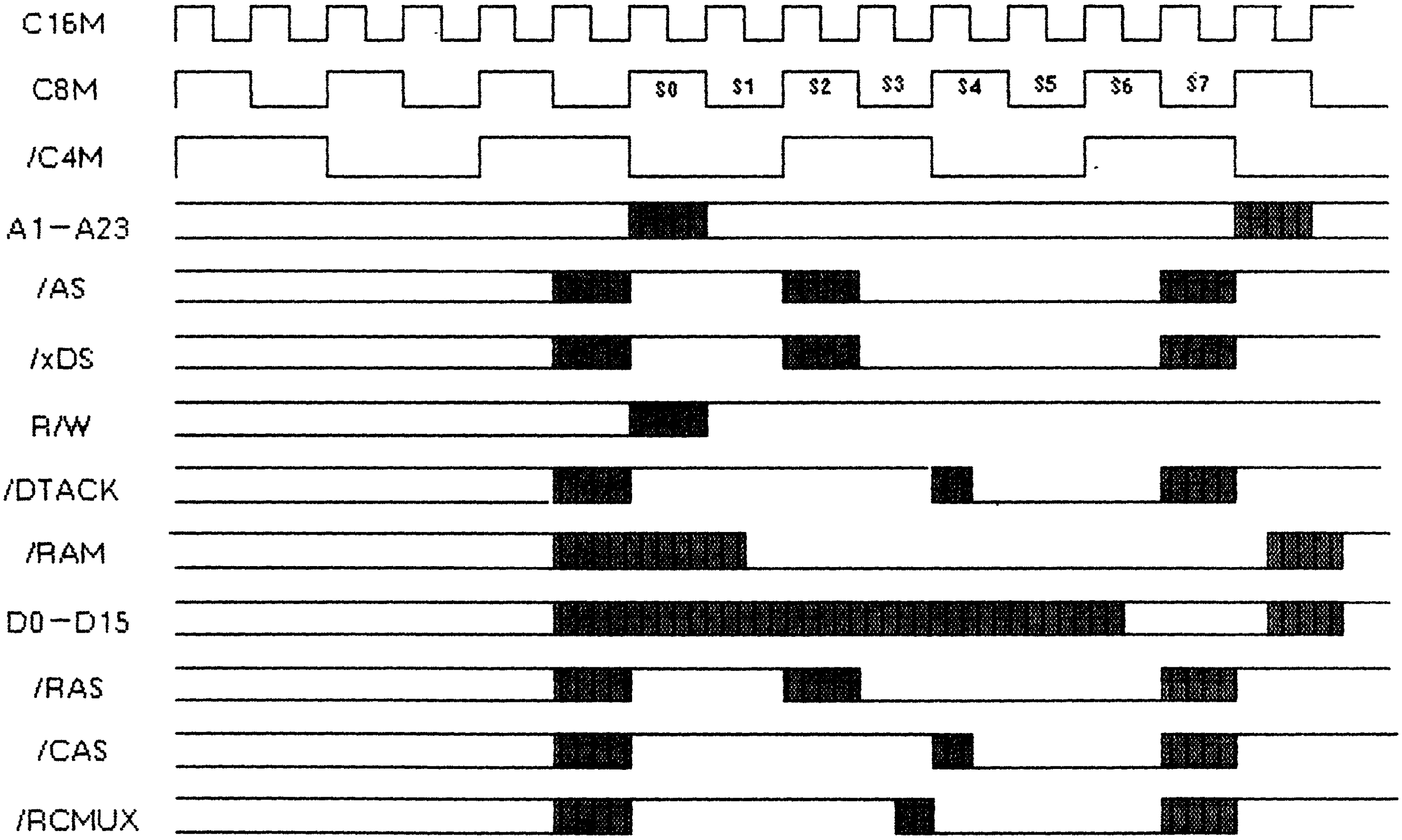

**Figure 1.4 68000 RAM read cycle**

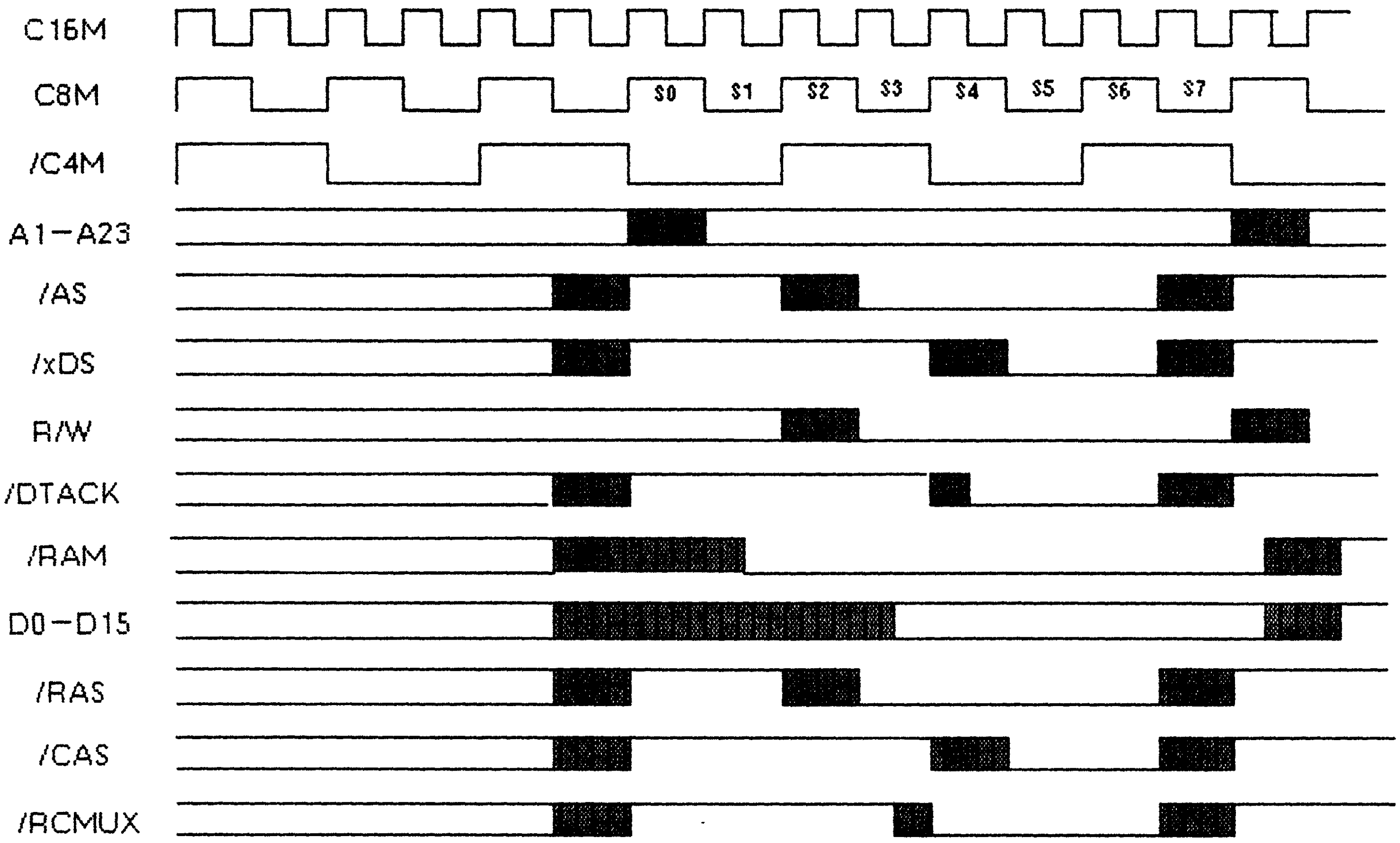

Figure 1.5  68000 RAM write cycle

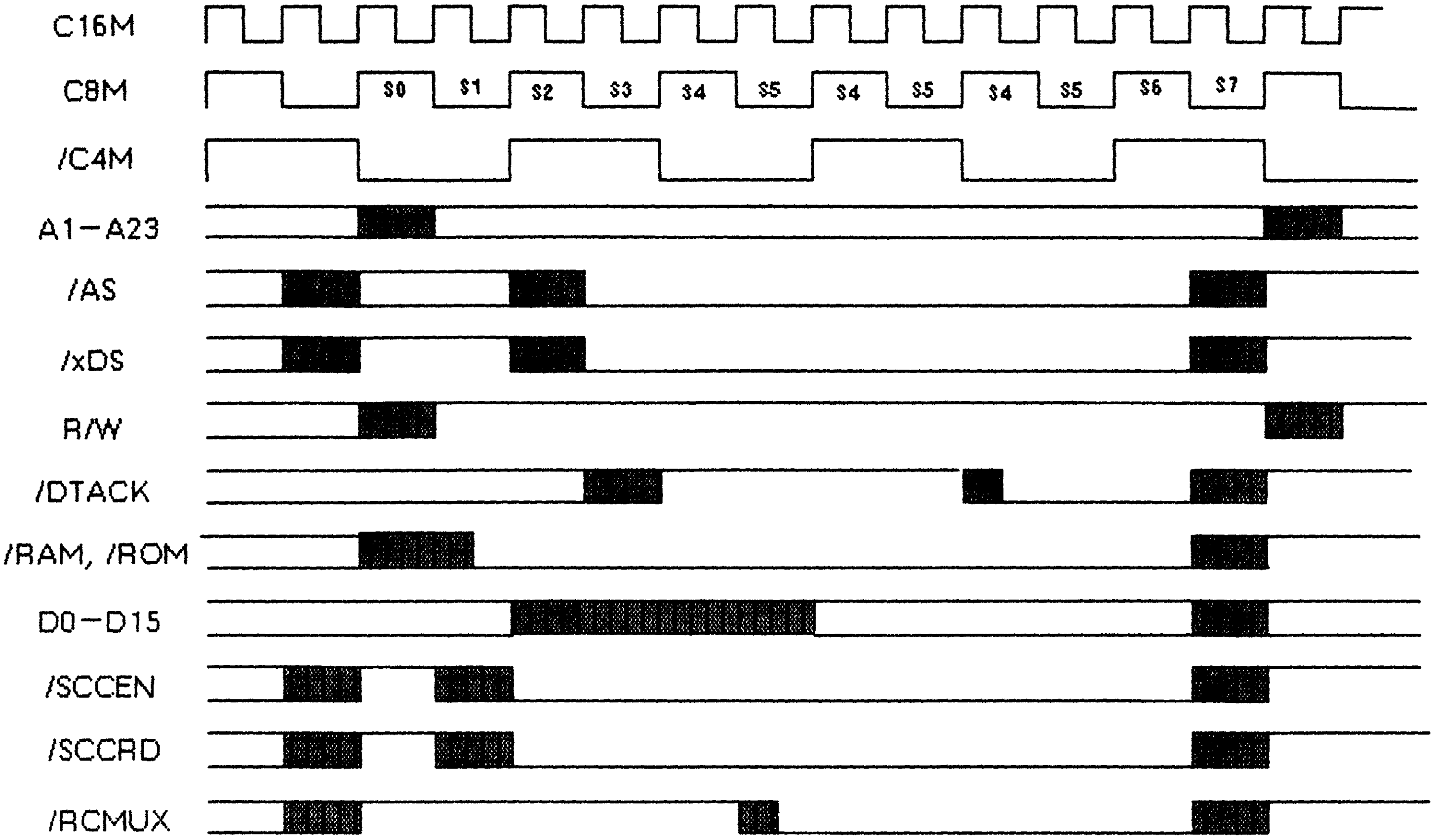

Figure 1.6  68000 SCC read cycle

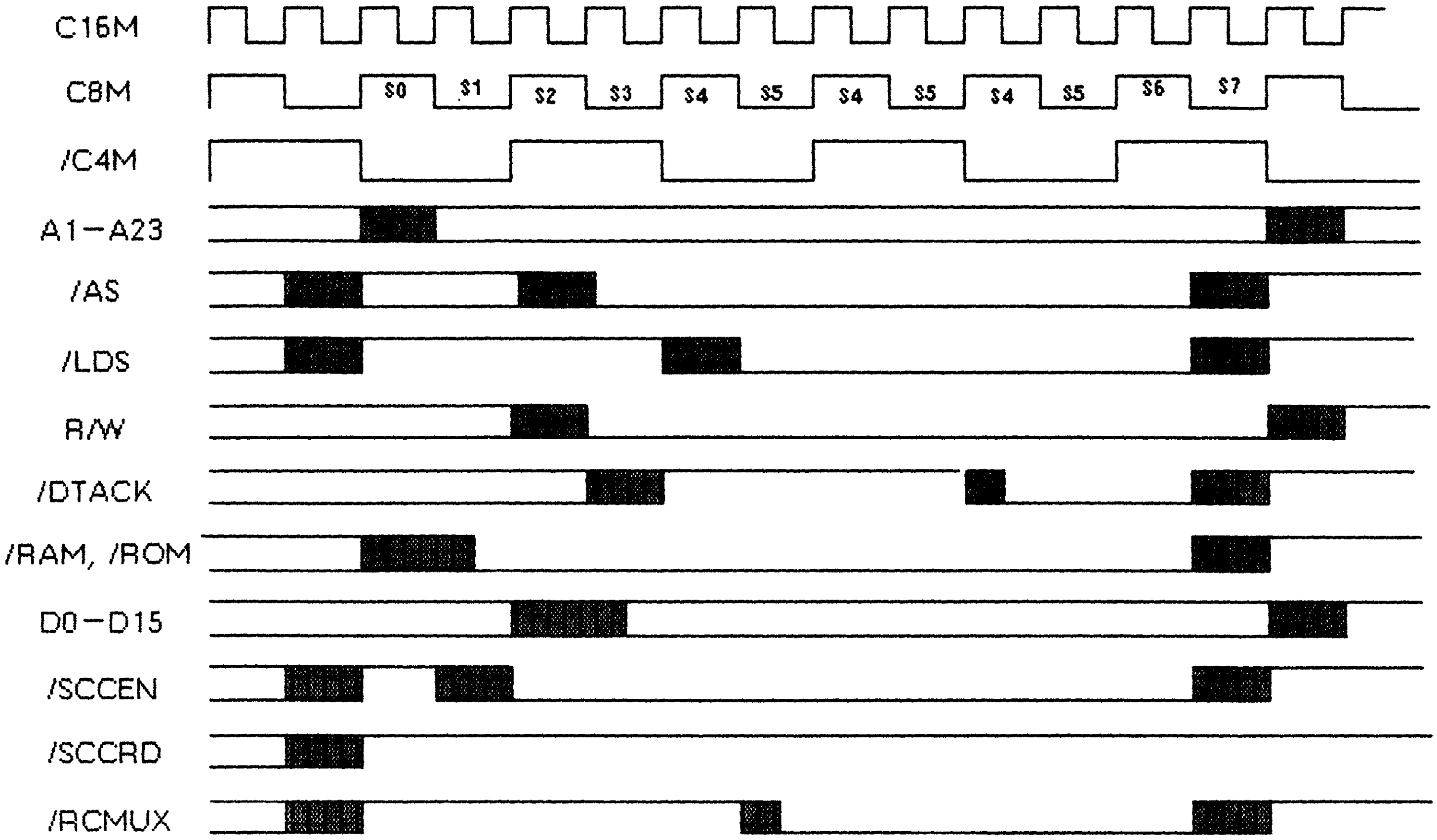

**Figure 1.7 68000 SCC write cycle**

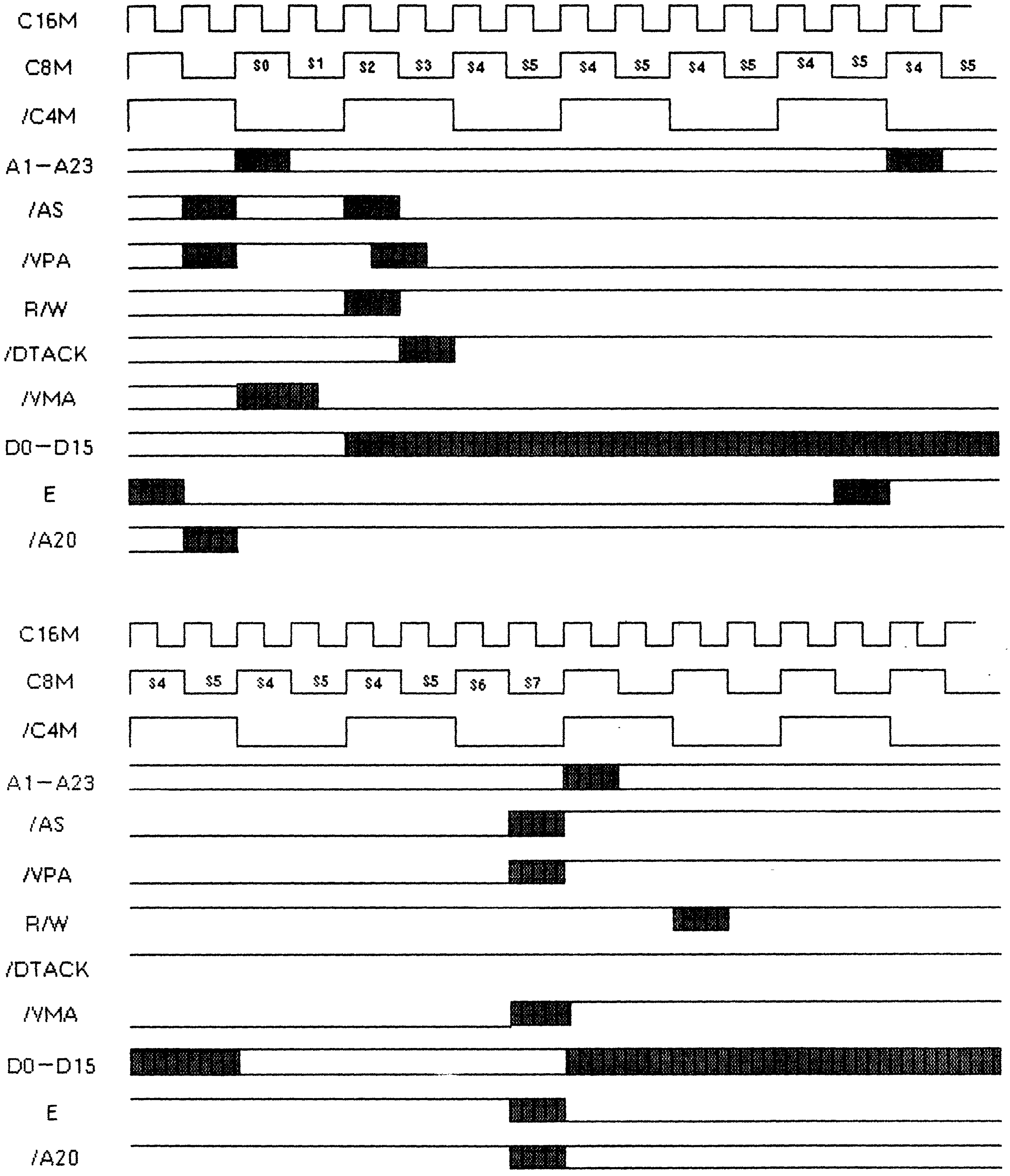

**Figure 1.8 68000 VIA read cycle (Best case)**

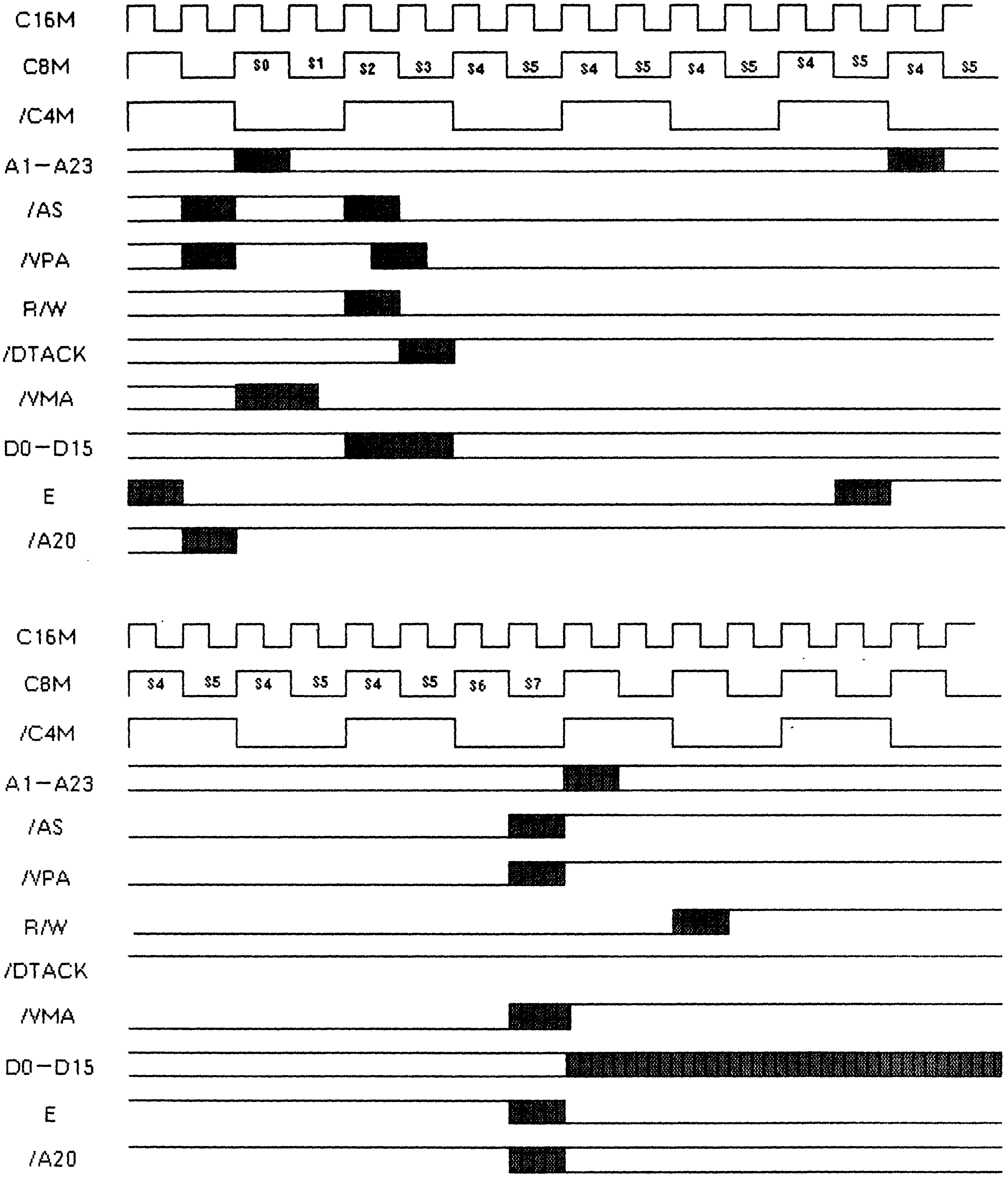

**Figure 1.9 68000 VIA write cycle (Best case)**

## 2.0 Disk State Machine

The disk state machine consists of: the PALs Endcnt, Shift, Bcount, Misc, Dstate, and Sync; a nine bit DMA address counter; an Error Correction Code (ECC) chip; a sixteen bit shift register; and part of the VIA. The disk state machine should not be confused with the disk servo / clock interface which moves the head on the disk. The disk state machine only deals with the data stream to and from the disk.

The disk state machine is an integral part of the SFS design. It performs the refresh of the dynamic RAMs as well as interfacing the bit stream on the disk into the 1 K byte buffer at the top of RAM. The disk state machine has many internal signals but few external. Basically the 16 data lines from the shifter and the 9 address bits from the counter to the RAM are the majority of the lines going into the rest of the SFS. The DMA control signals: DMA request, DMA read/write, DMA write pulse are the remaining signals to RAM. The disk is attached with 6 lines: data in and out, clock, read and write gate, and sector pulse.

The disk drive is organized into tracks and sectors. A track is a concentric circle on either side of the disk on which data can be written. 'Nisha' has 610 tracks on each of its two surfaces. The track are broken up into sectors. 'Nisha' has 38 sectors per track. Each sector has a header and a data section. The header identifies the track and sector number of the sector. The data section is 532 bytes of data.

The disk state machine operates on sectors. It can format, write, or read a sector. Normally the 1 K byte RAM buffer is set up to contain the header of the sector to be read and the disk state machine state compares this to the data stream coming off the disk. When a match is found the data is read from the disk and the disk state machine posts completion.

## Table 2.1 Signals used in Disk State Machine

| Name | Definition |
|---|---|
| RDGT | Read Gate. Signal to disk to read the data stream passing under the head. |
| SECTOR | Sector pulse. Each sector on the track begins with the positive edge of the sector pulse. The signal is derived from physical cuts in the motor spindle which are detected by a magnetic sensor. |
| RWCK | The Read/Write Clock. This is the 7.5 MHz clock from the disk drive which indicates valid data on the positive edge. When reading from the disk valid data is on /NRZRDATA. When writing to the disk valid data should be present on the /NRZWDATA line. |
| NRZWDATA | Non–Return to Zero Write Data. This is the data line to the disk during a write. Valid data is present on the positive edge of RWCK. |
| /WTGT | Write Gate. Active low signal to disk to write the data from NRZWDATA to the disk. |
| R0–R15 | RAM data bus. The data bus directly connecting the 16 bit shifter to the RAM. |
| DMAREQ | Direct Memory Access Request. The request line from the disk state machine to the memory state machine indicating the shift register will need access to the RAM. The memory state machine will finish the RAM cycle, if present, and will not complete a new one until the signal is deasserted. |
| /DMA | Direct Memory Access. This signal from the memory state machine acknowledges the DMA request and indicates the disk state machine has access to RAM. This signal will wait for 68000 to complete a memory access. This signal will also return access to the 68000 at a memory state machine state that is synchronous with the 68000 execution. |
| /DMAWP | Direct Memory Access Write Pulse. This signal from the disk state machine is used by the memory state machine to assert /CAS to the RAM during a DMA cycle. It is synchronized with the shifter register to perform the parallel load or dump of the register. |
| DA8–DA0 | Direct Memory Address bits 8 through 0. This is low order address bits to the word in memory to which the DMA cycle will be made. It is a 9 bit binary up counter. The higher order bits are all set to one. |
| /CP | Clock Pulse to increment the DMA address counter. |

**Table 2.1 Signals used in Disk State Machine (continued)**

| Name | Definition |
|---|---|
| MR | Master Reset to clear the DMA address counter. |
| CMD1—CMD0 | Command register bits 1 and 0. This holds the disk command currently in use. Format (00), Write (01), Read—ID (10), and Read (11) are the four possible commands. |
| Y3—Y0 | The disk state machine state register. A 4 bit register used to indicate to the rest of the state machine the general function to perform. See Table 2.3. |
| /BCD—/BCA | The 4 bit binary counter. This counts out the 16 bits in each word loaded or read from the shift register. It also provides the state number for a DMA cycle. |
| /XOR | Exclusive—OR of the disk data stream and the shifter data stream. Used to compare the disk header with the header in RAM. |
| /ECCERR | Error Correction Code Error. Asserted if after reading the data and 6 ECC bytes the shift register in the ECC chip is non—zero. This comes from the ECC chip and is connected to the VIA. |
| /CMDCOMP | Command Complete. Asserted during state 2 to indicate the state sequence has completed. |

## 2.1 Misc PAL

The 68000 controls the disk state machine through the disk command latch located in the Misc PAL and the /START line from the VIA. The disk command latch is a four bit latch which encodes the four command sequences the disk state machine can execute: Format, Read-ID, Read, and Write. The Decode PAL decodes the address space for the latch as $D8 0000 — $DF FFFF and qualifies the select with Address Strobe (/AS). The /DISK select only indicates that the address lines are stable during the access so the address lines indicate the command. Table 2.2 shows the relationship of the four commands to the disk command latch. The disk command latch also controls the state of the red/green Light Emitting Diode (LED) which is used to indicate the status of the SFS. The X's in the table indicate the state of these bits is not changed by this address access. Thus changing the state of the LEDs has no affect on the disk command selected. Note that only A1—A3, A19—A23 of the 68000 address lines are decoded so many address combinations will work but $DF FFFx is recommended.

The red/green LED is unusual in that both the red and green LED elements are in the same case. This results in yellow being produced when both red and green are active. The LEDs are dissabled by the low level of the control signal (RED, GREEN) shorting out the LED since the low level of 0.5 V is well below the 2.4 V the LED requires from the 200 Ohm pull—up resistor to +5 V.

**Table 2.2 Disk Command Latch**

| A3 | A2 | A1 | Function | CMD1 | CMD0 | RED | GREEN |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Turn off both red and green LEDs | X | X | 0 | 0 |
| 0 | 0 | 1 | Turn on red LED, turn off green LED | X | X | 1 | 0 |
| 0 | 1 | 0 | Turn off red LED, turn on green LED | X | X | 0 | 1 |
| 0 | 1 | 1 | Turn on both read and green LEDs | X | X | 1 | 1 |
| 1 | 0 | 0 | Format sector | 0 | 0 | X | X |
| 1 | 0 | 1 | Write sector (match header) | 0 | 1 | X | X |
| 1 | 1 | 0 | Read—ID sector (no header match) | 1 | 0 | X | X |
| 1 | 1 | 1 | Read sector (match header) | 1 | 1 | X | X |

## 2.2 Dstate PAL

The Dstate PAL generates the state sequence which the disk state machine will follow in executing one of the four commands. The signals Y3—Y0 are the four bit state. The state is formed each positive edge of the read/write clock (RWCK) which is the clock supplied by the disk drive. This clock is 7.5 MHz on the 'Nisha' disk drive. This is the nominal data rate of the bit stream on the disk and is used to clock the Non—Return to Zero (NRZ) data to and from the disk drive. When data is read from the disk the clock is actually separated from the data so the exact frequency varies with motor speed of the disk. Also the transition of starting to read and ending the read causes the RWCK to remain high while it resynchs to or from the crystal frequency.

Each of the state numbers has a specific meaning to the DMA controls, the disk interface, the shift register, and the address counter. Some states have slightly different functions depending on the command bits. Table 2.3 summarizes the functions of each state number qualified by the command.

Each command has a unique state sequence as shown in Figures 2.1 through 2.4. The Format state sequence is the simplest as it merely waits for the sector pulse and counts the number of words in each state. The Write state sequence is more complex as it matches the sector header before beginning a write. The Read state sequence has the same constraint. The Read—ID state sequence is simpler since it doesn't match headers.

## Table 2.3 Summary of Disk State Machine State Numbers

| State | Command | Description |
|---|---|---|
| 0 | all | Wait for the sector pulse. This syncs the state machine to start the sector on the leading edge of the sector pulse. /START must also be asserted. |
| 1 | all | Not used. |
| 2 | all | Completion of sector operation. This causes /CMDCOMP to be asserted and waits for the deassertion of /START. Dummy reads are performed to refresh RAM. |
| 3 | all | Wait to start next command. This state waits for /START to be asserted. RAM is refreshed. |
| 4 | Read, Write | Header compare failed. In comparing the header of the sector the header in RAM did not compare so this sector will be skipped. RAM is refreshed. |
| 5 | all | Not used. |
| 6 | all | Not used. |
| 7 | Read, Write, Read-ID | Synch up to header bit stream. This state reads the bit stream from the disk waiting for the first one in the field of zeroes to align the 16 bit shift register. |
| 8 | all | Not used. |
| 9 | Read, Write | Compare bits from the disk header with the RAM header. Stay in this state until all 6 bytes are compared or a bit does not match. Header is read from RAM. |
| A | Read, Read-ID | Read the ECC syndrome bytes from the ECC chip after reading the sector. ECC is written to RAM. |
|  | Write, Format | Write the ECC bytes to the disk from the ECC chip. |
| B | all | Not used. |
| C | Read, Read-ID | Sync up to the data bit stream. This state reads the bit stream from the disk waiting for the first one in the series of zeroes to align the 16 bit shift register. |
|  | Write, Format | Write the field of zeroes with a one at the end to preceed the data. |

## Table 2.3 Summary of Disk State Machine State Numbers (continued)

| State | Command | Description |
|---|---|---|
| D | Read, Read—ID | Skip the write seam preceeding the data. The read gate is disabled to keep the read chain from try to derive the clock written by different writes. |
| | Write, Format | Shut down the read chain before writing the synch stream leading to the data. This prevents the write current from saturating the read amplifier. A saturated read amplifier requires a long recovery time. |
| E | Read, Read—ID | Start the ECC chip and read the data to RAM. |
| | Write, Format | Start the ECC chip and write the data from RAM. |
| F | Read, Read—ID,<br>Write | Skip the first ten bytes of data from the disk after the sector pulse. Read RAM for first word of header. |
| | Format | Skip the first bit after the sector pulse. |

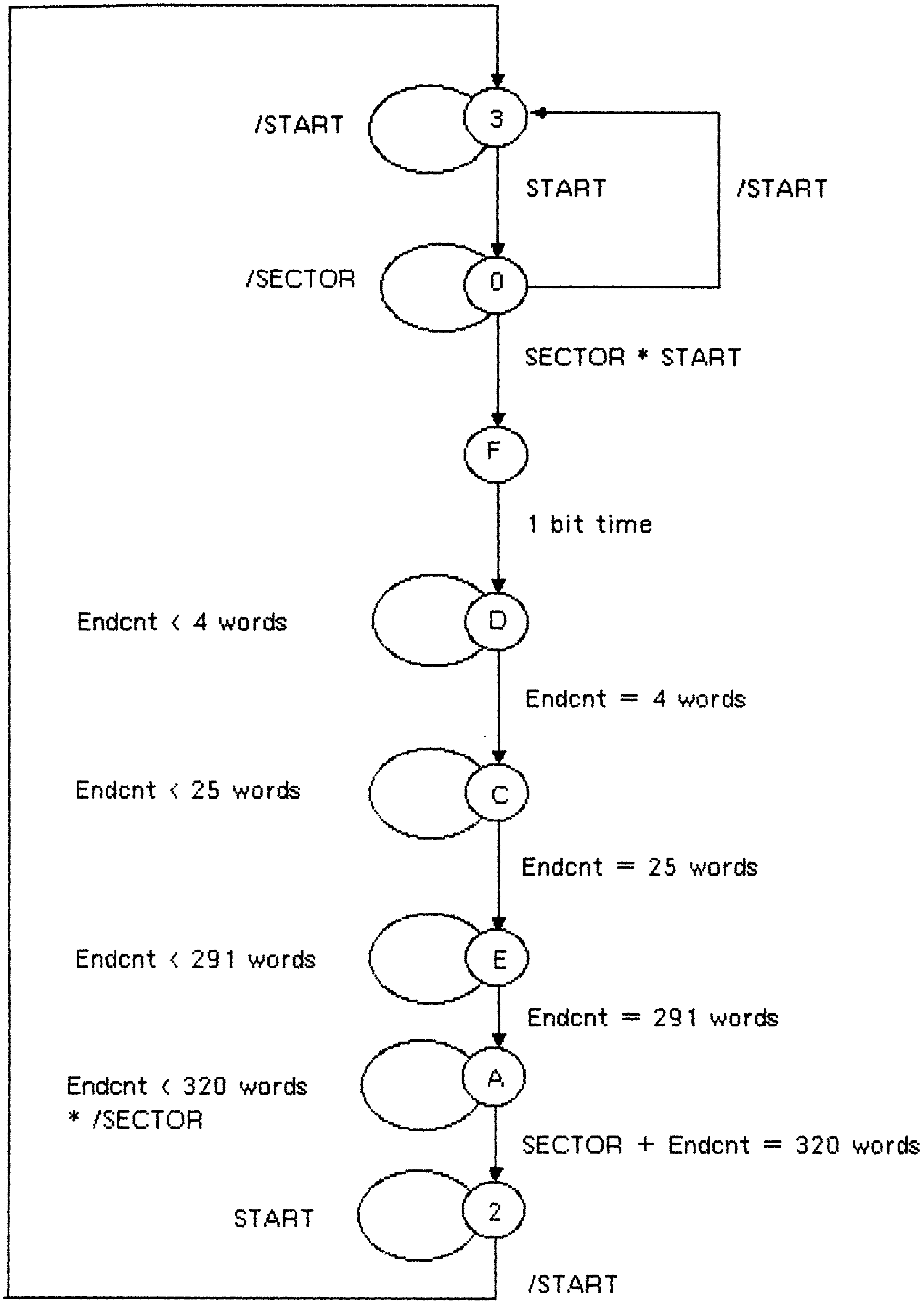

**Figure 2.1 Format state sequence**

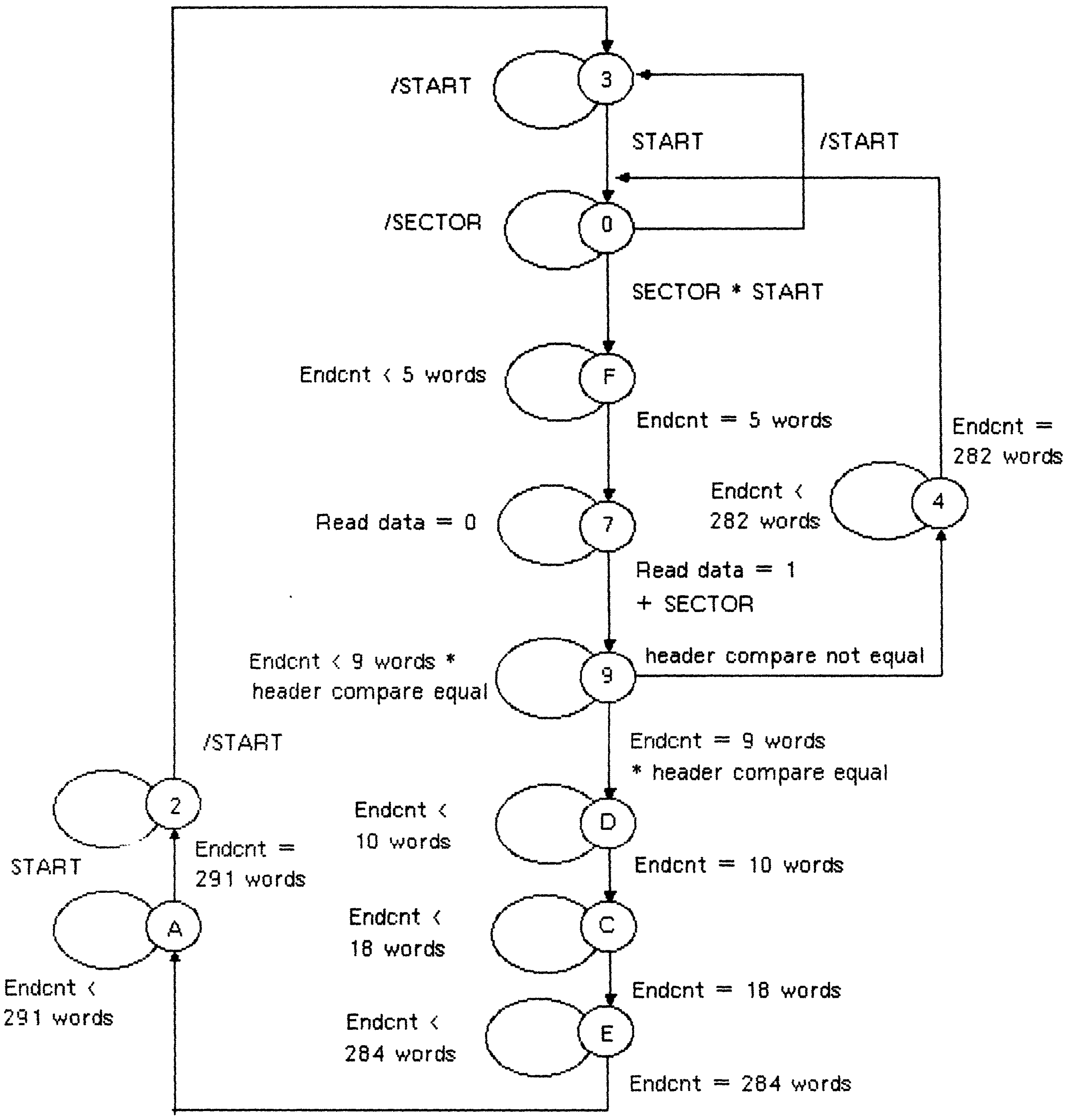

**Figure 2.2 Write state sequence**

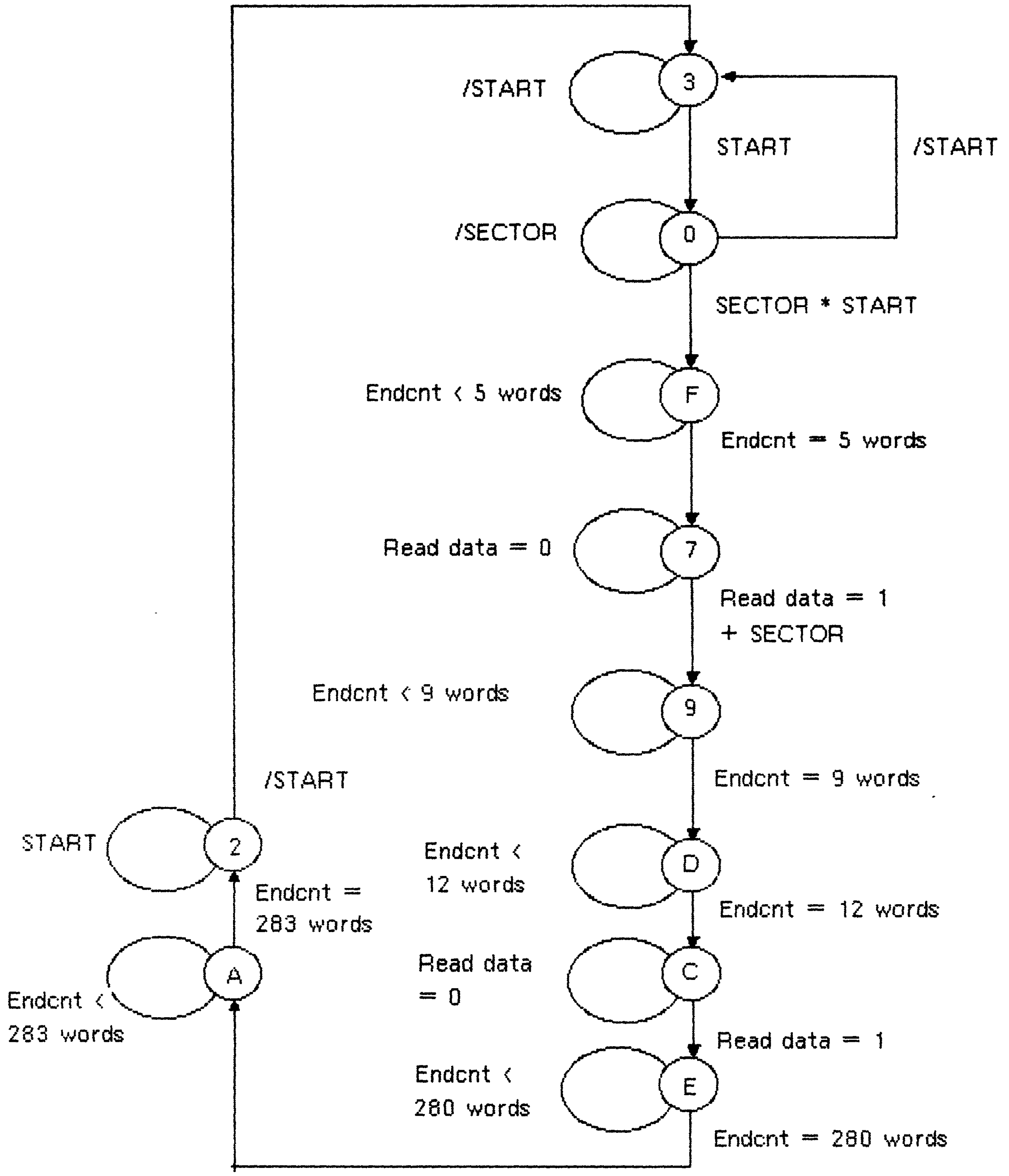

**Figure 2.3 Read—ID state sequence**

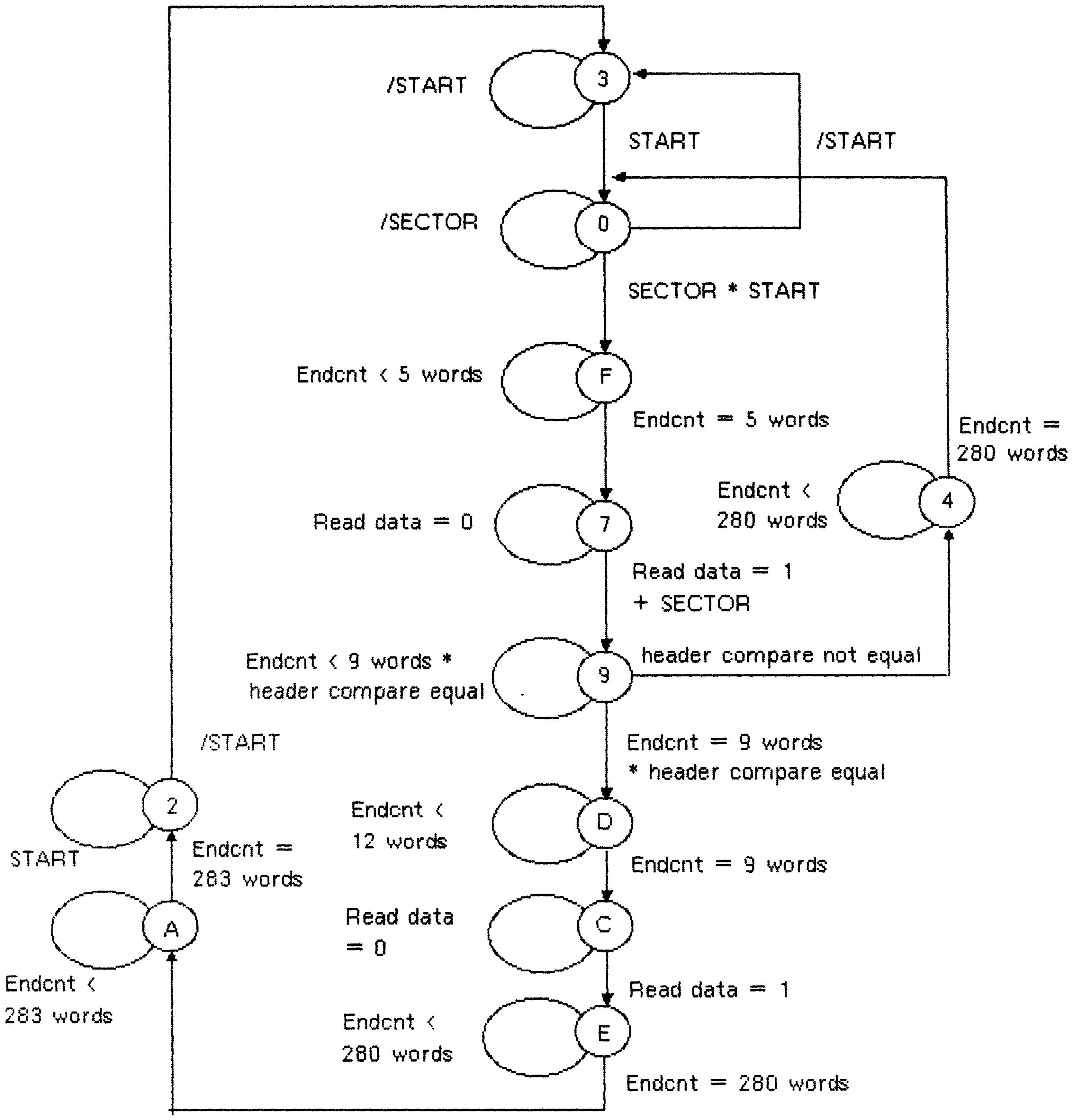

**Figure 2.4 Read state sequence**

## 2.3 Endcnt PAL

The Endcnt PAL is programmed as an asynchronous address decoder. It decodes the low order 9 bit address to the 1 K byte RAM buffer at the top of RAM. This address is formed in the 8 bit binary up counter (LS393) and the ninth bit in the Misc PAL. Normally this counter increments by one every 16 RWCK clocks to generate the next RAM address as a sector is being written to or from RAM.

The Dstate PAL uses the output signal /ENDCNT as a state transition signal in most states. The signal is asserted when the address, command, and /STCHG are all valid. The /STCHG signal is generated by the Bcount PAL and is used to guarantee the correct bit time relative to the 16 bit count for the Dstate PAL to receive /ENDCNT. Thus Endcnt is the means by which the number of word times spent in each state is determined. Note that in Figures 2.1 — 2.4 the Endcnt for a state is not always the same for each state sequence.

The address counter is not always counting by one every sixteen RWCK clocks. During state 7 for instance the counter is not incremented and instead the presence of a one from the disk data stream causes the state transition.

## 2.4 Bcount PAL

The Bcount PAL is basically a 4 bit binary up counter. It counts out the 16 bit times of the disk RWCK clock for each RAM word to be shifted to or from the disk. Normally it is free running but is can be synchronized. States 7 and C wait for the bit stream from the disk to go from zeroes to a one. This one restarts the count at the zero count so the data bits are aligned into the words they are in RAM.

The Bcount PAL is also the source of synchronizing information for DMA activity. Figure 2.5 shows a DMA read to RAM and Figure 2.6 shows a DMA write to RAM. The DMA request line (DMAREQ) is asserted 5 counts before the shift register must be loaded or dumped. This allows the 68000 to finish a RAM memory cycle and be held off from starting a new RAM memory cycle. The DMA write pulse (/DMAWP) is the actual data strobe indicating valid data is present in the shifter or the RAM should present valid data to the shifter. The memory state machine uses /DMAWP to assert /CAS to the RAM chips. When data is to be written from the shifter to RAM the /DMAWP is asserted for the second half of the RWCK cycle from the disk. This assures the data from the shifter is valid as the RAM will write the data on the falling edge of /CAS. When the data is to be read from RAM into the shifter the /DMAWP is asserted 2 RWCK cycles before the shifter will need the data to be sure the data is valid.

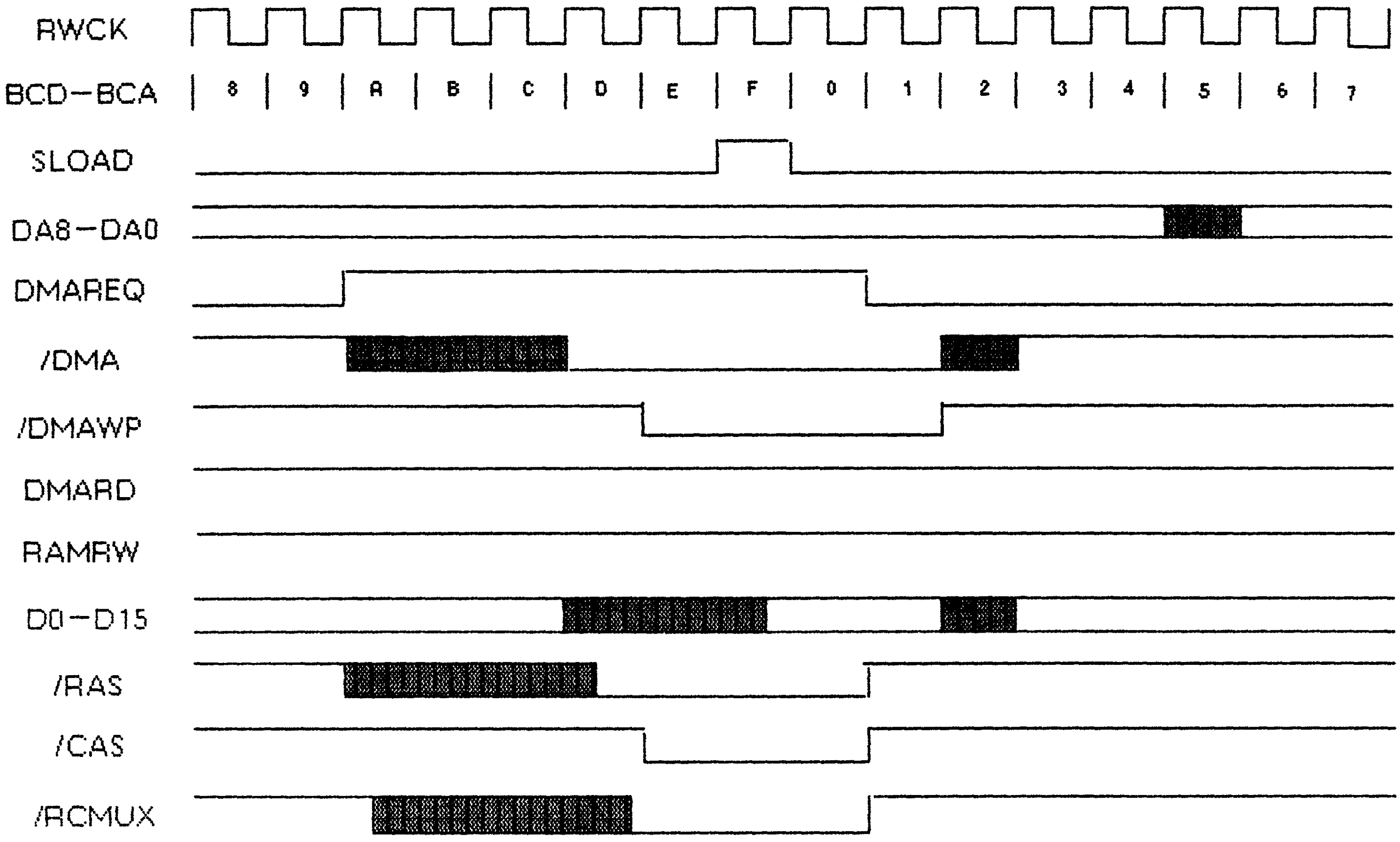

**Figure 2.5 DMA read cycle**

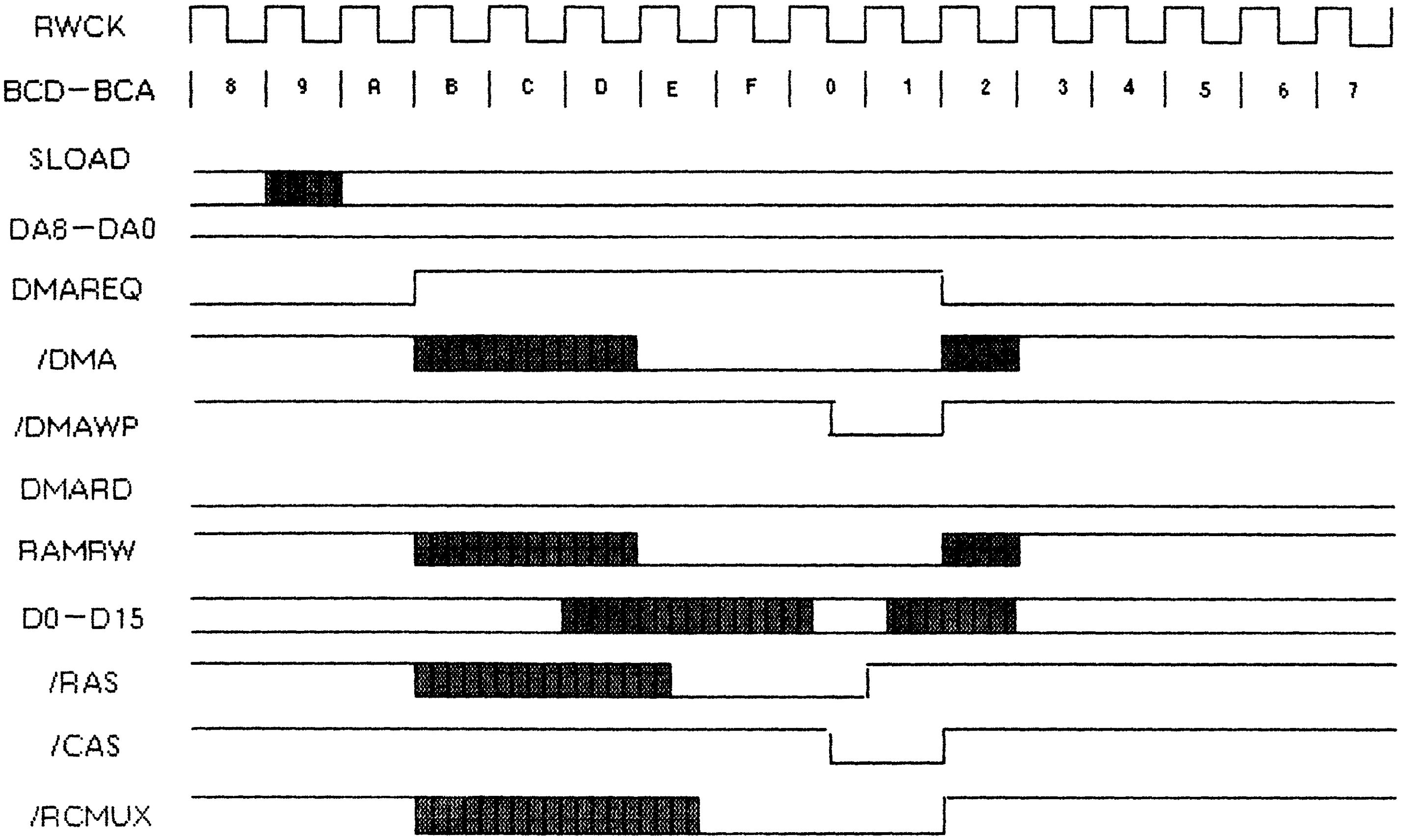

Figure 2.6  DMA write cycle

## 2.5 Shift PAL

The Shift PAL generates some of the shift register controls, the clock pulse to increment the DMA address counter, the disk state machine command completion signal, inverts the data stream from the disk.

The SLOAD signal is active when the next RWCK should do a parallel load from the RAM into the shifter. The DMARD signal indicates whether the DMA cycle is a read or a write. The /CP signal is asserted to increment the 9 bit binary up counter which is normally the next DMA address to be accessed. /CMDCOMP is asserted when the disk state machine is in state 2 to indicate the completion of the command.

## 2.6 Sync PAL

The Sync PAL is basically a double rank synchronizer to bring in signals not synchronous with the RWCK and make them synchronous. It also selects which clock will run the disk state machine — RWCK or /C4M.

The /START signal from the VIA which is used to start the execution of a command is feed into the PAL twice to sync it up. The Sector pulse (SECTOR) is likewise brought into sync.

The data stream from the disk (/NRZRDATA) is also brought into sync with additional function. When the internal /C4M clock is used to clock the state machine or during a sector pulse the data is set to ones. This prevents the state machine from hanging in states that require a one in the data stream to transition to the next state.

The selection of external or internal clocks is with the /SERVORST signal from the VIA. This signal is normally used to reset the disk servo. However it also selects the /C4M clock to be the disk state machine clock when asserted. This mode allows the SFS to function without a disk attached.

Changing the clock source requires care. The 68000 must be executing code in ROM and the /STBIT and SERVOCLK must be asserted when /SERVORST is to be changed. This suppresses the disk state machine clock so no glitches occur during the transition.

## 2.7 ECC Chip

The Error Correction Code chip is a CMOS gate array which contain a 48 bit shift register, a 6 bit binary up counter, and some control logic. The Chip is provided to generate the ECC polynomial on the fly as the data stream is being written or read from the disk. The particular polynomial used allowes the detection of single bursts up to 48 bits in length and correction of errors up to 12 bits in length. The correction is not performed by the chip but the 68000 can take 6 byte syndrome created by the shift register and perform the correction using a software algorithm.

The ECC is necessary because the surface of the disk is not perfect and local defects in the media can cause the loss of data. These defects are assumed to be small, on the order of a few bits, and normally they are found during the initial testing of the disk. Once found the defects are recorded in a spare table so that they can be avoided. However not all defects are apparent during the initial testing. Thus through the use of the drive additional sectors may be found and marked bad. The ECC provides data integrity to these marginal sectors by allowing small errors to be fully corrected.

The ECC chip is normally dormant. Only during the reading or writing of the sector data does it function. This is state E of the disk state machine and the ECC chip watches the state bits Y3—Y0 waiting for this state. This state is preceeded by state C which resets the shift register. The shift register has several tap points which are exclusive —ored together with the disk data stream. The result is fed into the shift register.

When the state C ends the transition to state E causes the shift register to count 48 bits times with the 6 bit up counter as it shifts out the content of the shift register. Each bit of the shift register is tested for non—zero. If any bits are non—zero an ECC error has been detected and /ECCERR will be asserted.

Further information regarding this ECC polynomial is contained in **Single Burst Error Correction Using a 48 bit Computer Generated Code**, Neal Glover, 7—24—82.

## 2.8 Disk RAM Buffer

The disk state machine uses the top 1 K bytes of RAM as a sector buffer. This is the addresses $01 C000 — $01 FFFF in a 128 K byte system and $07 C000 — $07 FFFF in a 512 K byte system. This is the address space to which all DMA accesses to RAM occur. The buffer is organized differently for each of the four commands. Table 2.4 shows the organization of the buffer for each of the commands. The table is organized by offset word addresses which is how the disk state machine deals with RAM. However the 68000 deals with byte addresses so the offset in the table should be doubled for 68000 memory references.

The format command is the simplest use of the buffer. The contents of the buffer are simply written out to the next sector that comes under the disk head. Thus care must be taken to disable interrupts and monitor the index pulse to assure the correct sector is written. The contents of RAM are written out until the next sector pulse so the next sector must be skipped. Thus a two to one interleave is required for format.

The write command both reads and writes the disk. The 6 byte header is read from the disk and compared to the one in RAM. If they match the data for the sector (D0—D531) is written to the disk. Write commands must be executed with a two to one interleave as the read circuit from the head is saturated by the preceding write and the recovery time is greater than the time to next header.

The read—ID command is again a simple use of the buffer. It simply reads the next sector to come under the disk head. This allows one to one reads of the disk. That is to say the entire contents of a track can be read in one revolution with interrupts disabled. Since no compare of the header information is done a sector which has a bad header can be read with this command.

The read command is the normal way to read the disk. The 6 byte header in RAM is compared with the one on the disk. If they match the data, CRC and ECC bytes are read in from the disk.

Each of the above commands should be executed in the following sequence:

1. Move the disk head into position over the track desired. This is explained in the Disk Servo / Clock section.

2. Setup the disk RAM buffer with the appropriate contents.

3. Set the disk command latch for the appropriate command.

4. Assert /START.

5. Count the number of index pulses. If two or more index pulses occur the header was not found and /START should be deasserted. Appropriate action includes reading all the sectors on the track and possibly reformatting the offending sector or marking it as bad in the bad block table.

6. If the /CMDCOMP is asserted by the disk state machine then the command has completed and /START should be deasserted. It is safe to change the RAM buffer of the command latch.

## Table 2.4 Symbols used in Disk RAM Buffer

| Symbol | Description |
|---|---|
| 00 | This byte must be zero. |
| 01 | This byte must be one. |
| T1 | Track number (high order byte). This is the high order byte of the unsigned 16 bit track number on which this sector is located. The 'Nisha' has track numbers from 0 to 609. |
| T0 | Track number (low order byte). This is the low order byte of the unsigned 16 bit track number on which this sector is located. The 'Nisha' has track numbers from 0 to 609. |
| HS | Head select / Sector number. The upper 2 bits are the head select and the lower 6 bits are the sector number. 'Nisha' has 2 heads (0,1) and 38 sectors (0 – 37). |
| /T1 | The ones complement of T1. Used to verify the value of T1. |
| /T0 | The ones complement of T0. |
| /HS. | The ones complement of HS. |
| D0–D531 | The data bytes of the sector. There are 532 bytes of data in each sector. |
| CRC1–CRC0 | The Cyclical Redundancy Check of the data bytes. The CRC–16 polynomial is used. These byte are not checked so a valid CRC is not necessary on a 'Nisha'. |
| ECC0–ECC5 | The ECC syndrome bytes. They are the 48 bit syndrome of the zero byte, 532 data bytes, and the two CRC bytes. They are inserted by the ECC chip during format and write commands. They are read into the buffer from the disk during read and read–ID commands. The 68000 uses these bytes in a software algorithm to correct the data bytes if a error is detected. |
| **(Boldface)** | The symbols in boldface type are written in RAM by the disk state machine, the normal type symbols are supplied by the user 68000 code. |
| (space) | These memory locations are not used with the indicated command. |

## Table 2.5  Disk RAM Buffer

| MEMORY OFFSET WORD ADDRESS | | FORMAT (00) | WRITE (01) | READ-ID (10) | READ (11) |
|---|---|---|---|---|---|
| HEX | DECIMAL | | | | |
| 0 | 0 | 0 | | | |
| 1 | 1 | 0 | | | |
| 2 | 2 | 0 | | | |
| 3 | 3 | 0 | | | |
| 4 | 4 | 0 | | | |
| 5 | 5 | 0 | 00 T1 | | 00 T1 |
| 6 | 6 | 0 | T0 HS | **00 T1** | T0 HS |
| 7 | 7 | 0 | /T1 /T0 | **T0 HS** | /T1 /T0 |
| 8 | 8 | 0 | /HS 00 | **/T1 /T0** | /HS 00 |
| 9 | 9 | 0 | | **/HS 00** | |
| A | 10 | 0 | 0 | | |
| B | 11 | 0 | 0 | | |
| C | 12 | 0 | 0 | | |
| D | 13 | 00 01 | 0 | | |
| E | 14 | 00 T1 | 0 | **D0 D1** | **D0 D1** |
| F | 15 | T0 HS | 0 | **D2 D3** | **D2 D3** |
| 10 | 16 | /T1 /T0 | 0 | **D4 D5** | **D4 D5** |
| 11 | 17 | /HS 00 | 01 00 | **D6 D7** | **D6 D7** |
| 12 | 18 | 0 | D0 D1 | **D8 D9** | **D8 D9** |
| 13 | 19 | 0 | D2 D3 | **D10 D11** | **D10 D11** |
| 14 | 20 | 0 | D4 D5 | **D12 D13** | **D12 D13** |
| 15 | 21 | 0 | D6 D7 | **D14 D15** | **D14 D15** |
| 16 | 22 | 0 | D8 D9 | **D16 D17** | **D16 D17** |
| 17 | 23 | 0 | D10 D11 | **D18 D19** | **D18 D19** |
| 18 | 24 | 01 00 | D12 D13 | **D20 D21** | **D20 D21** |
| 19 | 25 | D0 D1 | D14 D15 | **D22 D23** | **D22 D23** |
| 1A | 26 | D2 D3 | D16 D17 | **D24 D25** | **D24 D25** |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| 116 | 278 | D506 D507 | D520 D521 | **D528 D529** | **D528 D529** |
| 117 | 279 | D508 D509 | D522 D523 | **D530 D531** | **D530 D531** |
| 118 | 280 | D510 D511 | D524 D525 | **CRC0 CRC1** | **CRC0 CRC1** |
| 119 | 281 | D512 D513 | D526 D527 | **ECC0 ECC1** | **ECC0 ECC1** |
| 11A | 282 | D514 D515 | D528 D529 | **ECC2 ECC3** | **ECC2 ECC3** |
| 11B | 283 | D516 D517 | D530 D531 | **ECC4 ECC5** | **ECC4 ECC5** |
| 11C | 284 | D518 D519 | CRC0 CRC1 | | |
| 11D | 285 | D520 D521 | ECC0 ECC1 | | |
| 11E | 286 | D522 D523 | ECC2 ECC3 | | |
| 11F | 287 | D524 D525 | ECC4 ECC5 | | |
| 120 | 288 | D526 D527 | 0 | | |
| 121 | 289 | D528 D529 | 0 | | |
| 122 | 290 | D530 D531 | | | |
| 123 | 291 | CRC0 CRC1 | | | |
| 124 | 292 | ECC0 ECC1 | | | |
| 125 | 293 | ECC2 ECC3 | | | |
| 126 | 294 | ECC4 ECC5 | | | |
| 127 | 295 | 0 | | | |
| 128 | 296 | 0 | | | |
| 129 | 297 | 0 | | | |
| 12A | 298 | 0 | | | |
| 12B | 299 | 0 | | | |
| 12C | 300 | 0 | | | |
| 12D | 301 | 0 | | | |
| 12E | 302 | 0 | | | |
| 12F | 303 | 0 | | | |
| 130 | 304 | 0 | | | |
| 131 | 305 | 0 | | | |
| 132 | 306 | 0 | | | |
| 133 | 307 | 0 | | | |
| 134 | 308 | 0 | | | |