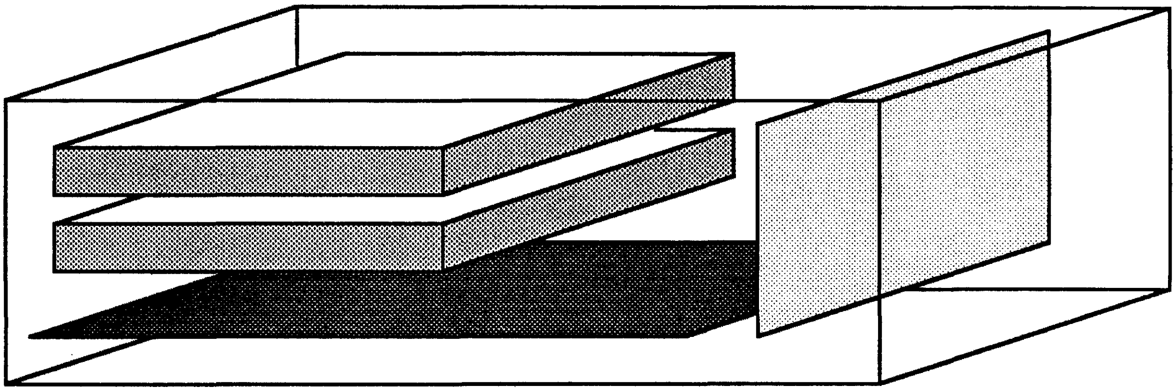
# File Server

# Theory of Operation

(Preliminary)

Michael Dhuey
Ron Hochsprung

4 - 3 - 85



# Table of Contents

Introduction 2

1.0 Memory State Machine 4

1.1 Memory Map 7
1.2 CLOCKS PAL 11
1.3 DECODE1 PAL 12
1.4 DECODE2 PAL 12
1.5 RAMSM1 PAL 13
1.6 RAMSM2 PAL 13
1.7 IRQ PAL 26

2.0 Disk State Machine 27

2.1 Control Register 31
2.2 MISC PAL 32
2.3 DSTATE PAL 33
2.4 ENDCNT PAL 40
2.6 BCOUNT PAL 40
2.7 SHIFT PAL 42
2.8 SYNC1 PAL 42
2.9 SYNC2 PAL 42
2.10 ECC Chip 45
2.11 Disk RAM Buffer 45

3.0 Disk Servo / Clock Interface 50

3.1 Versatile Interface Adapter (65C22) 50
3.2 Real Time Clock 56
3.3 SERIAL PAL 56
3.4 Soft Power 56

4.0 Serial Interface 58

4.1 Serial Communications Controller (8530) 58
4.2 MINI DIN-8 connectors 59

5.0 Small Computer Standard Interface (5380) 62

# Introduction

The File Server (FS) is designed to be a shared 20 (or 40) M byte file system for Macintosh. It contains one (or two) 20 Mbyte Winchester technology hard disks, 1 M byte of memory mapped RAM with 8 contexts, 64 K bytes of ROM, an 8 MHz 68000 CPU, two serial ports, and a SCSI compatible port. It is normally a shared resource on AppleTalk 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 FS 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 31.3344 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, VIA, SCSI and SCC access timing. The memory state machine gives priority access of 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 high order 68000 address bits are sent through a memory mapper. This maps the 2 M bytes of virtual RAM into 1024 2K byte pages for each of eight contexts. Each page can be protected from any access or write protected from write access. These protection bits are used in the user state of the 68000. All I/O devices are protected from access in the User state of the 68000.

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 physical 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 driver and receiver chips. The SCC is the 4 MHz Z8530 chip from Zilog and the drivers are two 26LS30 chips and the receivers are two 75175 chips. 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 75175 driver and receiver chips allow RS-232C as well as RS-422 modes of connection. The RS-232C mode provides the additional handshake lines Data Carrier Detect (DCD), Request To Send (RTS), and Clear To Send (CTS). This allows the use of asynchronous auto-answer modems. Port A of the SCC can also support synchronous modems.

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 lines 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 of the VIA 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 FS. It consists of an NCR5380 SCSI controller chip. This 40 pin DIP is the complete interface and drives the SCSI bus directly.

Additional documents related to this design are:

MC68000 16-BIT MICROPROCESSOR, Motorola Semiconductor, 1983;
ZILOG 1983/84 COMPONENTS DATA BOOK, Zilog, 1983;
PAL PROGRMMABLE ARRAY LOGIC HANDBOOK, Monolithic Memories Inc., 1983;
SERIAL CLOCK AND 256 BYTE RAM SPECIFICATION, Bob Bailey, 7-2-84;
50 WATT INTERNATIONAL POWER SUPPLY - THEORY OF OPERATION, David Egner;
SPECIFICATION - 20MB RIGID DISK DRIVE, John Moon, 1-12-85;
NCR 5380 SCSI INTERFACE CHIP PRELIMINARY DESIGN MANUAL, NCR Microelectronics Division;
SCSI SMALL COMPUTER SYSTEM INTERFACE, ANSI X3T9.2/82-2 - Rev. 14, 5 - 2-84.
MINI-DIN CONNECTOR SCHEME, Peter Ashkin, 1-25-85.

# 1.0 Memory State Machine

The memory state machine consists of the PALs: CLOCKS, RAMSM1, RAMSM2, and DECODE1, DECODE2. They generate the control signals for the 68000 CPU, the RAM chips, the RAM address multiplexers, the RAM data bus buffers, the memory map, the ROMs, the VIA, the SCC, the SCSI, and the disk command latch.

The 16 Mbyte address space of the FS is divided into 4 sections: RAM, ROM, memory map, and I/O (Figure 1.1). Each of these sections is 4 Mbytes of the address space. The RAM section represents the first 4 Mbytes of the address space. This section is implemented as a 2 Mbyte logical address space with 2 Kbyte pages. This is mapped to a physical address space of 1 Mbytes. The ROM section represents the second 4 Mbyte section. It is a 64 Kbyte ROM physical address space. The memory map section is the third 4 Mbyte section. It holds the physical page address and protection for each of the logical pages. There are 8 versions or contexts of the logical address space. This is to allow rapid context switching from user to user to supervisor states. When the supervisor state is active context 0 is used. When the user state is active the context register, which is part of the control register, contains the active context number. The top 4 Mbyte section of the address space is the I/O address space. The control register, SCC, SCSI, and VIA occupy the I/O address space. This address space, as well as the memory map, is only accessible in supervisor state.

The FS data bus is separated into two 16 bit components as shown in Figure 1.2. The main component connects the 68000 CPU to the RAM buffers, the memory map, the control register, 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 buffer. The RAM buffer connects 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 buffer prevents 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.

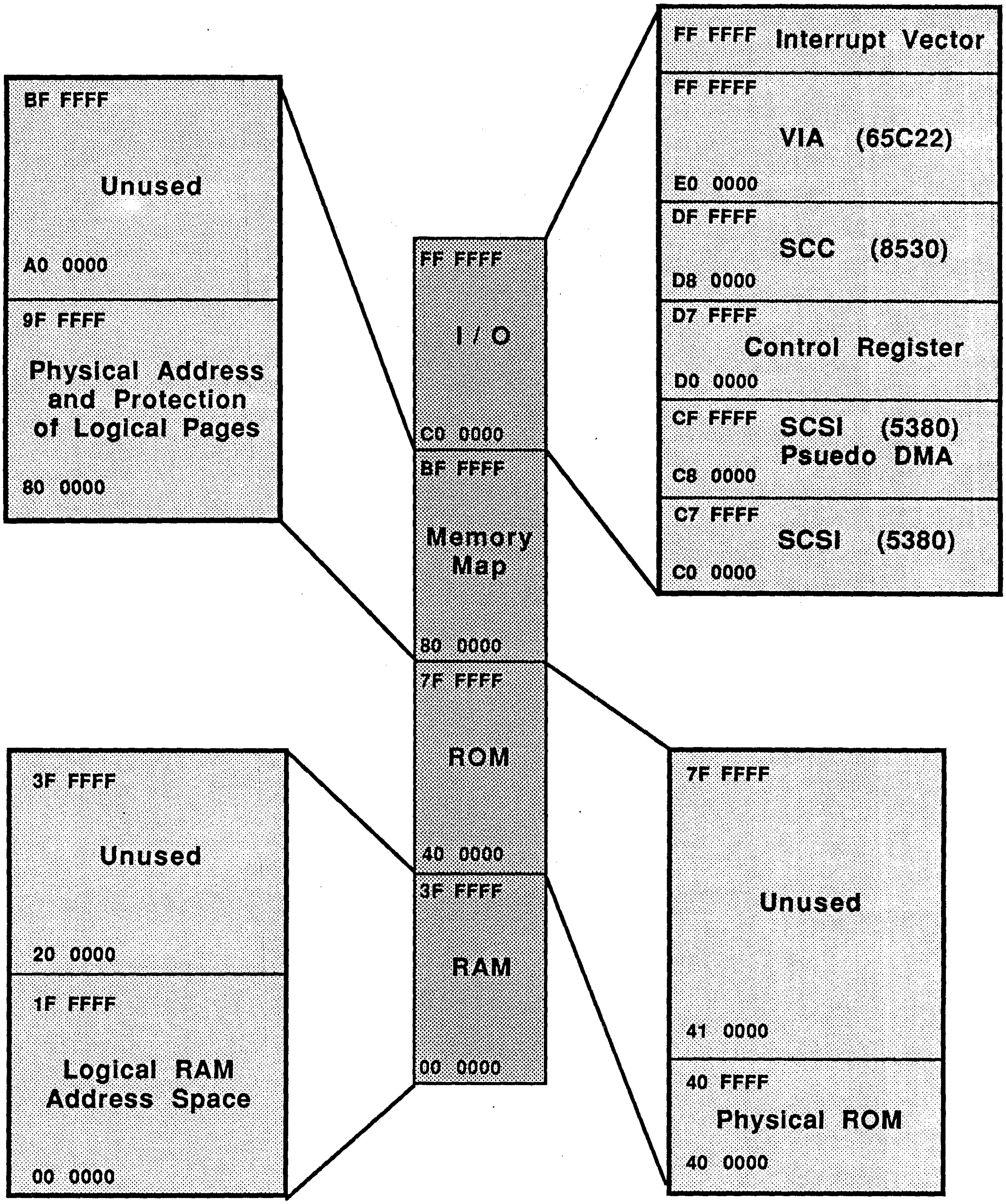


Figure 1.1 Address Space

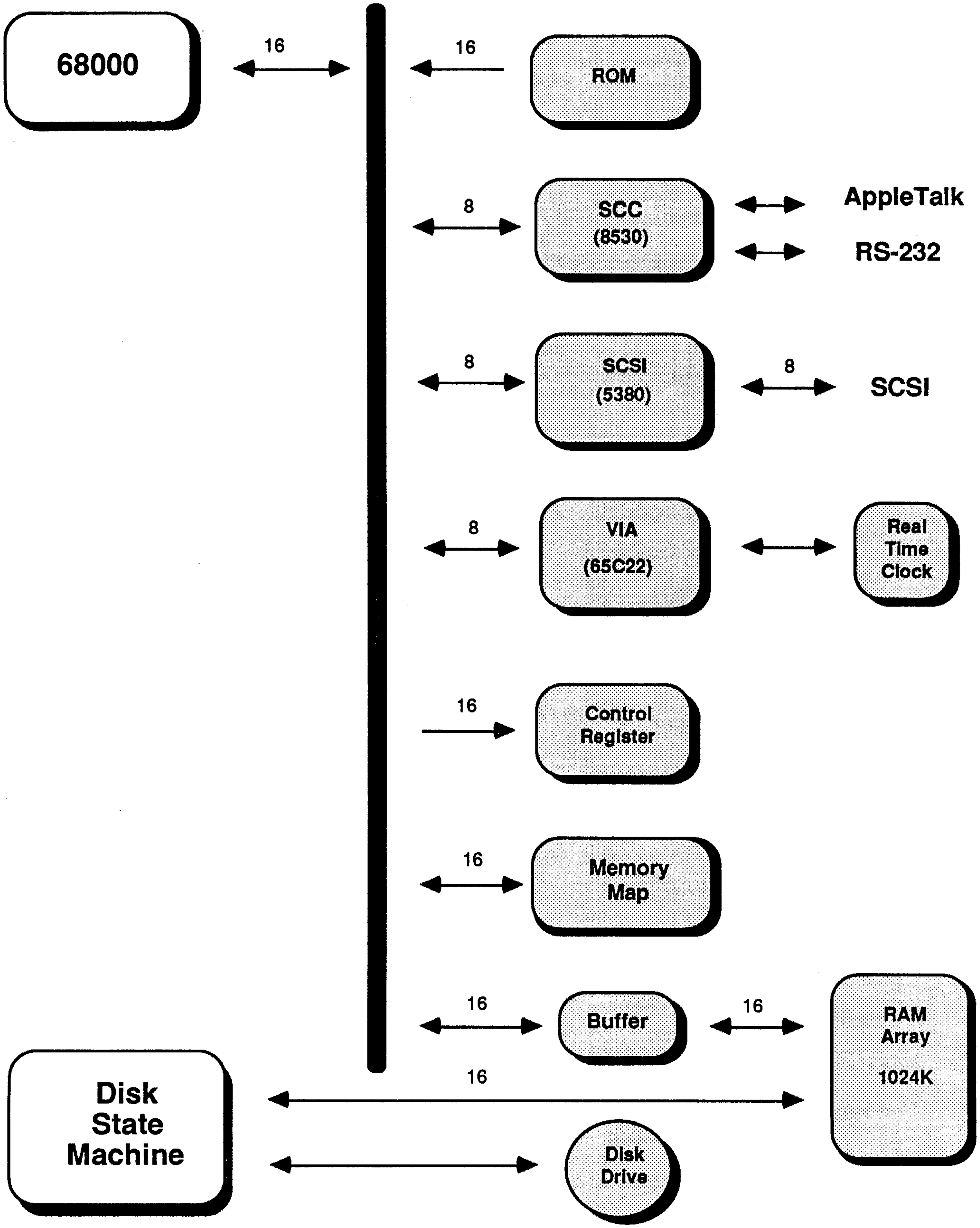


Figure 1.2 Data Bus Block Diagram

## 1.1 Memory Map

The memory map consists of two 8 Kbyte x 8 bit static RAM chips, two 74HCT245 buffers, part of the DECODE PALs, and part of the control register.

The memory map is intended to allow several programs to use the same logical RAM address space while occupying different physical RAM address spaces. This is a requirement of operating systems such as Unix. The FS has eight contexts or versions of logical RAM address space (Figure 1.3). Each context is a 2 Mbyte address space ($00 0000 - $1F FFFF). The supervisor state of the 68000 always uses context 0 when accessing RAM. The other 7 contexts are normally used to run programs in the user state of the 68000. The user state active context number is stored in the Control Register. Thus whenever user state accesses to RAM occur the value in this register indicates which mapping of logical RAM will be used.

Each context of the memory map is divided into 1024 logical pages (Figure 1.5) of 2 Kbytes. The memory map lookup table contains the physical address of each logical page. The mapping is many to one as several logical pages from several different contexts can all point to the same physical page (Figure 1.4). Each logical page has two protection bits - Fault and Write Protect - which indicates whether an access to this page is allowed or if only read accesses are allowed. Thus each access to logical RAM is converted to a physical RAM address through the memory map lookup table (Figure 1.5). If the protection bits indicate a violation and the user mode is active then a Bus Error occurs and a trap to the supervisor state occurs. The instruction which caused the Bus Error is not restartable with the 68000.

The memory map lookup table parallels the logical RAM address space in several ways. First, the context number in the control register indicates which context of lookup table is accessible in the Memory Map address space ($80 0000 - $9F FFFF). Second, the address of each entry in the lookup table is separated by 2 Kbytes. Third, the address of the lookup table entry for a logical RAM address can be simply computed by adding $80 0000 to the RAM address.

Each entry in the lookup table is a 16 bit quantity which is the address of the physical page and the protection bits for that page. Data bit 15 represents the Page Fault bit, bit 14 represents the Write Protect bit, and bits 0 through 11 represent physical address bits 11 through 22. Data bits 12 and 13 are undefined. Setting the Page Fault bit will cause any access to the logical address of the page to result in a Bus Error. Setting the Write Protect bit will cause a write access to the logical page to result in a Bus Error. All accesses to the lookup table must be word wide as the upper and lower data strobes are not used in the access. The bus timing of these accesses are shown in Figures 1.10 and 1.11.

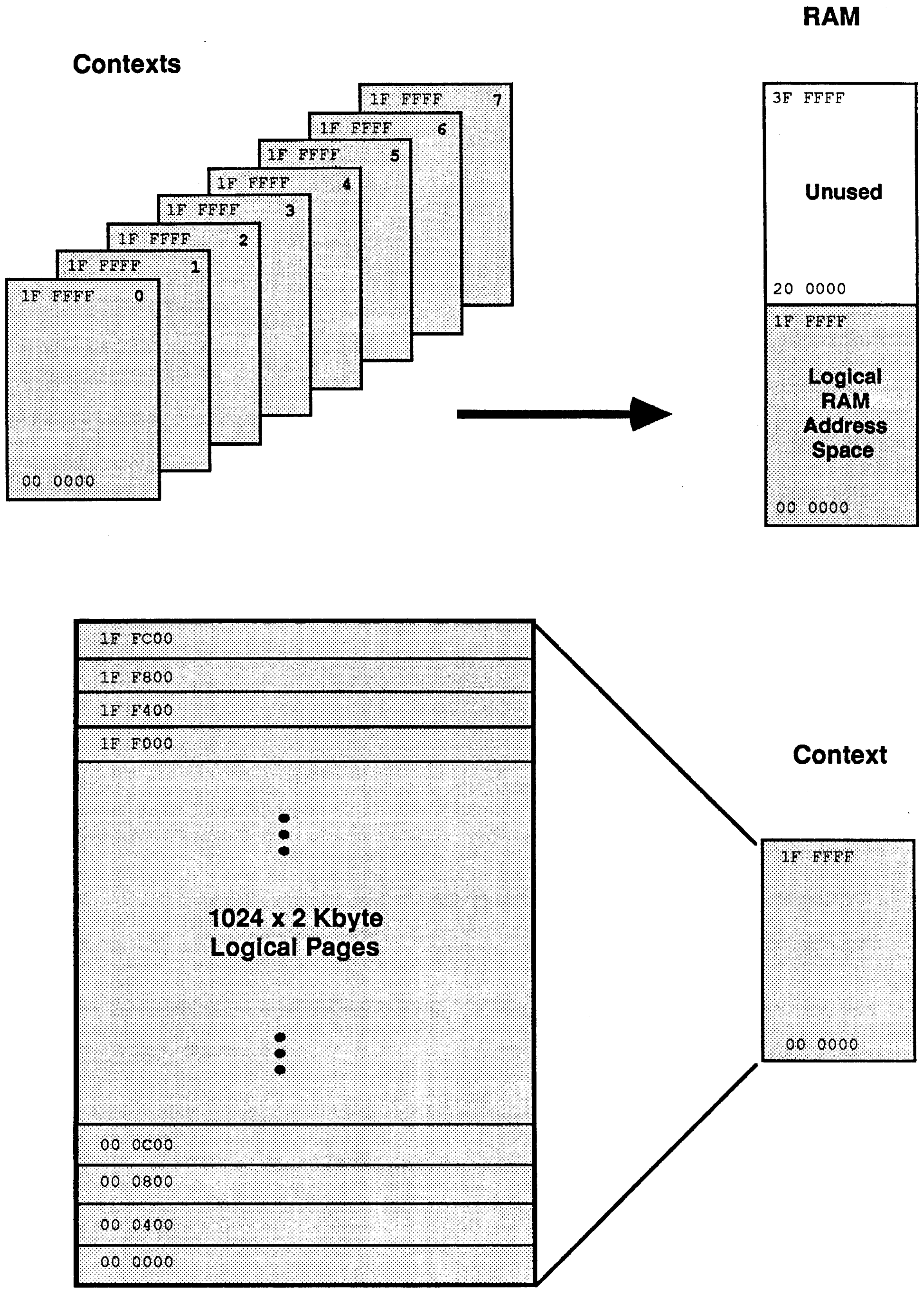


Figure 1.3 Contexts

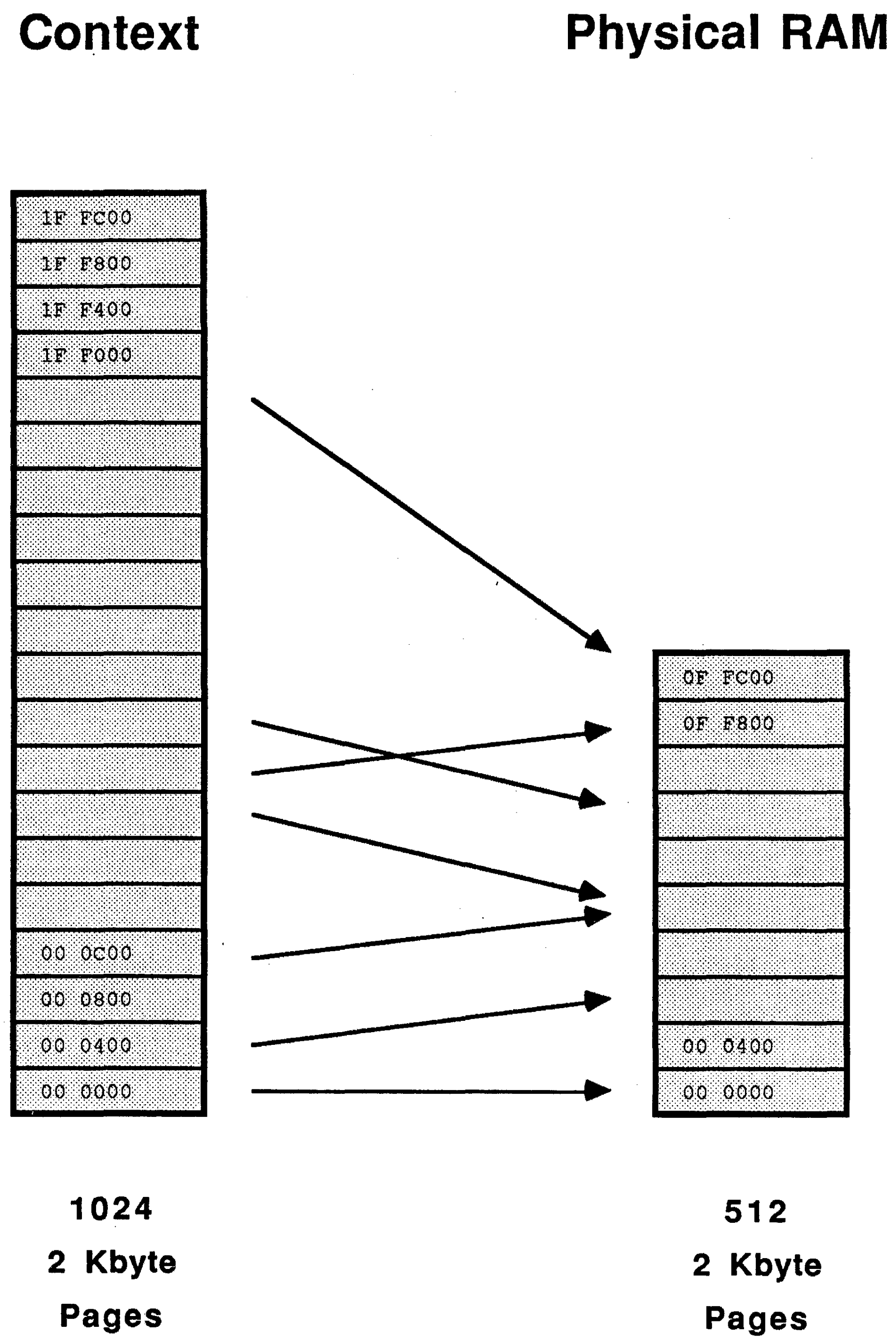


Figure 1.4 Various Memory Mappings

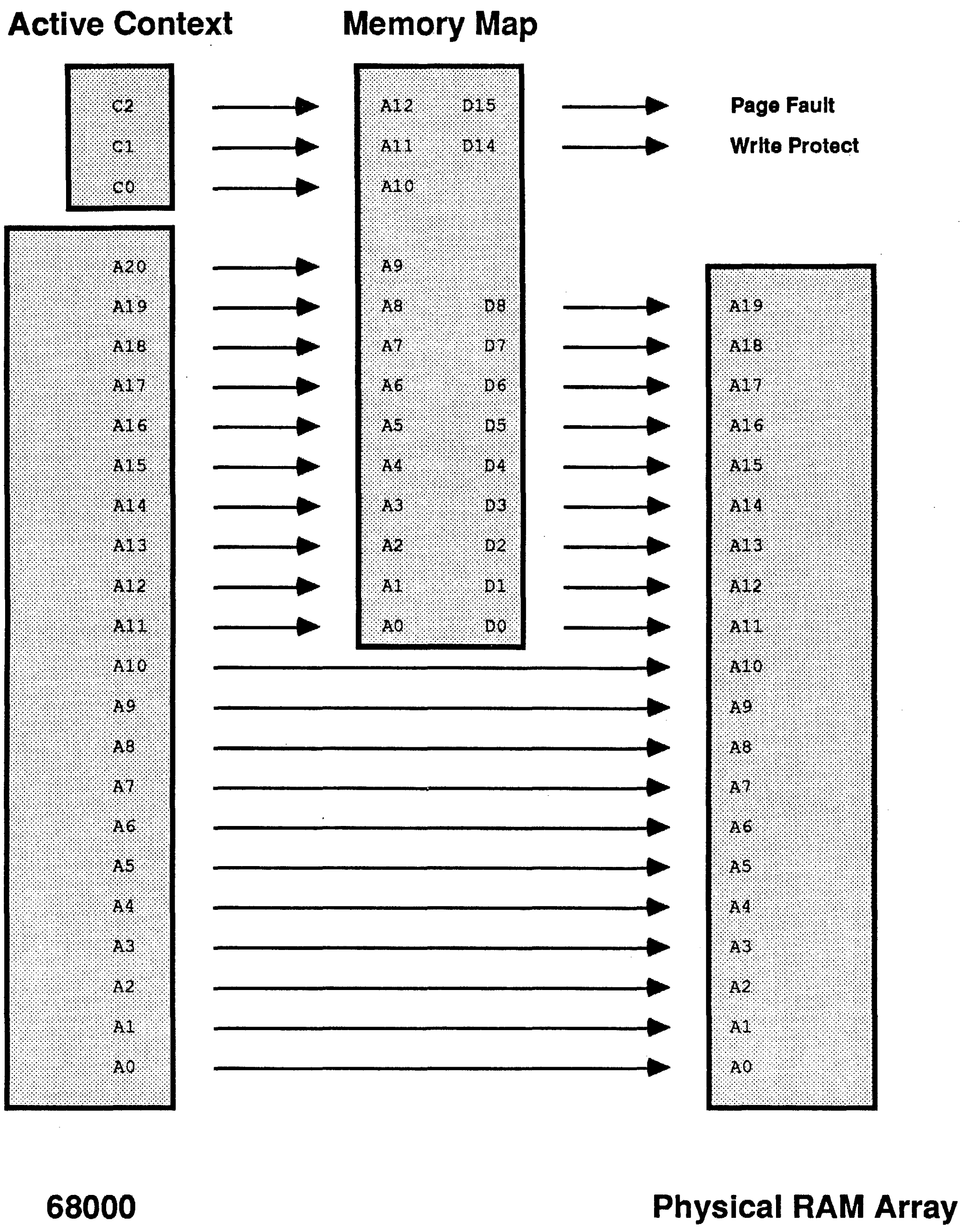


Figure 1.5 68000 to RAM Address lines

## 1.2 CLOCKS PAL

The CLOCKS PAL generates all the timing signals used in the memory state machine. It uses the 15.6672 MHz signal (C16M) from the SYNC2 PAL 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 1.6 shows the phase relationship of the clocks which is used by the RAMSM PALs 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. The /DMA signal is generated by the disk state machine asserting /DMAREQ and neither /RAS signals being asserted. This assertion and deassertion are synchronized with the C8M and C4M clocks so the 68000 will continue a RAM cycle at the appropriate phase.

### Table 1.1 Clock Frequencies

| Clock name | Frequency (MHz) | Period (ns) |
|---|---|---|
| C32M | 31.3344 | 31.91 |
| C16M | 15.6672 | 63.82 |
| C8M | 7.8336 | 127.65 |
| /C7M | 7.3728 (avg.) | |
| /C4M | 3.9168 | 255.31 |
| /C3M | 3.6864 (avg.) | |
| /C2M | 1.9584 | 510.62 |
| /C1M | 0.9792 | 1021.24 |

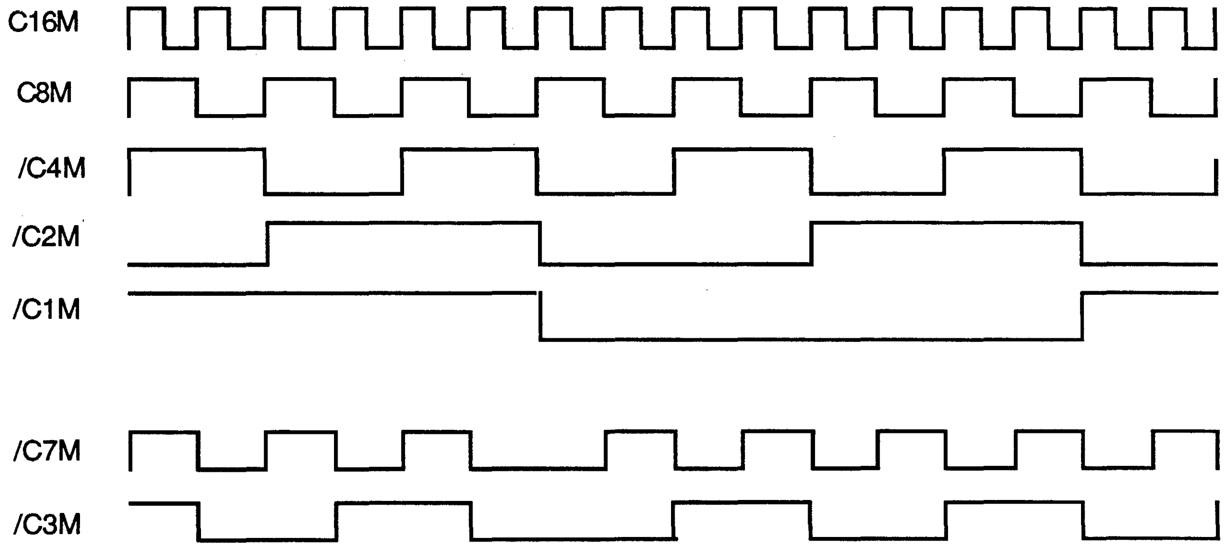


Figure 1.6 Phase relationship of the clocks

## 1.3 DECODE1 PAL

The DECODE1 PAL decodes the RAM, ROM, and Memory Map Address spaces. It also decodes possible bus errors and generates two of the context bits.

Note that the OVERLAY bit causes the RAM and ROM address spaces to be mapped differently. This is to put the ROM in the $00 0000 - $3F FFFF address space to allow the 68000 CPU power-up reset vector to be in ROM at power on. The OVERLAY bit is supplied by the VIA and is high on power-up. The power-up reset vector in ROM points to an address in the $40 0000 - $7F FFFF address space where the ROM code begins execution. Once execution begins the OVERLAY bit is set low. This returns the logical RAM address space to $00 0000 - $3F FFFF. Next the memory map lookup table is loaded to map the logical address space to the physical before RAM accesses begin.

Access to the Memory Map and RAM are controlled by the state of function code 2 of the 68000. When function code 2 is high the 68000 is in supervisor state and all address spaces are accessable. When function code 2 is low the 68000 accesses to RAM are conditional and the memory map is inaccessable. The /BERRIN is the asynchronous indication of a user state violation of the RAM, memory map, or I/O address spaces. It is connected to /BERR which synchronizes the signal to the 68000 bus timings.

The two context bits are selected from the Control Register when in user mode and forced to zero when in supervisor state.

## 1.4 DECODE2 PAL

The DECODE2 PAL decodes the SCSI, SCC, VIA, and control register address spaces. It also generates the asynchronous version of /DTACK - /DTACKIN.

All of these address spaces are accessable only from the supervisor state of the 68000. A user state access will generate a bus error.

The SCSI chip has two address spaces to allow normal access to the registers in the chip and a psuedo DMA mode to transfer data to or from the chip. The psuedo DMA mode lets the chip perform the SCSI bus handshake automatically so the 68000 can simply poll for completion. The bus timing for SCSI accesses are shown in Figures 1.16 and 1.17.

The SCC and SCSI interfaces to the data bus take advantage of a 68000 implementation feature (which is also found on the 68010 and 68020). The feature is the presence of valid data on both the upper and lower data bytes when a byte write occurs. This allows the /UDS and /LDS signals to be the read and write selects, respectively, to the chips. The bus timings for the SCC chip are shown is Figures 1.12 and 1.13.

The /VPA signal is generated for VIA accesses and for autovectoring of interrupts. The VPA signal is generated for the address space $E0 0000 - $FF FFFF. This includes both the VIA ($E0 0000 - $EF FFFF) and interrupt vector ($FF FFFF) address space. The /VPA signal cause the 68000 to execute a 6800 bus cycle which synchronizes to the E clock of the 68000. This is necessary because the VIA was designed for a synchronous bus. The bus timing for VIA accesses are shown in Figures 1.14 and 1.15.

## 1.5 RAMSM1 PAL

The RAMSM1 PAL generates the timings for the /CAS signals to RAM and the /BERR signal to the 68000.

The /CAS signals are controlled from two different sources: the 68000 and the disk DMA cycles. The /DMA signal controls which of the two is in control. When /DMA is active the disk DMA cycles are occuring and the DMAWP signal causes both /CASL0 and /CASU0 to be active. When /DMA is inactive the 68000 is controlling the /CAS lines. These cycles use the /RAS0, /RAS1, /UDS, /LDS signals of the 68000 and the C8M and /C4M clocks. The /RASx signal selects which row of RAMs to access and the /xDS indicates which or both bytes are being accessed.

## 1.6 RAMSM2 PAL

The RAM cycles (Figures 1.8 and 1.9) 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. The DMA request is made before the RAM is actually needed by the disk state machine to allow the 68000 CPU to complete its cycle.

Each of the cycle timings makes 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 RAMSM2 PAL will only start a cycle when the 68000 Address Strobe (/AS) falls and C8M is high and /C4M is low. This allows the RAMSM2 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.

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.

The /SCCWAIT signal is used to delay the /DTACK signal to the 68000 during an SCC cycle. This signal is asserted after the /SCCEN to cause the /DTACK to be delayed two extra C8M clock periods as shown in Figures 1.12 and 1.13.

## Table 1.2 Symbols used in Figures 1.7 - 1.17

| Symbol | Definition |
|---|---|
| **A1-A23** | **Address** bits 1 through 23 of the 68000 (/UDS and /LDS form A0) |
| **/UDS** | **Upper Data Strobe** of the 68000 |
| **/LDS** | **Lower Data Strobe** of the 68000 |
| **/xDS** | **Upper** and/or **Lower Data Strobes**of the 68000 |
| **/AS** | **Address Strobe** of the 68000 |
| **R/W** | **Read/Write** of the 68000 |
| **/DTACK** | **Data Acknowledge** of the RAMSM2 PAL to the 68000 |
| **ROM** | **Read Only Memory** output enable from the DECODE1 PAL |
| **D0-D15** | **Data Bus** bits 0 through 15 |
| **/RASx** | **Row Address Select** from RAMSM2 PAL to each row of RAMs |
| **/CASxy** | **Column Address Select** from RAMSM1 PAL to upper and lower byte of each row of 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 DECODE2 to 68000 |
| **/VMA** | **Valid Memory Address** from 68000 to VIA |
| **E** | E clock from 68000 ( 783.36 KHz ). Clock into the VIA |
| **VA10-VA20** | **Memory Map physical address** bits output by the lookup table |
| **/BERRIN** | **Bus Error Input** used to decode possible bus error conditions |
| **/BERR** | **Bus Error** sent to 68000 synchronized to the bus cycle timings |
| **S0-S7** | **State** numbers of the 68000 bus timings |
| **/MAPOE** | **Memory Map Output Enable** to the 68000 data bus on map read cycle |
| **/MAPWE** | **Memeory Map Write Enable** to load lookup table with entry from 68000 data bus |

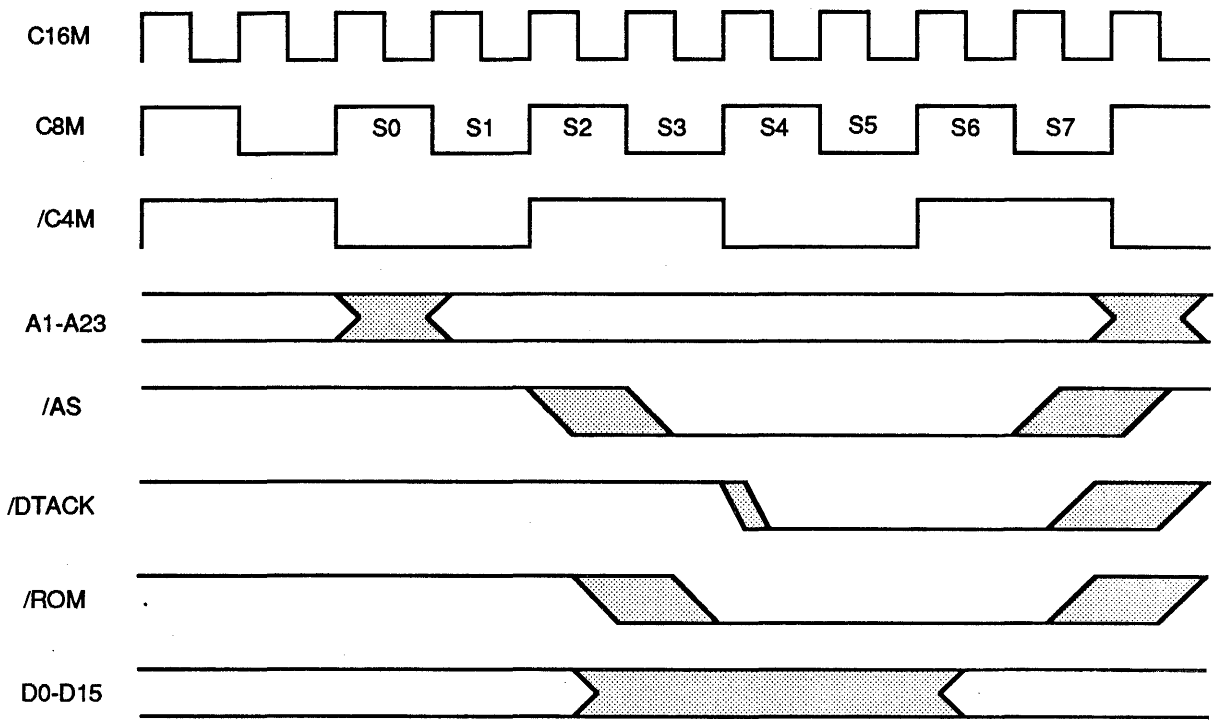


Figure 1.7 ROM cycle

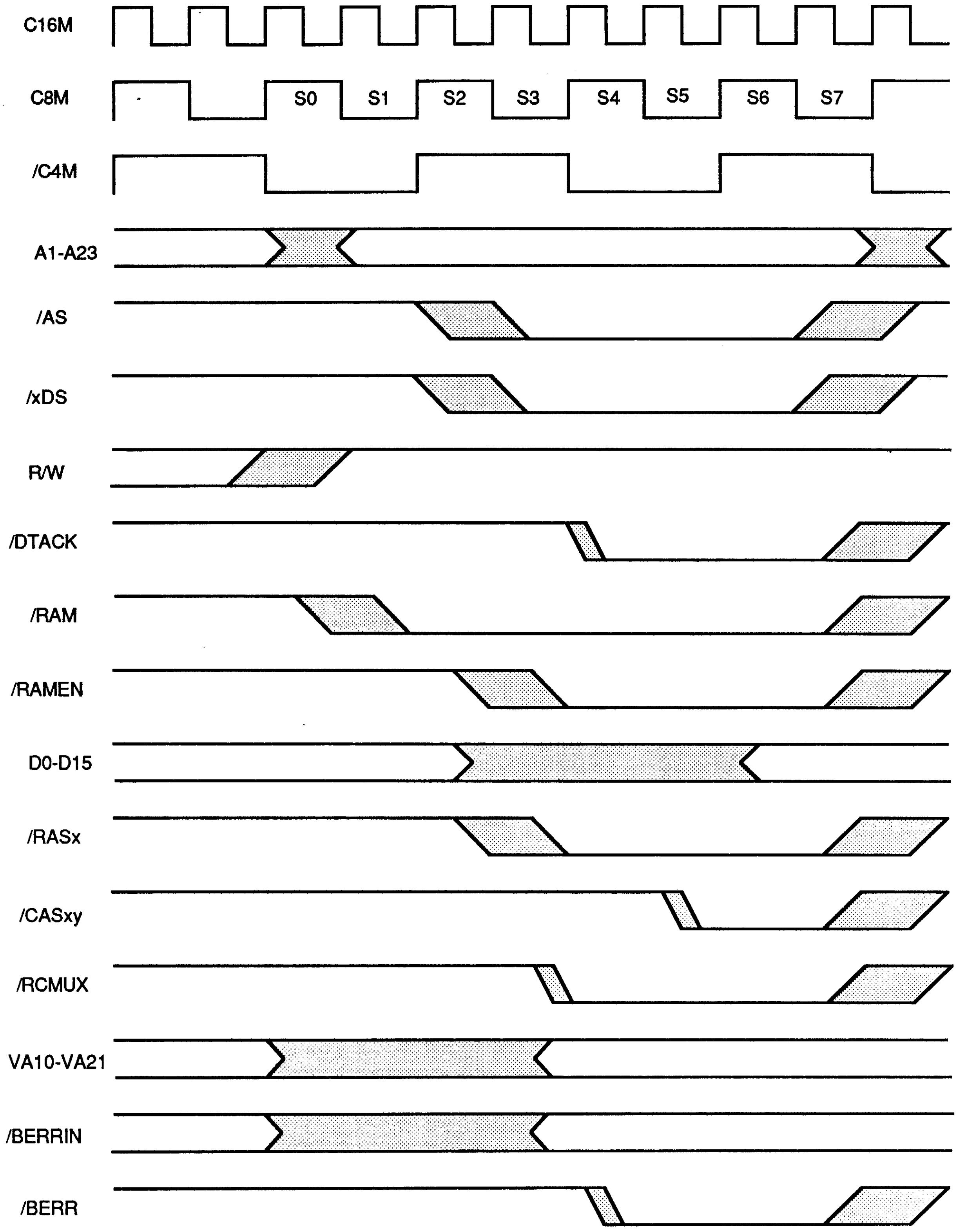


Figure 1.8 RAM read cycle

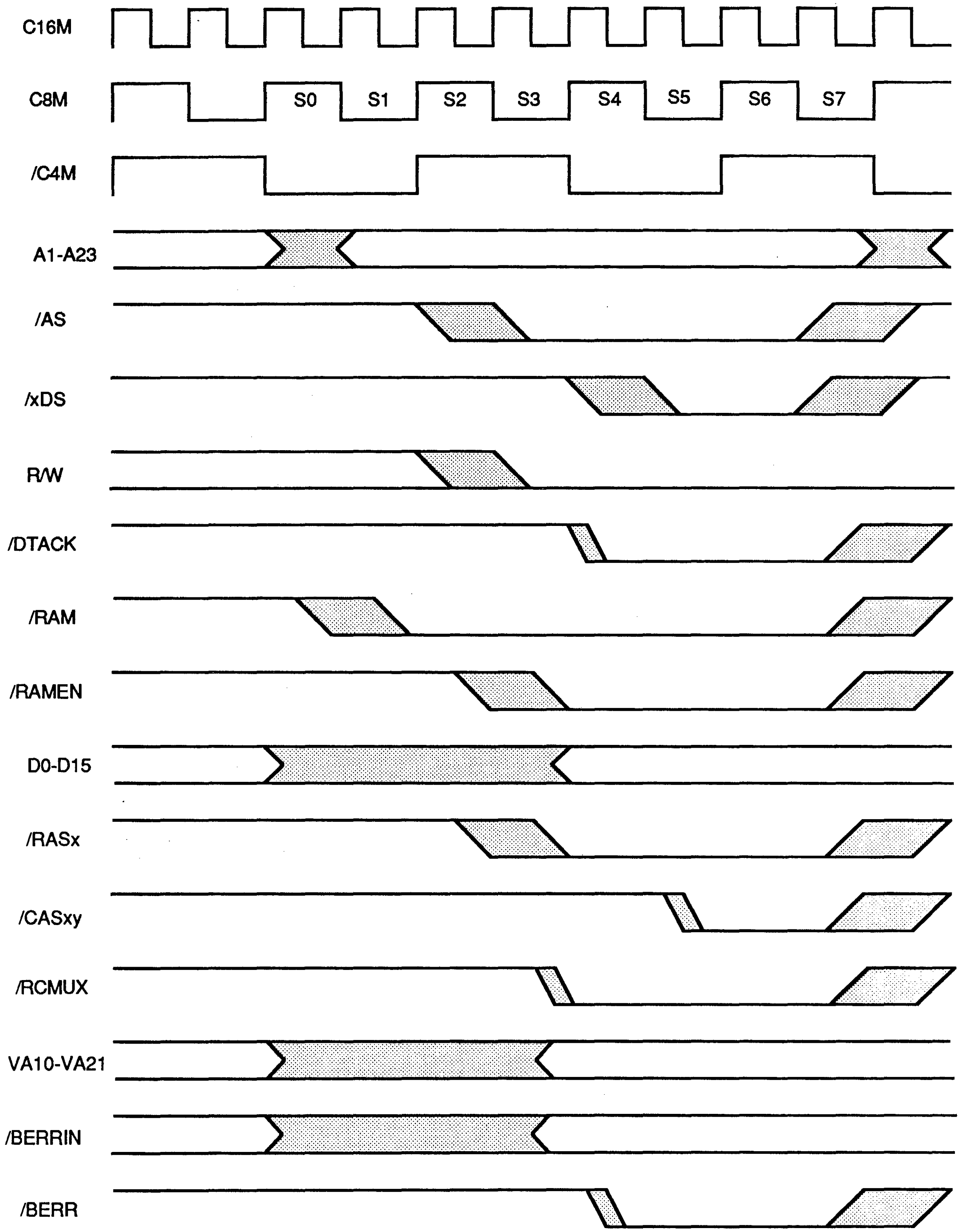


Figure 1.9 RAM write cycle

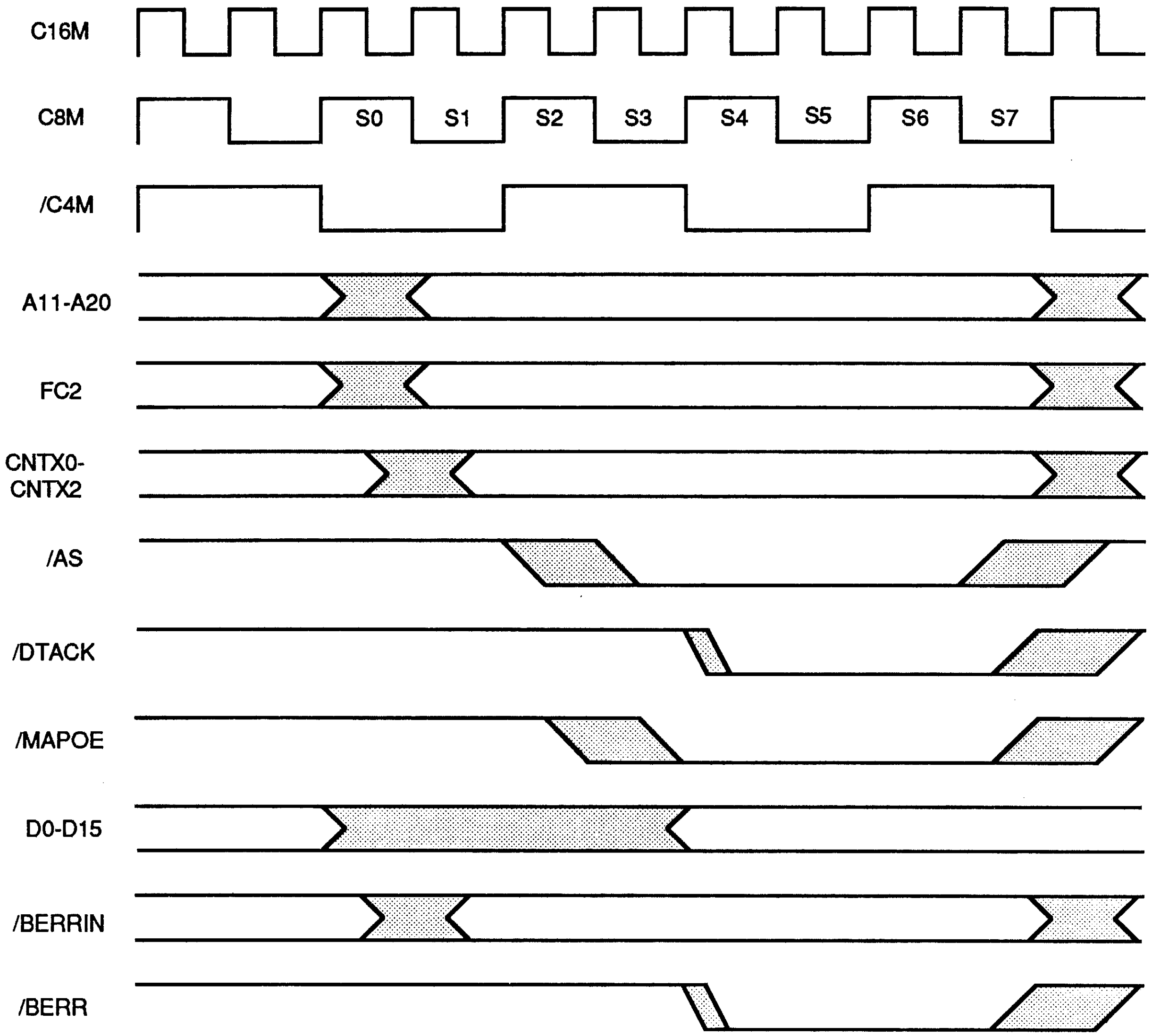


Figure 1.10 Memory Map Read cycle

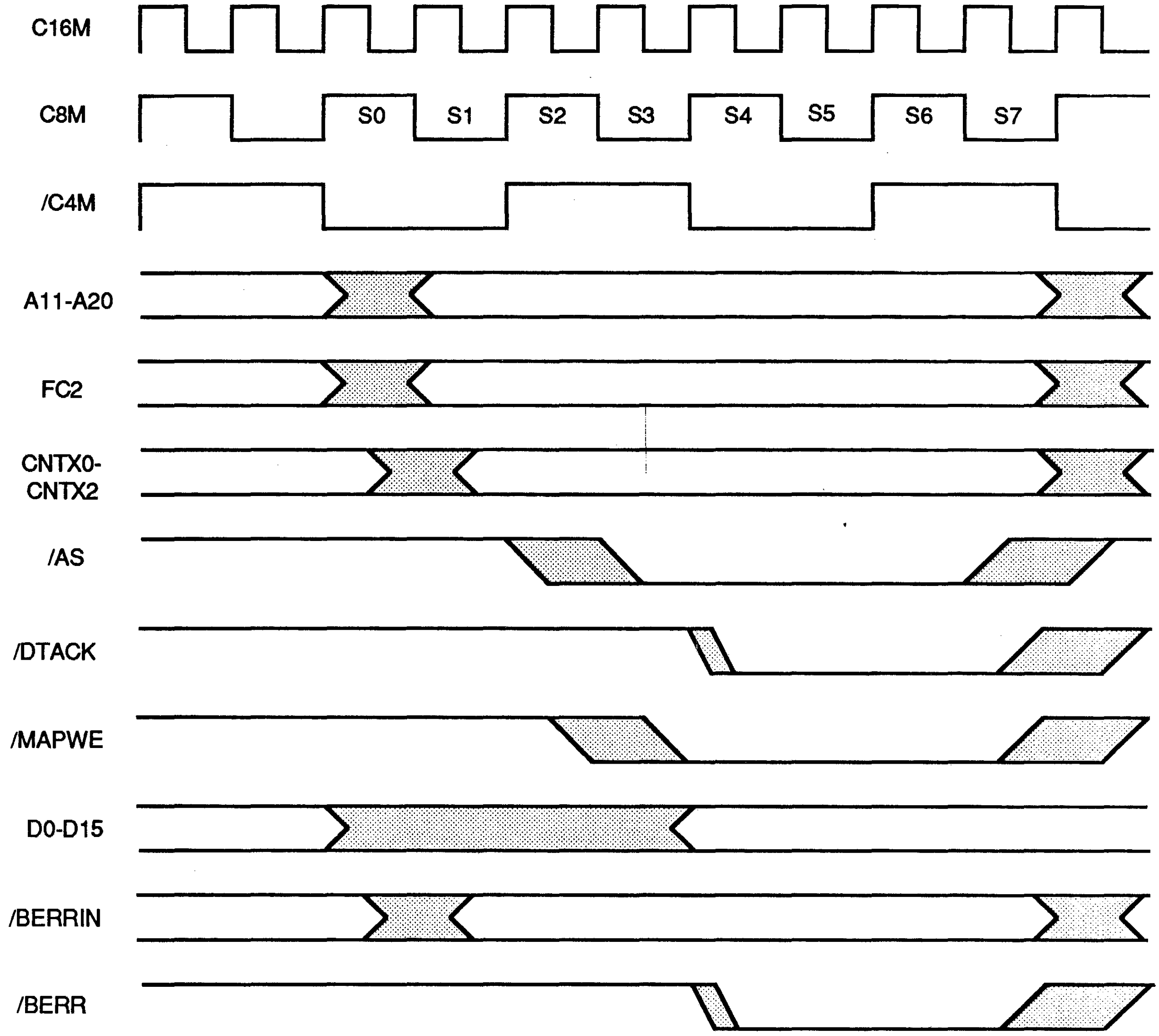


Figure 1.11 Memory Map Write cycle

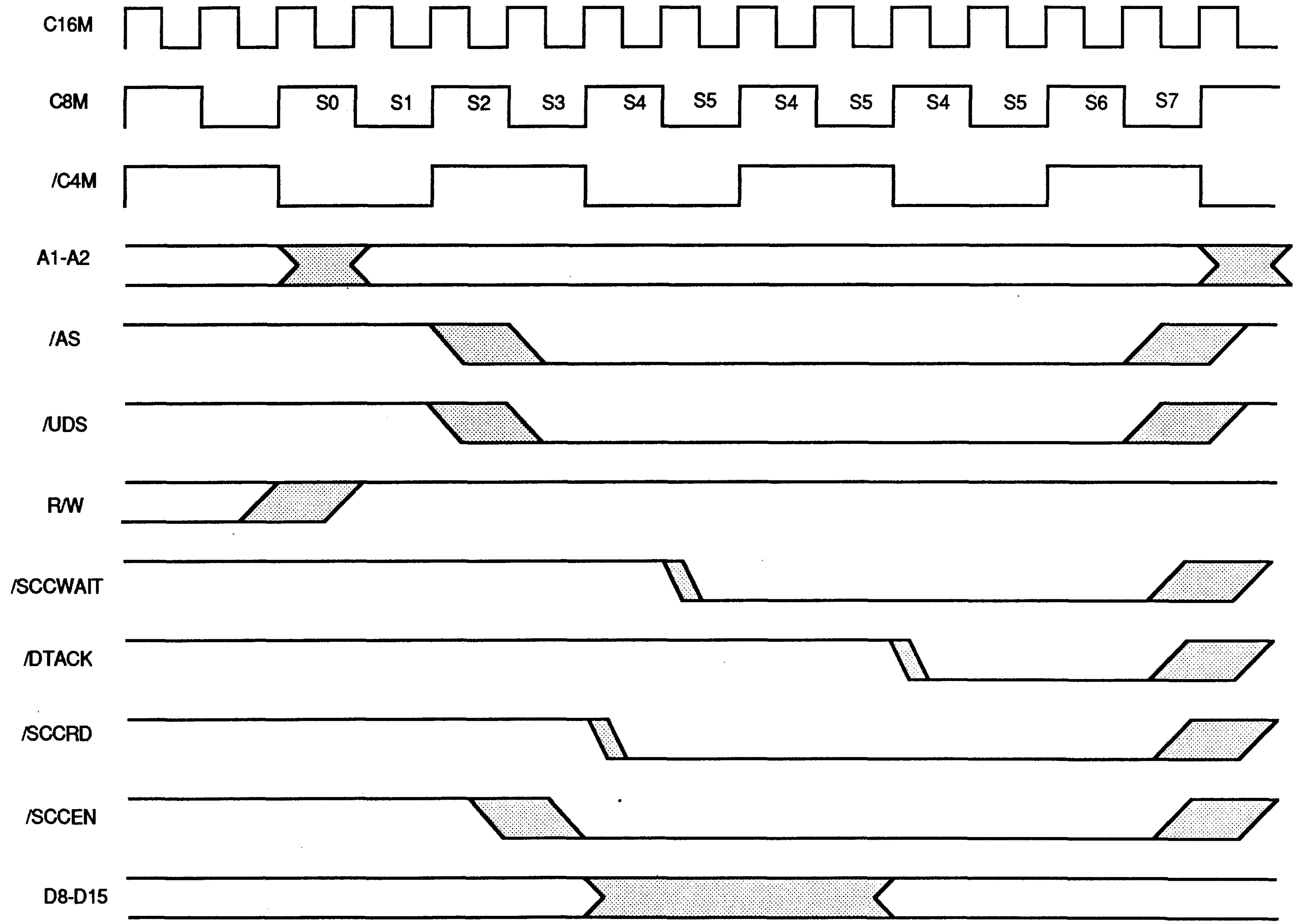


Figure 1.12 SCC read cycle

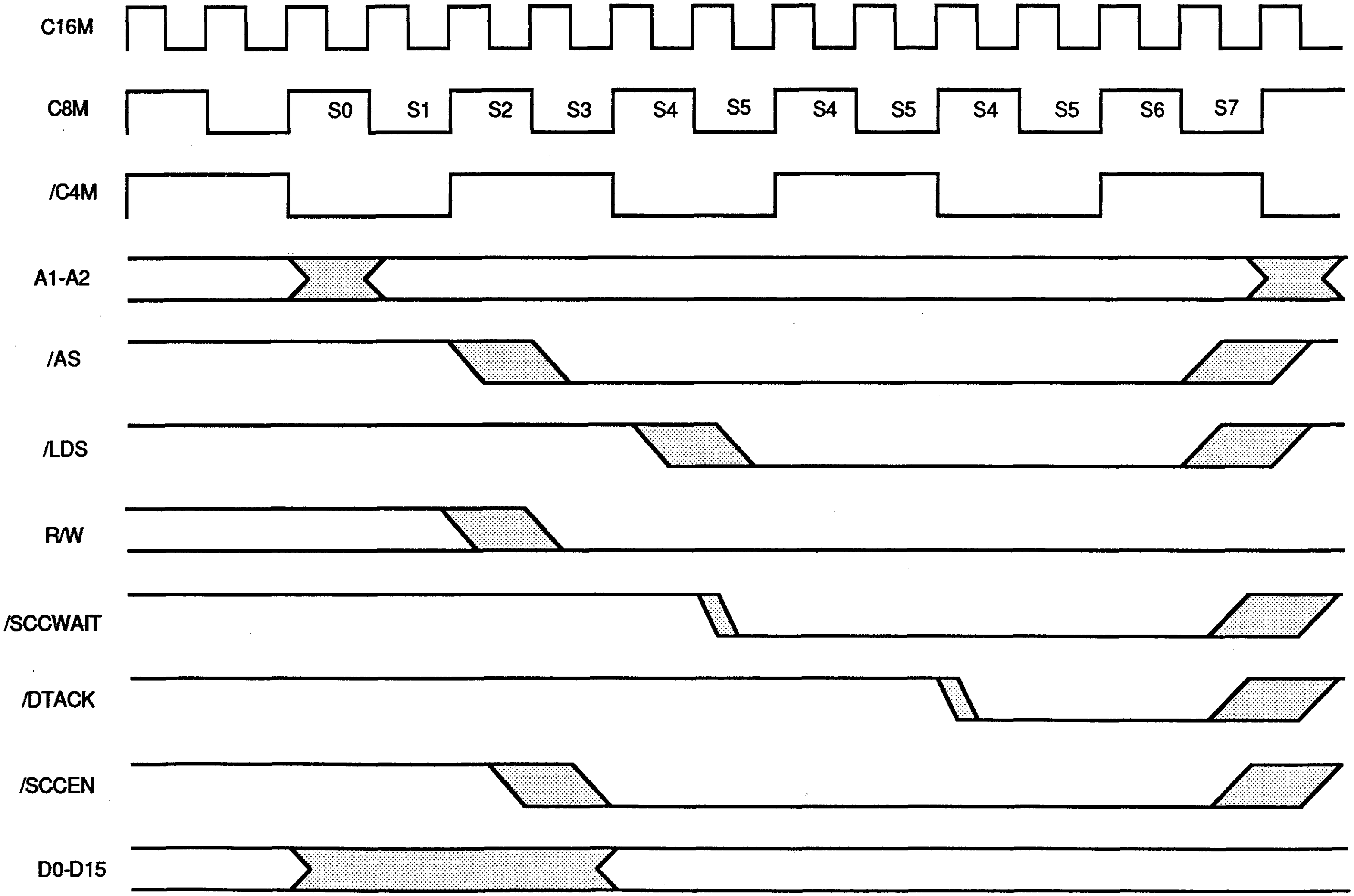


Figure 1.13 SCC write cycle

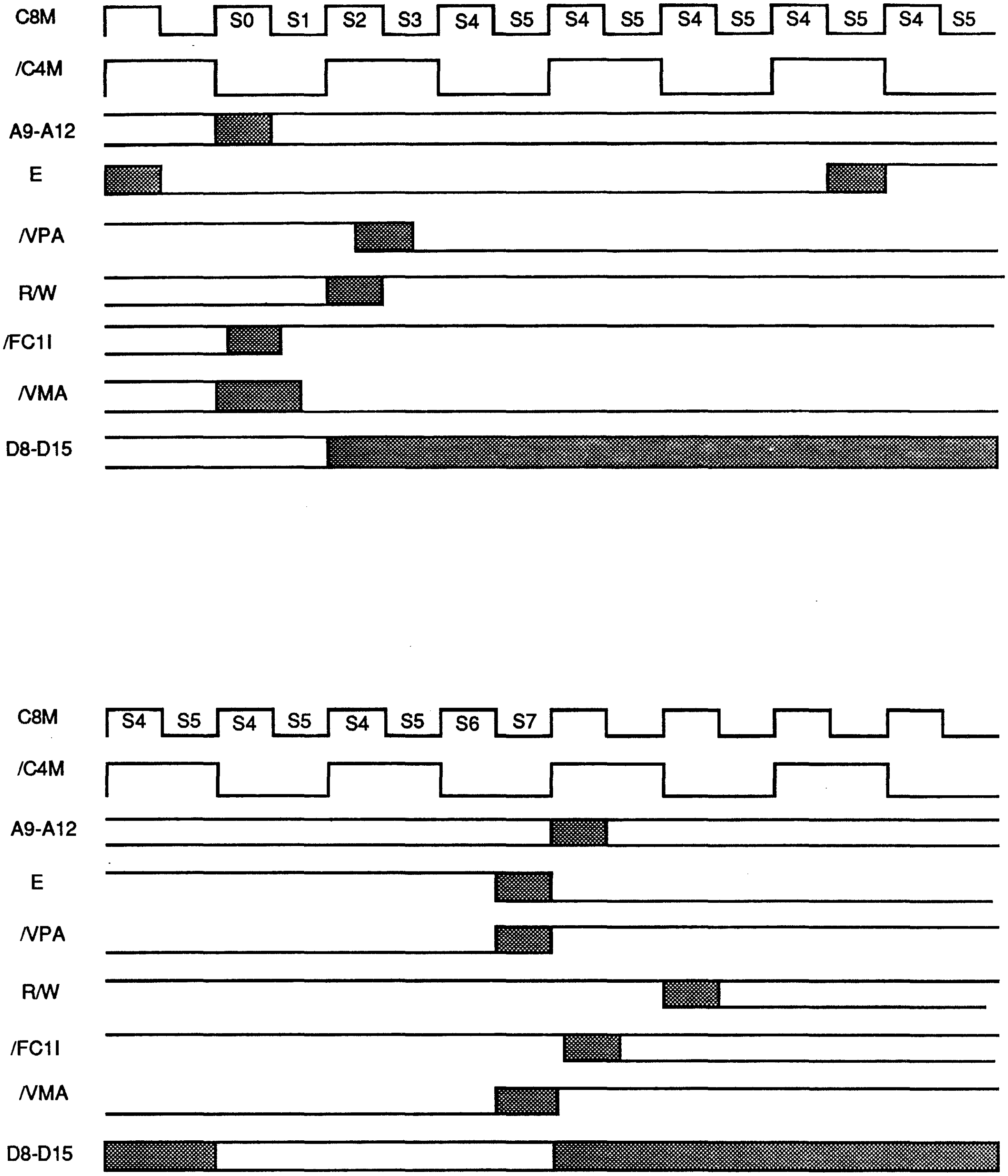


Figure 1.14 VIA read cycle (Best case)

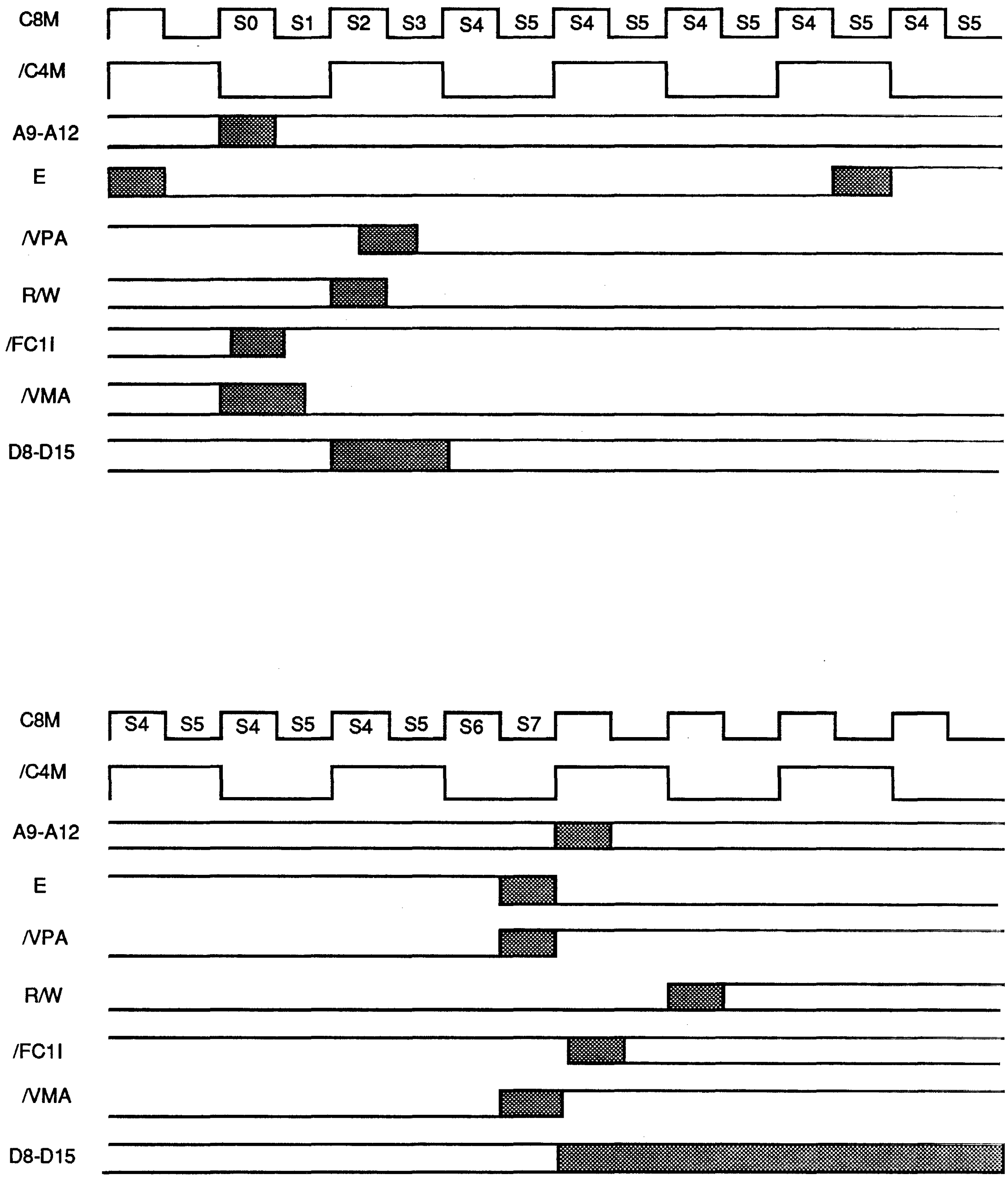


Figure 1.15 VIA write cycle (Best case)

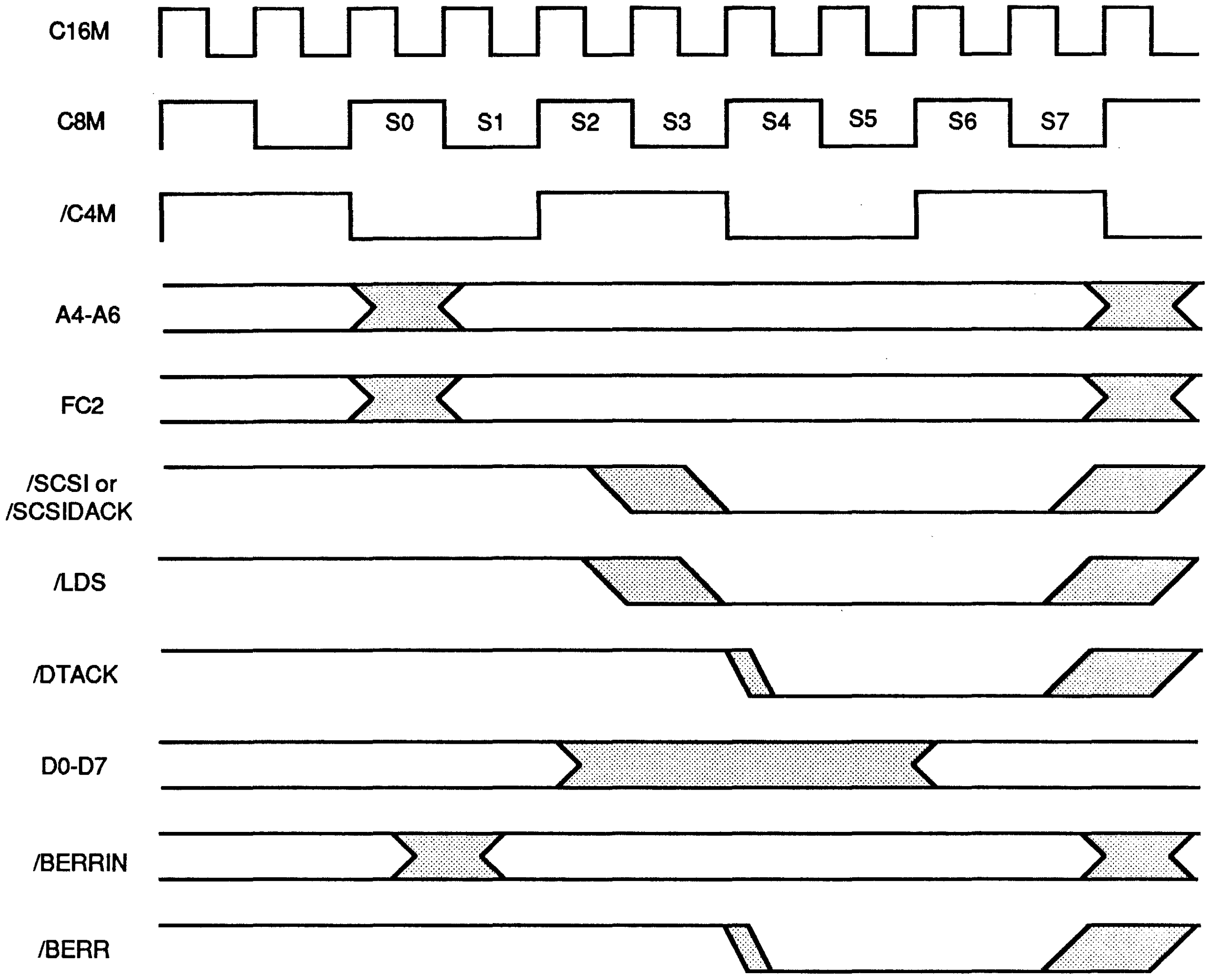


Figure 1.16 SCSI read cycle

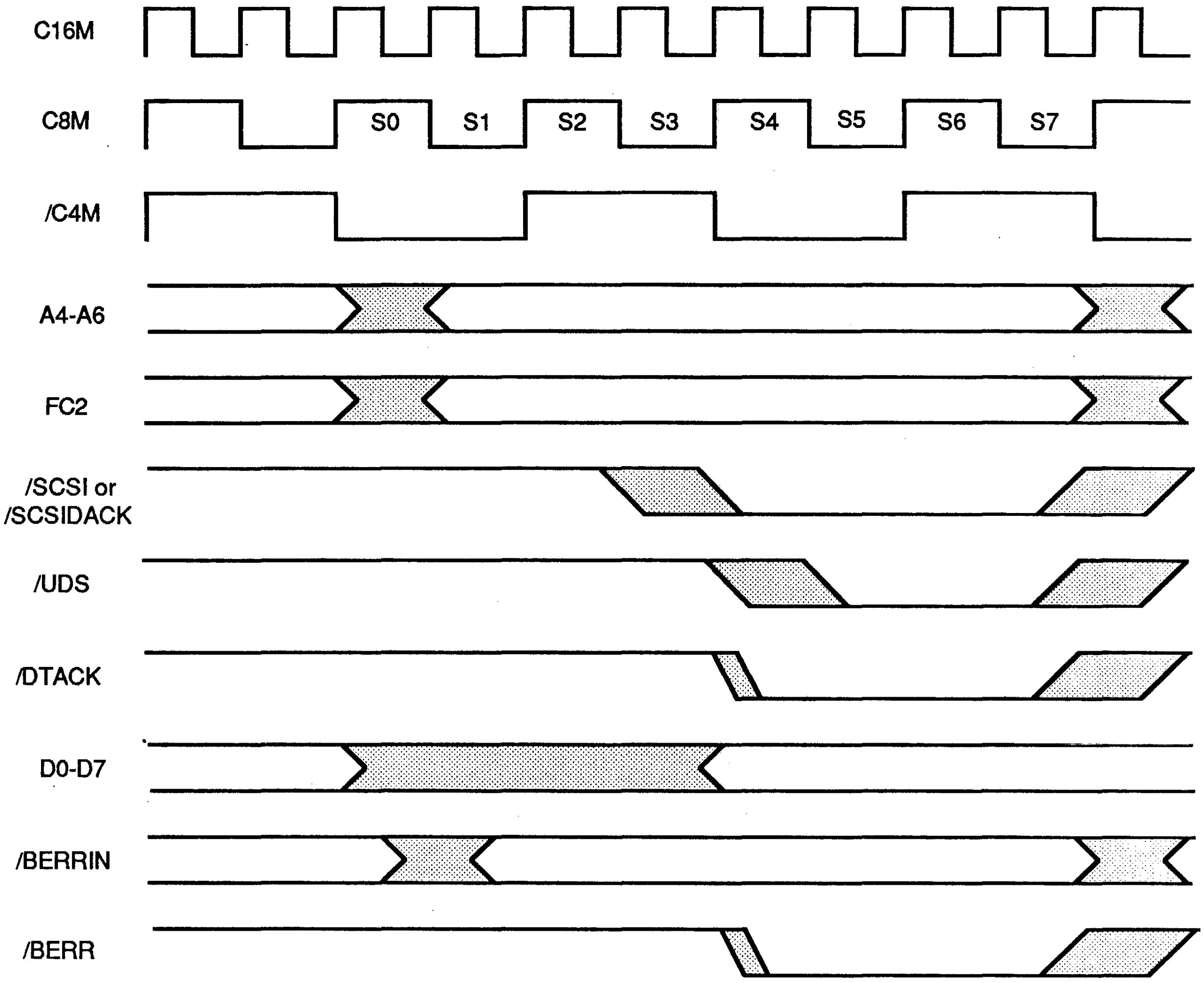


Figure 1.17 SCSI write cycle

## 1.7 IRQ PAL

The IRQ PAL encodes the interrupts to the 68000 and holds some of the control register bits.

The interrupts come from several sources and have different priorities. The IRQ PAL assures the interrupt with the highest prioity is the number encoded into /IPL0 - /IPL2 of the 68000. The interrupt numbers and thus priority levels are listed in Table 1.3.

The control register bits 8, 9, and 10 are stored in the IRQ PAL. Bit 8 low indicates the A port of the SCC should receive the /C3M to allow AppleTalk usage instead of a synchronous modem. Bit 9 high disables the PWRIRQ. The 10 bit low causes the test mode of the ECC generator to be activated.

## Table 1.3 Interrupts to the 68000

| Number | Source |
|---|---|
| 1 | **PWRIRQ.** The power switch is pressed. |
| 2 | **CMDCOMP.** The disk state machine has completed the command. |
| 3 | **SCSIRQ.** The SCSI chip has an interrupt. |
| 4 | **VIAIRQ.** The VIA chip has an interrupt. |
| 5 | **unused.** |
| 6 | **SCCIRQ.** The SCC chip has an interrupt. |
| 7 | **NMIRQ.** The Nonmaskible interrupt switch is pressed. |

## 2.0 Disk State Machine

The disk state machine consists of: the PALs Endcnt, SHIFT, BCOUNT, MISC, DSTATE, SYNC1 and SYNC2; a ten 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 FS 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 FS. The DMA control signals: DMA request (DMAREQ), DMA read/write (DMARD), DMA write pulse (DMAWP) 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 32 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. The data path for the read and write data streams is shown in Figures 2.1 and 2.2. When a match is found the data is read from the disk and the disk state machine posts completion.

The disk state machine is synchronized to the data clock used by the disk drive. The drive clock is digitally sampled and an internal clock is formed to prevent the loss of clocking or too high a frequency.

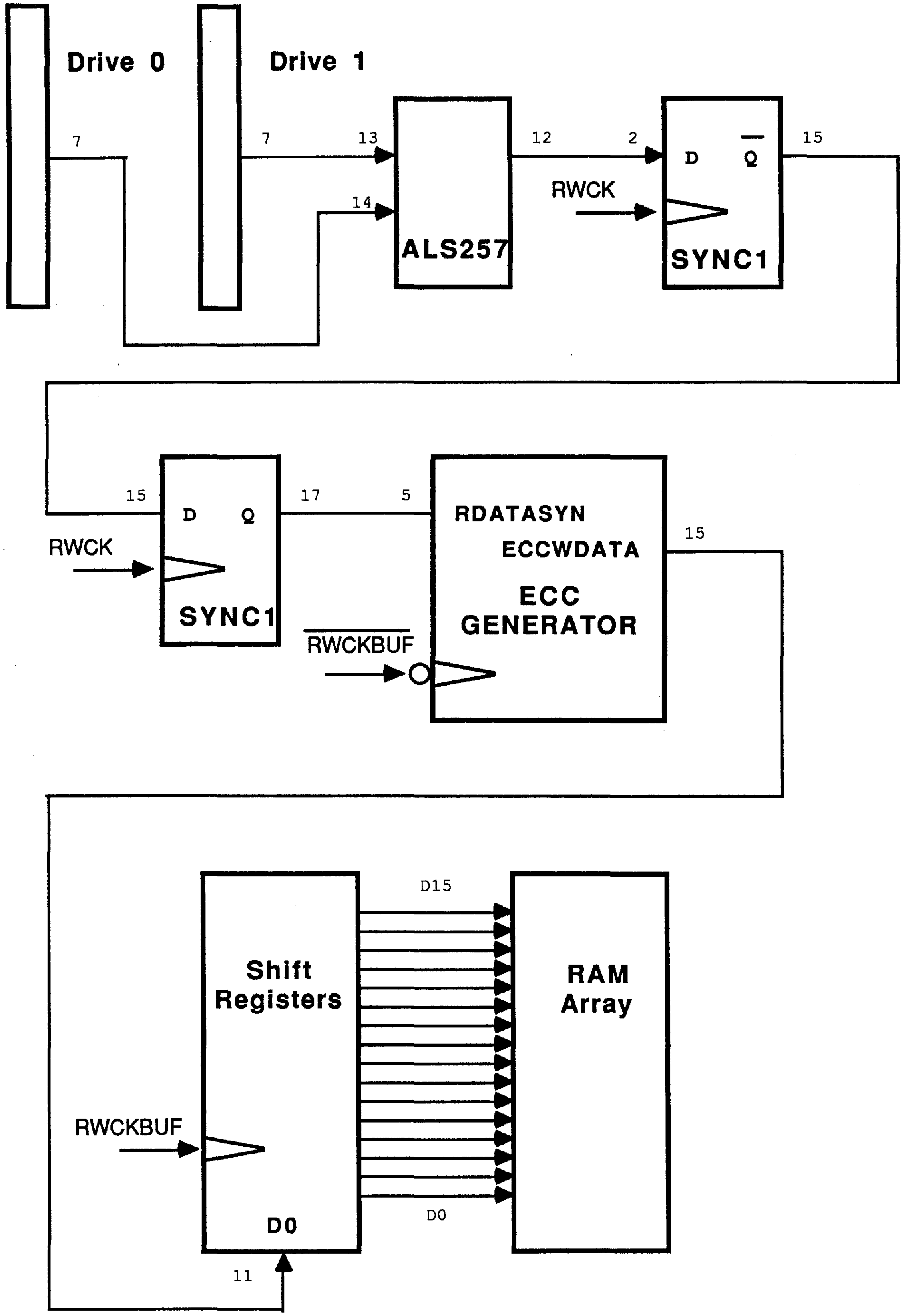
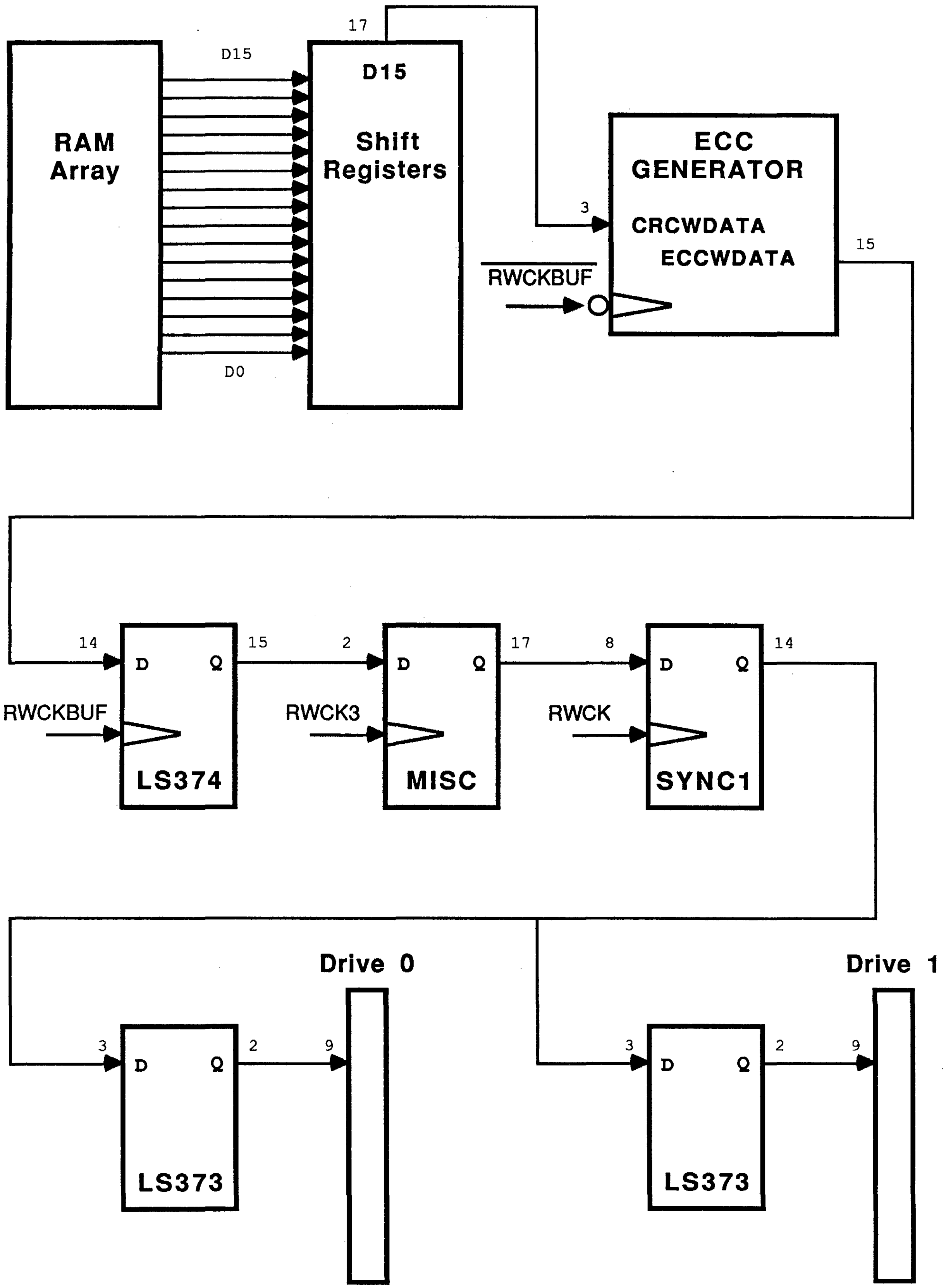


**Figure 2.1 Read data stream**



## Figure 2.2 Write data stream

## 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 | **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 start 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 the 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 shift register to perform the parallel load or dump of the register. |
| DA0-DA9 | **Direct Memory Address bits 0 through 9:** 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:** The negative edge is used to increment the DMA address counter. |
| MR | **Master Reset:** Used to clear the DMA address counter and the shift register. |
| CMD0-CMD1 | **Command register bits 0 and 1:** This holds the disk command currently in use. Format (00), Write (01), Read-ID (10), and Read (11) are the four possible commands. |

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

| Name | Definition |
|---|---|
| Y0-Y3 | **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 | **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. This generates an interrupt to the 68000. |

## 2.1 Control Register

The Control Register consists of a 74ALS374 octal register and 2 bits located in the IRQ PAL.

The 68000 controls the disk state machine through the Control Register and the /START line from the VIA. Two bits (D6, D7) of the Control Register encode the four command sequences the disk state machine can execute: Format, Read-ID, Read, and Write. The DECODE2 PAL decodes the address space for the latch as $D8 0000 - $D7 FFFF. Table 2.2 shows the assignment of bits in the Control Register. The Control Register also controls the state of the red/green Light Emitting Diode (LED) which is used to indicate the status of the FS.

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 Control Register**

| Bit # | Description |
|---|---|
| 0 | **CNTX0IN:** The low order bit of the context number used during user state of the 68000. |
| 1 | **CNTX1IN:** The 2nd bit of the context number used during user state of the 68000. |
| 2 | **CNTX2IN:** The 3rd bit of the context number used during user state of the 68000. |
| 3 | **LEDGREEN:** Turns on th green Light Emitting Diode. If the red LED is on yellow is produced. |
| 4 | **LEDRED:** Turns on the red Light Emitting Diode. If the green LED is on yellow is produced. |
| 5 | **DRIVE0:** Selects which drive is actively connected to the disk state machine. Drive 0 is high and drive 1 is low. |
| 6 | **CMD0:** The low order bit of the disk command. |
| 7 | **CMD1:** The high order bit of the disk command. |
| 8 | **/APTKA:** The selection bit of the connection of clock sources to the A port of the SCC. |
| 9 | **/PWRIRQEN:** When high the bit disables the PWRIRQ. |
| 10 | **/ECCTESTL:** Activates the test mode of the ECC chip. |

## 2.2 MISC PAL

The MISC PAL generates the 9th and 10th bits of the DMA address counter and synchronizes the write data bit stream. The 10th bit (DA9) is used to suppress the refresh cycles of state 3.

## 2.3 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 2.3 through 2.6. 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. The disk state machine powers up in this state. |
| 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 trying 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, 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. |

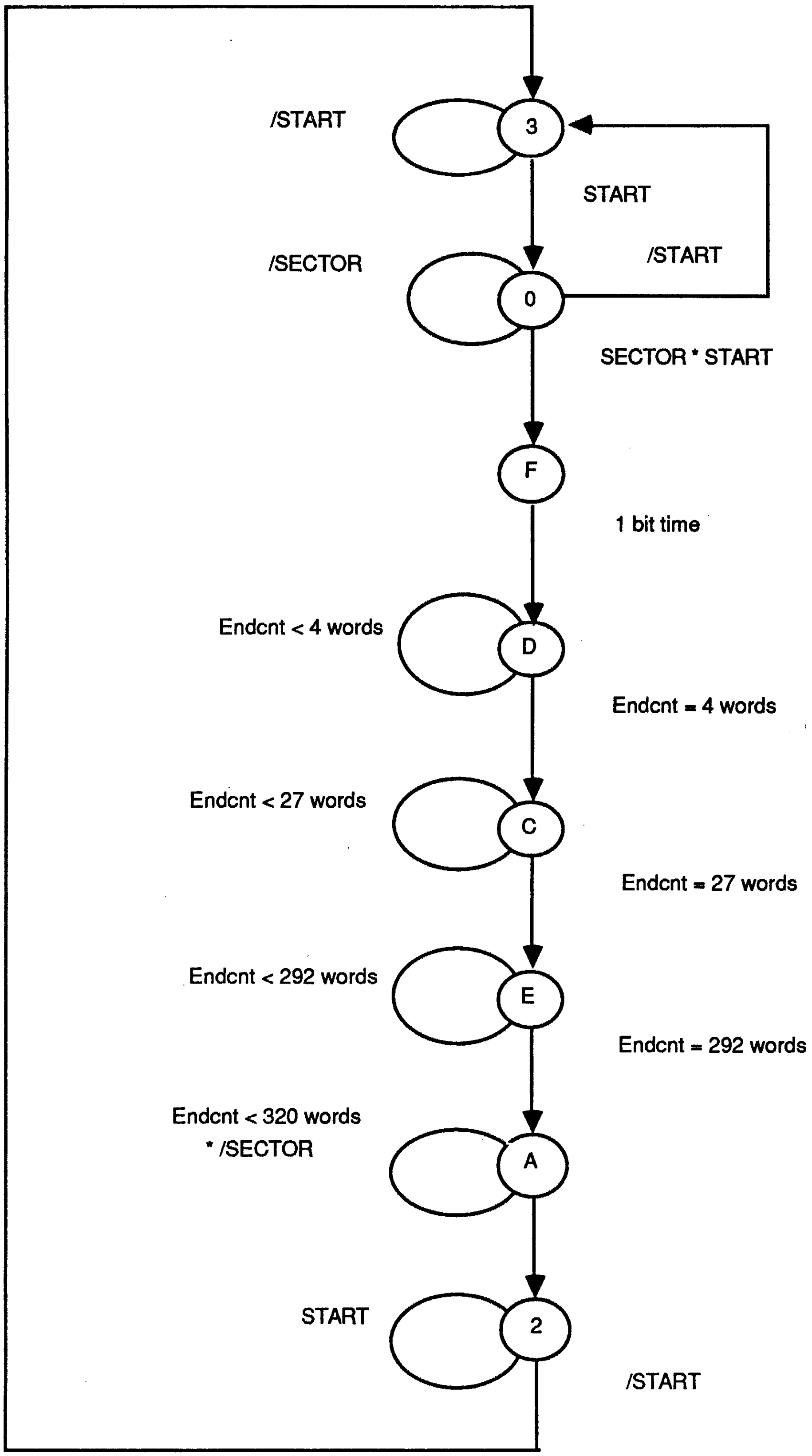


Figure 2.3 Format state sequence

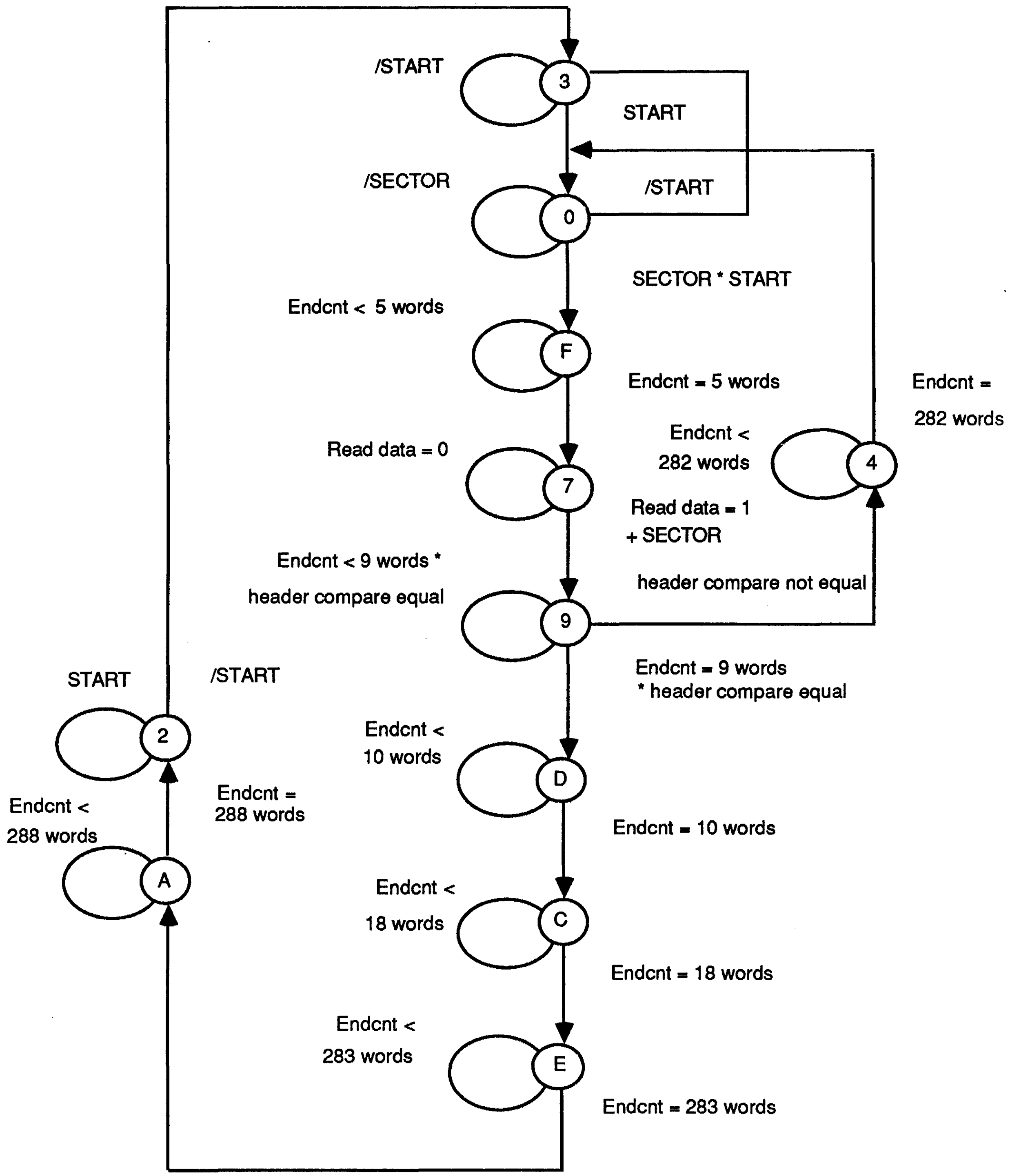


Figure 2.4 Write state sequence

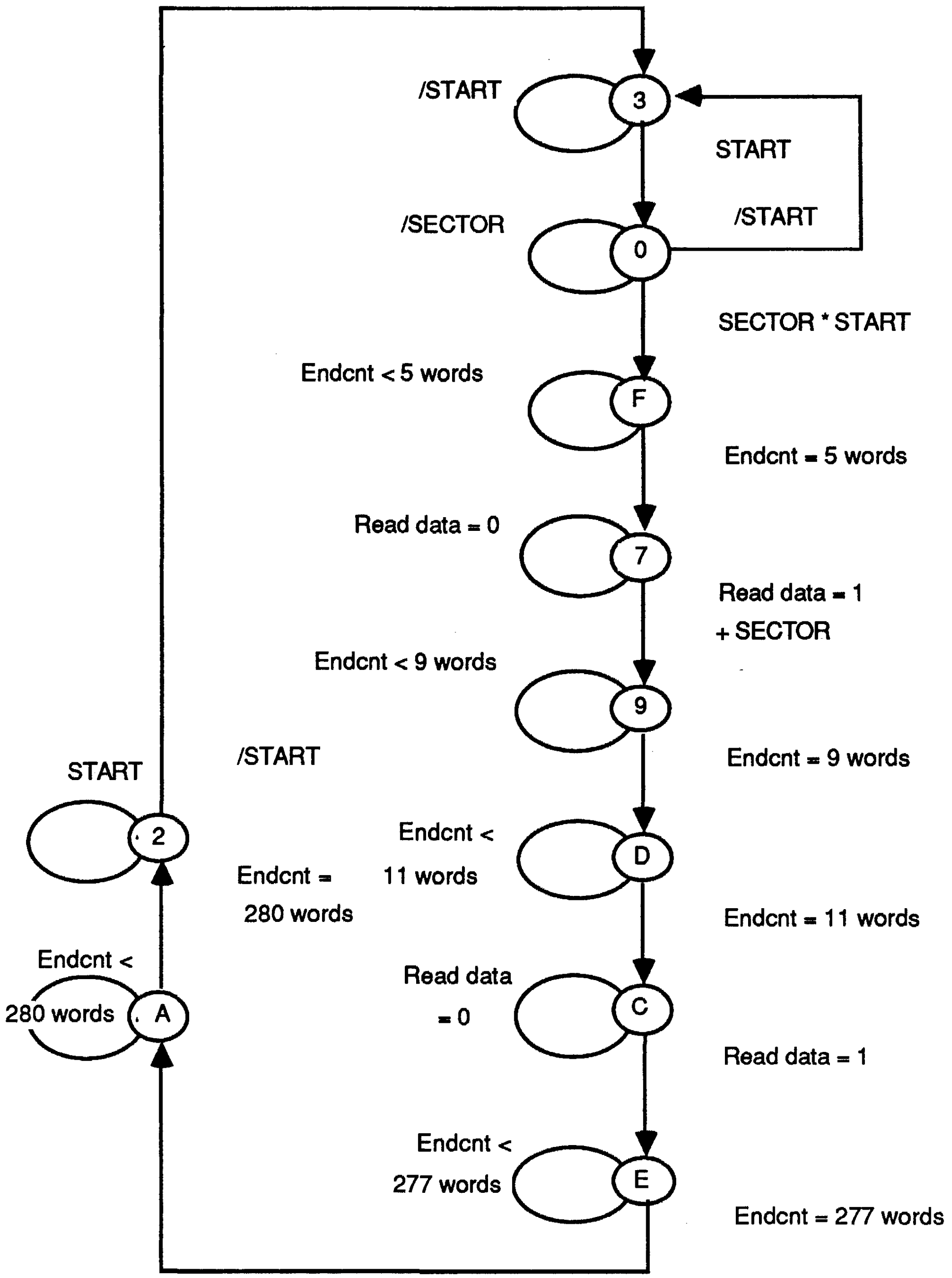


Figure 2.5 Read-ID state sequence

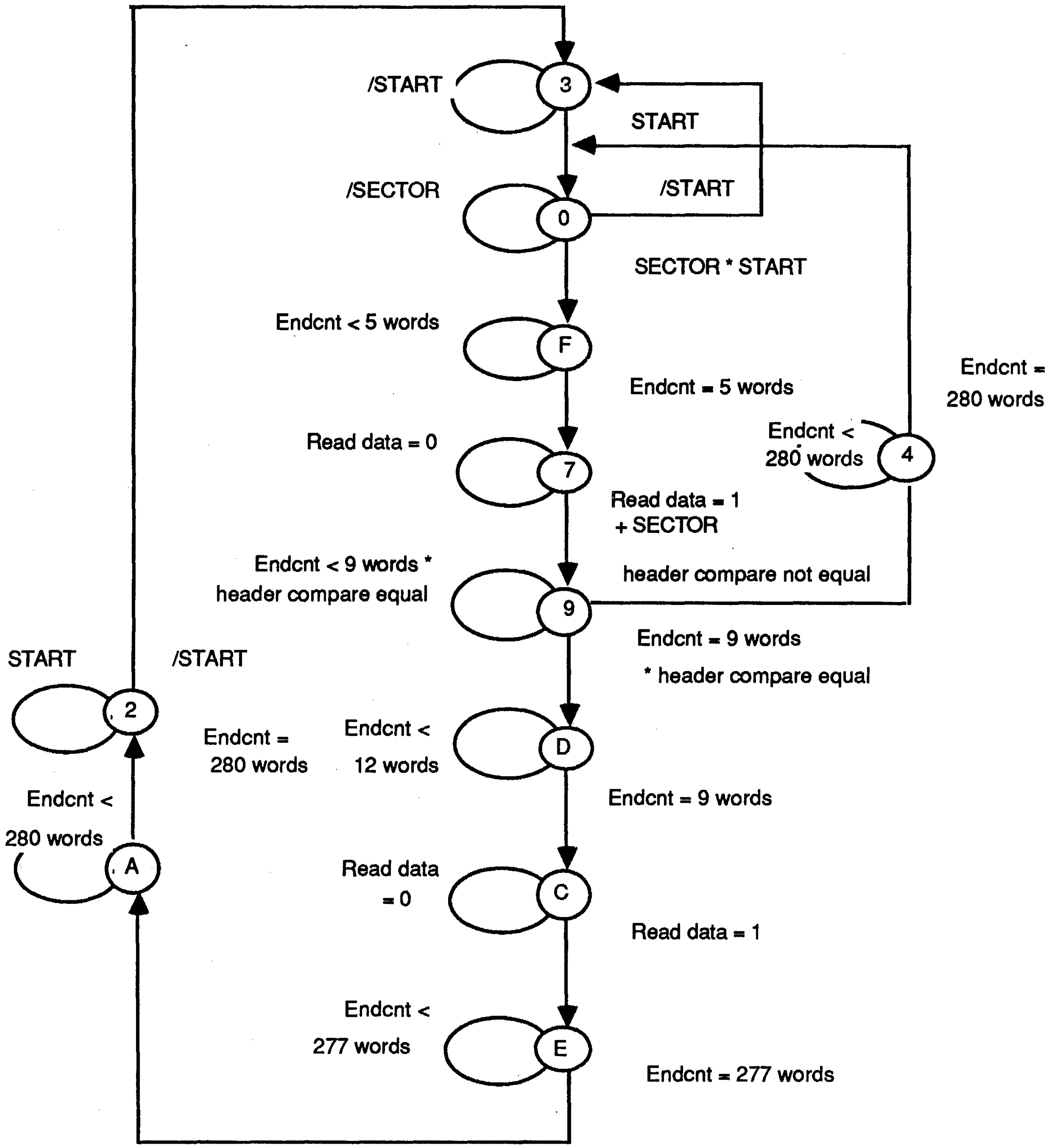


Figure 2.6 Read state sequence

## 2.4 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 RWCKBUF 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 RWCKBUF 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.

## 2.5 BCOUNT PAL

The BCOUNT PAL is basically a 4 bit binary up counter. It counts out the 16 bit times of the disk RWCKBUF clock for each RAM word to be shifted to or from the disk. Normally it is free running but it 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 2.7 shows a DMA read from RAM and Figure 2.8 shows a DMA write to RAM. The DMA request line (DMAREQ) is asserted eight 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.

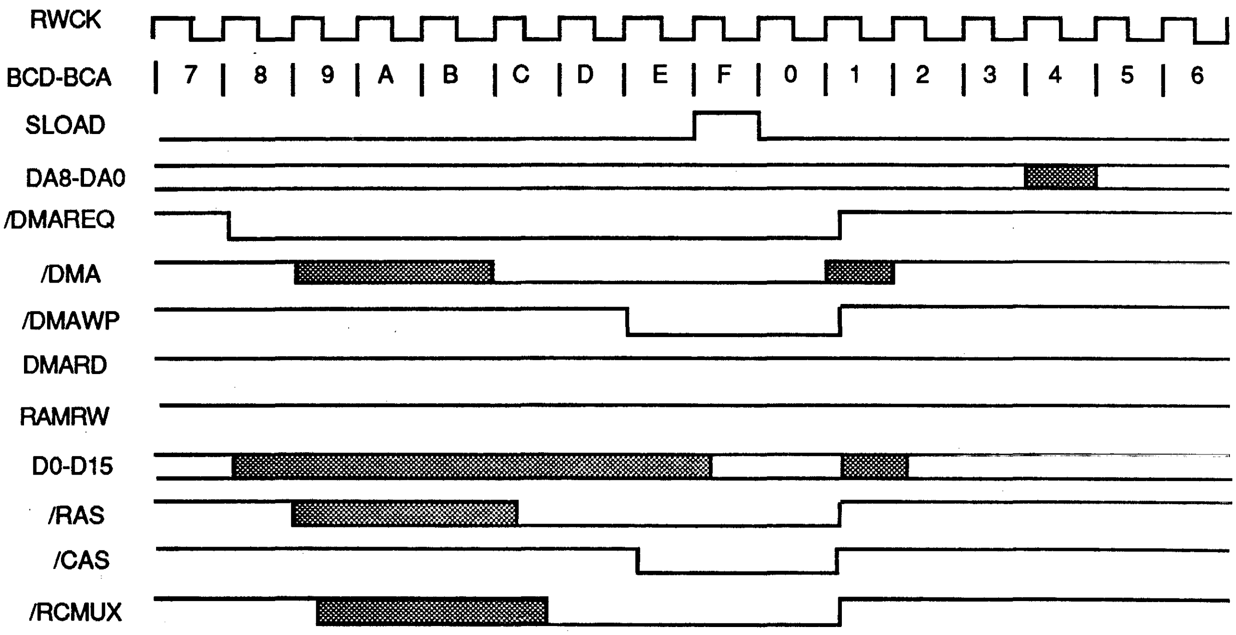


Figure 2.7 DMA read cycle

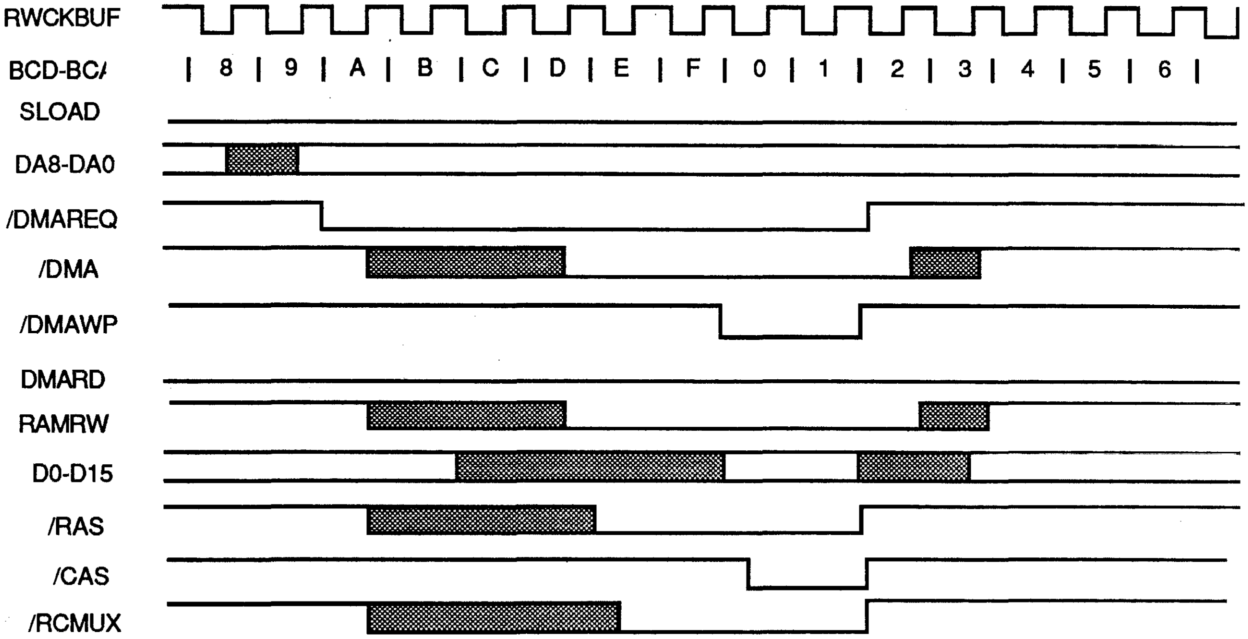


Figure 2.8 DMA write cycle

## 2.6 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.

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.

## 2.7 SYNC1 PAL

The SYNC1 PAL is a synchronizer for the read and write data streams and The generator of the exclusive-or for header compares. The synchronizer is the last clocked point in the data steam and uses the disk drive RWCK. It also inserts a one in the read data stream during the rising edge of the sector pulse. This prevents the disk state machine from getting stuck if no data is coming from the drive.

## 2.8 SYNC2 PAL

The SYNC2 PAL is a digital clock sampler which samples the RWCK from the disk drive and produces RWCKBUF and C16M. C16M is just a simple divide by two of the C32M clock from the oscillator module. The digital sampling is done to prevent the RWCKBUF from going too fast or too slow.

The RWCK can be either 5 MHz or 7.5 MHz and the RWCKBUF will track with some phase delay. Figure 2.9 shows the normal tracking of a 7.5 MHz RWCK. Figure 2.10 shows how the RWCKBUF is continued even if RWCK goes away.

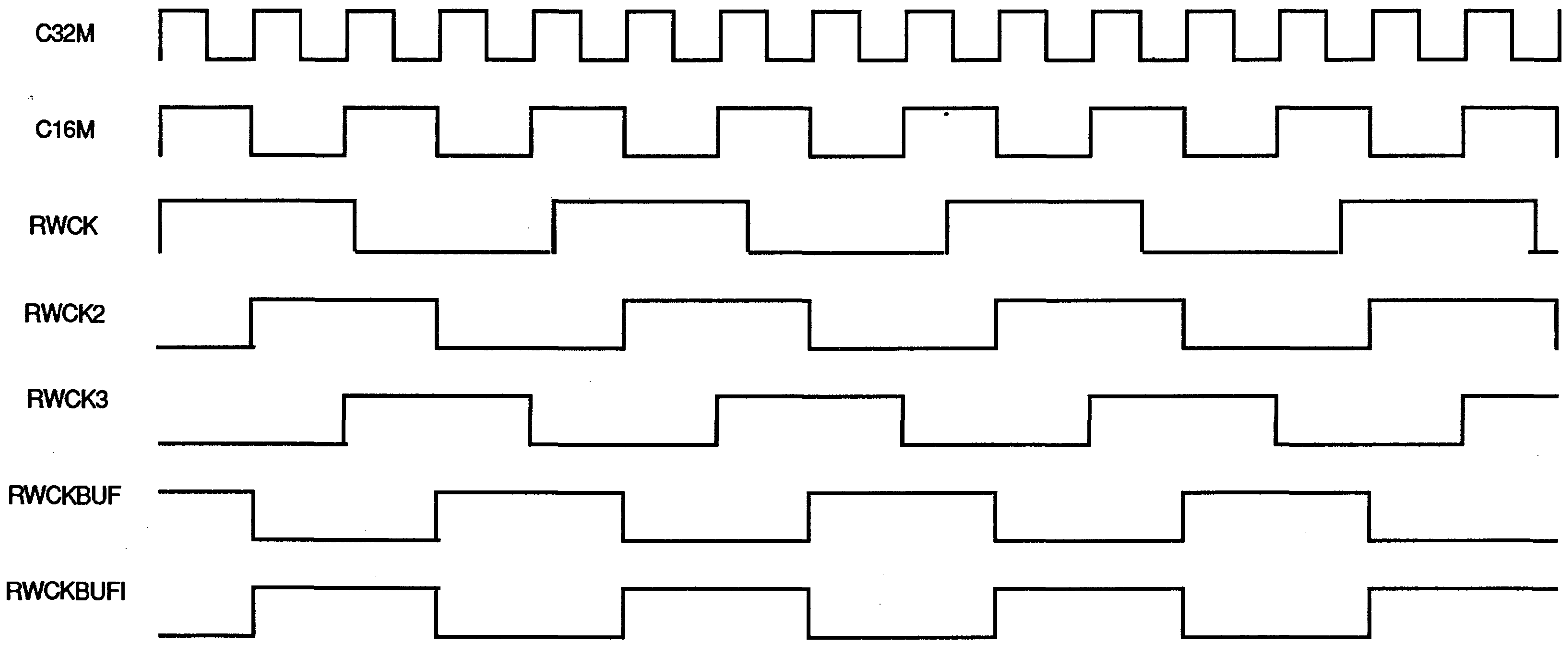


Figure 2.9 Normal RWCKBUF (7.5 MHz RWCK)

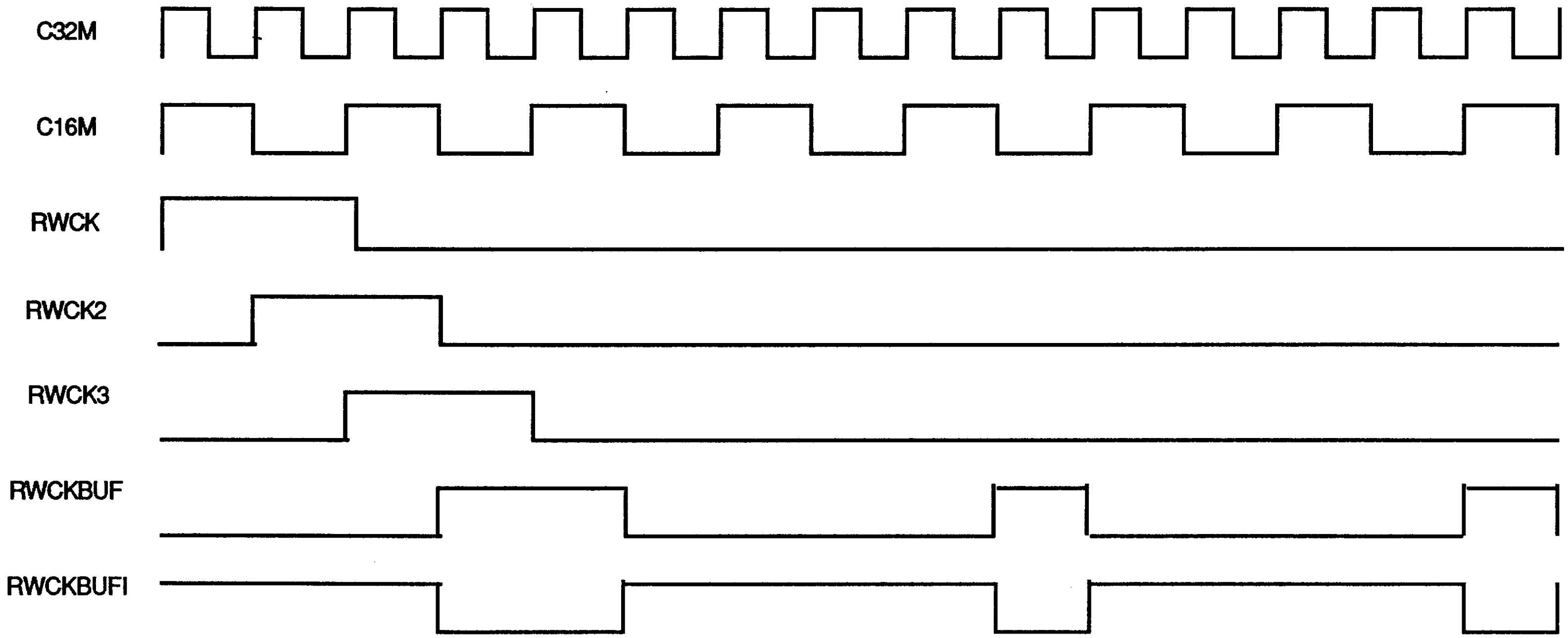


Figure 2.10 Fixed RWCKBUF

## 2.9 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 allows 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 bit 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.

The ECC chip can be put in test mode by asserting the /ECCTESTL signal in the Control Register.

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.

## 2.10 Disk RAM Buffer

The disk state machine uses the top 1 K bytes of RAM as a sector buffer. This is the address space $0F FC00 - $0F FFFF in physical RAM as DMA accesses are not mapped. 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 in the Control Register for the appropriate command.

4. Assert /START.

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 and/or the Control Register.

5. If two or more index pulses occur without /CMDCOMP being asserted 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.

## 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. |
| **ECC0-ECC5** | The ECC syndrome bytes. They are the 48 bit syndrome of the zero byte, 532 data 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 an 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 | D0 D1 | D0 D1 |
| C | 12 | 0 | 0 | D2 D3 | D2 D3 |
| D | 13 | 00 01 | 0 | D4 D5 | D4 D5 |
| E | 14 | 00 T1 | 0 | D6 D7 | D6 D7 |
| F | 15 | T0 HS | 0 | D8 D9 | D8 D9 |
| 10 | 16 | /T1 /T0 | 0 | D10 D11 | D10 D11 |
| 11 | 17 | /HS 00 | 01 00 | D12 D13 | D12 D13 |
| 12 | 18 | 0 | D0 D1 | D14 D15 | D14 D15 |
| 13 | 19 | 0 | D2 D3 | D16 D17 | D16 D17 |
| 14 | 20 | 0 | D4 D5 | D18 D19 | D18 D19 |
| 15 | 21 | 0 | D6 D7 | D20 D21 | D20 D21 |
| 16 | 22 | 0 | D8 D9 | D22 D23 | D22 D23 |
| 17 | 23 | 0 | D10 D11 | D24 D25 | D24 D25 |
| 18 | 24 | 0 | D12 D13 | D26 D27 | D26 D27 |
| 19 | 25 | 0 | D14 D15 | D28 D29 | D28 D29 |
| 1A | 26 | 01 00 | D16 D17 | D30 D31 | D30 D31 |
| 1B | 27 | D0 D1 | D18 D19 | D32 D33 | D32 D33 |
| 1C | 28 | D2 D3 | D20 D21 | D34 D35 | D34 D35 |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| 113 | 275 | D496 D495 | D514 D515 | D528 D529 | D528 D529 |
| 114 | 276 | D498 D497 | D516 D517 | D530 D531 | D530 D531 |
| 115 | 277 | D500 D501 | D518 D519 | ECC0 ECC1 | ECC0 ECC1 |
| 116 | 278 | D502 D503 | D520 D521 | ECC2 ECC3 | ECC2 ECC3 |
| 117 | 279 | D504 D505 | D522 D523 | ECC4 ECC5 | ECC4 ECC5 |
| 118 | 280 | D506 D507 | D524 D525 | | |
| 119 | 281 | D508 D509 | D526 D527 | | |
| 11A | 282 | D510 D511 | D528 D529 | | |
| 11B | 283 | D512 D513 | D530 D531 | | |
| 11C | 284 | D514 D515 | ECC0 ECC1 | | |
| 11D | 285 | D516 D517 | ECC2 ECC3 | | |
| 11E | 286 | D518 D519 | ECC4 ECC5 | | |
| 11F | 287 | D520 D521 | 0 | | |
| 120 | 288 | D522 D523 | 0 | | |
| 121 | 289 | D524 D525 | 0 | | |
| 122 | 290 | D526 D527 | | | |
| 123 | 291 | D528 D529 | | | |
| 124 | 292 | D530 D531 | | | |

## Table 2.5 Disk RAM Buffer (continued)

| MEMORY OFFSET<br>HEX | WORD ADDRESS<br>DECIMAL | FORMAT (00) | WRITE (01) | READ-ID (10) | READ (11) |
|---|---|---|---|---|---|
| 125 | 293 | ECC0 ECC1 | | | |
| 126 | 294 | ECC2 ECC3 | | | |
| 127 | 295 | ECC4 ECC5 | | | |
| 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 | | | |

## 3.0 Disk Servo / Clock Interface

The Disk Servo / Clock Interface consists of: a 65C22 Versatile Interface Adapter (VIA), a Real time Clock chip (RTC), the SERIAL PAL, two 74ALS373s, two 74ALS257 and a 74LS32. The primary function of the interface is to move the head on the 'Nisha' disk and provide the time of day. The VIA has two 8 bit parallel ports, an 8 bit bidirectional shift register, and some handshake lines. Most of the control lines are connected to the 26 pin 'Nisha' interface. The remaining pins control the mapping of ROM in the address space (OVERLAY), start the disk state machine (/START), or power down the FS (POWER). The SERIAL PAL is used to share the 8 bit shift register of the VIA between the disk servo and the RTC. The SERIAL PAL also provides the start and stop bits of an asynchronous interface to the disk servo. The clock chip contains the time of day, date, and 64 bytes of RAM. A 10 year battery powers the RTC when the AC power is disconnected.

The interface supports two drives by multiplexing the control signals. The signal DRIVE from the Control Register selects which drive is active at any time. This signal must not be changed when /START is asserted. The 74ALS373s hold the state of the disk control lines for one drive while the other is used. Thus whenever the drive is changed the control signals should be is a safe state. When two drives are present the servo for one can be given a command and then the drive switched and another command given. Thus the seek time of one drive can be overlapped with the accessing of the other.

### 3.1 Versatile Interface Adapter (65C22)

The Versatile Interface Adapter is a 40 pin DIP implemented in CMOS technology. It contains two 8 bit parallel ports, two 16 bit binary counters/timers, 4 handshake lines, and an 8 bit shift register. All of these features are used in the FS.

The VIA occupies the address space from $E8 0000 to $EF FFFF. Access to the VIA should be done with $E8 0000 as the base address and the byte offsets listed in Table 3.1. The 68000 executes a VPA cycle accessing the VIA which makes the access take up to 2 us.

The VIA has some unique characteristics. When power-on or reset occurs all the PB0-PB7 and PA0-PA7 pins are programmed as inputs and internal pull-ups provide a one output. Thus the VIA is used to control power-on conditions. Also, an implementation bug makes the address lines RS0-RS3 sensitive to changes during the phase 2 clock edges. A 74LS32 is used to keep these lines high when the chip is not being accessed. The phase 2 clock is the E clock of the 68000 CPU. It is a 783.36 KHz clock with a 60 / 40 duty cycle.

To send a byte to the disk servo the following procedure is used:

1. Set the bit rate. The bit rate is set to 57.6 K baud with T1C-L = $05 and T1C-H = $00.

2. Write the byte to be sent into SR.

3. Assert the /STBIT and SERVO/CLK lines.

4. Wait for completion by setting interrupt or polling IFR bit 2 for one.

5. Wait for two time outs of T1 by setting interrupt or polling IFR bit 6.

6. Deassert /STBIT line.

To send a byte to the clock chip the following procedure is used:

1. Set the bit rate. Since the clock chip can run at the full 783.36 KHz of phase 2 the timer 1 count is not needed. Thus ACR = $F8.

2. Disable SERVO/CLK and /STBIT lines.

3. Write the byte to SR.

4. Wait for completion by setting interrupt or polling IFR bit 2 for one.

To read a byte from the clock chip the following procedure is used:

1. Set the bit rate. Since the clock chip can run at the full 783.36 KHz of phase 2 the timer 1 count is not needed. Enable the SR for shift in under control of phase 2.

2. Disable SERVO/CLK and /STBIT lines.

3. Read a byte from SR. Throw it away.

4. Wait for completion by setting interrupt or polling IFR bit 2 for one.

5. Disable the shift register.

6. Read the byte in SR.

## Table 3.1 VIA registers

| Name | Offset Address | Comments |
|---|---|---|
| **ORB/IRB** | **$0000** | **Output Register 'B' / Input Register 'B'.** This is the 8 bit parallel port PB0-PB7. When written this register sets the value for those bits which are programmed as an output. When read this register returns the value programmed for the output pins and the value at the pin for those programmed as inputs. |
| **ORA/IRA** | **$1E00** | **Output Register 'A' / Input Register 'A'.** This is the 8 bit parallel port PA0-PA7. When written this register sets the value for those bits which are programmed as an output. When read this register returns the value at the pin. |
| **DDRB** | **$0400** | **Data Direction Register 'B'.** This register programs the pins of PB0-PB7 as input or output. A one indicates an output, a zero indicates an input. Normally this is set to $BB. |
| **DDRA** | **$0600** | **Data Direction Register 'B'.** This register programs the pins of PB0-PB7 as input or output. A one indicates an output, a zero indicates an input. Normally this is set to $5E. |
| **T1C-L** | **$0800** | **T1 Low-Order Latches / Counter.** This counter is used to time the bit clock for the serial bit stream to the RTC and disk servo. This counter is decremented at 783.36 KHz. When the count is zero the latch reloads the counter and PB7 (OCLK) is toggled. |
| **T1C-H** | **$0A00** | **T1 High-Order Counter.** Writing this register loads the latch and transfers the high and low latches into the counter. |
| **T1L-L** | **$0C00** | **T1 Low-Order Latches.** Writing this register loads the low order latch. |
| **T1H-L** | **$0E00** | **T1 High-Order Latches.** Writing this register loads the high order latch. |
| **T2C-L** | **$01000** | **T2 Low-Order Latches / Counter.** This counter is normally used to simulate the Vertical Blanking (VBL) interrupt of Macintosh. It counts the INDEX pulse on PB6 which has a 45.817 Hz frequency on 'Nisha'. |
| **T2C-H** | **$1200** | **T2 High-Order Counter.** Writing this register loads the latch and transfers the high and low latches into the counter. |

## Table 3.1 VIA registers (continued)

| Name | Offset Address | Comments |
|---|---|---|
| SR | $1400 | **Shift Register.** This is the 8 bit synchronous shift register. This register contains the byte to be written out when the shift register sends data and the byte sent when the shift register receives serial data. |
| ACR | $1600 | **Auxiliary Control Register.** This register controls what modes the timers and shift register operate under. |
| PCR | $1800 | **Peripheral Control Register.** This register control how the handshake lines CA1, CA2, CB1, CB2 respond to edges on the pins. |
| IFR | $1A00 | **Interrupt Flag Register.** This register indicates which of the interrupt sources is interrupting. |
| IER | $1C00 | **Interrupt Enable Register.** This is a mask for the interrupts in the IFR. If a bit is one the appropriate interrupts is allowed to cause an interrupt to the 68000 CPU. |

# Table 3.2 VIA signals

| VIA Name | FS Name | Input/Output | Description |
|---|---|---|---|
| CA1 | /SIORDY | I | **Serial I/O Ready.** This signal is asserted by the selected drive to indicate completion of the command given. This input to the VIA is edge triggered and can be used to generate an interrupt. |
| CA2 | ONESEC | I | **One Second Interrupt.** This active low signal is generated by the clock chip once a second. Thus the negative edge is significant. |
| PA0 | /ECCERR | I | **Error Correction Code Error.** This active low signal indicates the last sector read from the disk had a ECC error. This signal is valid while /CMDCOMP is asserted. |
| PA1 | /PC | O | **Precompensation.** This active low signal is sent to the 'Nisha' write electronics to change the amount of write current and the phase of the Write signal. Tracks 257 and higher on the drive are written with precompensation. |
| PA2 | /SIORDY | I | **Serial I/O Ready.** This signal is asserted by the selected drive to indicate completion of the command given. This input to the VIA is level sensitive and can be used to poll for completion if a drive was deselected. |
| PA3 | /START | O | **Start disk state machine command.** This active low signal starts the disk state machine executing the command contained in the disk command latch. |
| PA4 | OVERLAY | O | **Overlay the ROM at the bottom of memory.** This active high signal causes the ROM to begin at $00 0000 in addition to $40 0000. This overlays the RAM normally present at this address so the power-on reset vector at $00 0000 will be present in ROM. |
| PA5 | SERVORDY | I | **Servo ready.** This active high signal indicates the disk servo of the drive selected is ready. This normally indicates the servo is positioned over a track and is ready to read or write. It will be low at power-on before a servo reset and while the head is seeking. |
| PA6 | /SERVORST | O | **Servo reset.** This active low signal is used to reset the servo of the drive selected after power-on or if the SERVORDY signal fails to appear after a reasonable amount of time. |

## Table 3.2 VIA signals (continued)

| VIA Name | FS Name | Input/Output | Description |
|---|---|---|---|
| PA7 | /SCCWREQ | I | **Serial Communications Controller Write Request.** This active low signal indicates the SCC is ready to transmit the next byte. Used by high speed serial protocols to avoid reading the SCC directly. |
| PB0 | SERVO/CLK | O | **Servo/Real Time clock select.** This signal is used to set the SERIAL PAL for either disk servo or clock chip communication as the 8 bit shift register of the VIA is shared between the two. This connected to the enable of the clock chip so a low value talks to the clock, a high value to the disk servo. |
| PB1 | STBIT | O | **Start Bit.** This signal is asserted when a byte is to be transmitted from the 65C22 shift register to the disk drive selected. The SERIAL PAL starts counting out the 10 bit times of the asynchronous character and lowers SDATA to form the start bit. |
| PB2 | SERVOERR | I | **Servo Error.** The disk drive servo of the selected drive is in an error state. Servo Reset is required to use the drive again. |
| PB3 | HS0 | O | **Head Select 0.** Head select 0 selects the low order bit of the surface number to access. |
| PB4 | HS1 | O | **Head Select 1.** Head select 1 selects the high order bit of the surface number to access. |
| PB5 | /POWEROFF | O | **Power off.** Asserting this signal turns off the power supply. |
| PB6 | INDEX | I | **Index pulse.** The index pulse of the disk drive selected. |
| PB7 | OCLK | O | **Output clock.** The output of the T1 timer used as the bit rate of the serial bit streams to the disk drive and the RTC. |
| CB1 | SCLK | I | **Shift Clock.** This is the shift clock fed back in from the SERIAL PAL to shift the 8 bit shift register. This OCLK is the source of this clock. |
| CB2 | SDATA | I/O | **Shifter data.** The output and input to the 65C22 shifter. This is the serial bit stream to the disk drive and RTC. |

## 3.2 Real Time Clock

The Real Time Clock chip is an 8 pin DIP implemented in CMOS technology. It contains 256 bytes of RAM and a 4 byte binary up counter. The counter forms a 32 bit binary up counter which is clocked at 32.768 KHz.

The clock chip is selected by lowering SERVO/CLK so the Chip Enable of the clock chip is asserted. The SR is programmed for output under control of phase 2. When the command byte is written to SR the data will be clocked to the clock chip. Then, the byte to send to the clock is written to the shift register. If data is to be read back, the shift register is programmed for input and a dummy read to SR will start clocking the data from the clock chip.

Further information can be found in SERIAL CLOCK AND 256 BYTE RAM SPECIFICATION, Bob Bailey, 7-2-84.

## 3.3 SERIAL PAL

The SERIAL PAL controls the 8 bit bidirectional synchronous shift register of the VIA to send and receive the 10 bit asynchronous characters of the disk servo. The PAL is controlled by the /STBIT and SERVO/CLK inputs.

When SERVO/CLK is low, the SDATA output is disabled and the clock chip is selected. When SERVO/CLK is high the /STBIT is significant. When /STBIT is high, serial data from the disk servo is allowed to pass into the shift register. When /STBIT is asserted it indicates the start bit should be sent to the disk servo serial input. The timing for this bit and the rest of the bits comes from OCLK.

The OCLK is the bit timer which increments the 4 bit up counter implemented in the SERIAL PAL. Count 0 waits for /STBIT to be asserted, count 1 is the start bit, counts 2 through 9 are the data bits from the shift register, and count 10 waits for /STBIT to be deasserted. Counts 2 through 9 also pulse the SCLK which shifts the shift register.

The SERIAL PAL has an additional function of inverting the FC2 signal of the 68000 to use in selecting the 65C22.

## 3.4 Soft Power

The FS has a 'Hard On/Soft Off' circuit to control the power supply. The circuit uses a 74HC132, a 3.6 V lithium battery, a 3906 transistor, a thermal switch, and several capacitors and resistors.

The circuit is designed to attempt to turn on the power supply for 3 seconds after the power switch is pressed. The 74HC132 is a Schmitt triggered set of NAND gates which are configured to form a SR flip-flop. When the power switch is pressed it discharges a 47 µF capacitor through a 100 Ω resistor. The capacitor is connected to the /Set input of the flip-flop which is the power-on request. The capacitor is charged through a 100K Ω resistor to the 3.6 volts of the battery. While the /Set input is asserted a 3906 transistor is saturated between the battery and and optotriac located on the power supply board. The optotriac is an infared LED which when driven allows AC current to flow in the power supply. If AC current is present the power supply will power the 3906 and the circuit will remain in this state.

The power off function is under software control. When the power switch is pressed it also generates an interrupt to the 68000. This allows the FS to clean up any pending activity before power down. When the FS is ready to power down the /POWEROFF signal from the VIA is asserted which is connected to the /Reset of the flip-flop, turning off the 3906, optotriac and the power supply in general.

The power circuit has a safety feature to prevent the box from overheating in the case of a fan failure or blocked ventilation. The 3906 to optotriac current path has a thermal switch in series to open if the temperature exceeds 65 C. The opening of this switch will immediately turn off the power supply.

If the software fails to turn off the machine the user has the ultimate control - unplug it.

## 4.0 Serial Interface

The Serial Interface consists of a 4Mz Z8530 Serial Communications Controller (SCC) two 26LS30 driver chips, two 75175 receiver chips, two RC filter networks, and two miniature DIN 8 pin connectors.

The serial interface is very flexible in the types of serial protocols supported. Asynchronous modems and printers, synchronous modems with IBM Bisync or SDLC, and AppleTalk are all supported.

## 4.1 Serial Communications Controller (8530)

The SCC contains 13 write registers in each channel that are programmed by the FS separately to configure the functional personality of the channels.

In the SCC, register addressing is direct for the data registers only. In all other cases ( with the exception of WR0 and RR0 ), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 (the Command Register) and contains three bits that point to the selected register. The second write (also to the Command Register) is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed. All of the registers in the SCC, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 ( or RR0 ) is addressed again. All address references to the SCC use $D8 0000 as the base address and the offsets to the command and data registers listed in Table 4.1

The SCC has a timing restriction in the time between accesses to the chip. Accesses to the chip must be at least 1.8 µs from the end of the first access to the beginning of the second. This is equivalent to executing a 12 clock period instruction between accesses to the SCC.

The PCLK into the SCC is /C4M which is 3.9168 MHz. The /RTxC inputs normally receive /C3M which is 3.6864 MHz. Port A has an optional mode GPi is connected to /RTxCA when /APTKA is deasserted. This is used to connect a synchronous modem clock to /RTxCA.

Before using the SCC the chip should be reset using a ***word*** access to $D8 0000.

## Table 4.1 SCC register offsets

| Address Offset | Comments |
|---|---|
| $0 | **Reset the SCC (Read access).** This is the only word access to the SCC and is useful to reset the SCC from a hardware point of view since the SCC doesn't have a power on reset input. |
| $2 | **Port A Command Register (Read access).** SCC registers RR0-RR3, RR10, RR12, RR13, and RR15 are accessed through this address. |
| $3 | **Port A Command Register (Write access).** SCC registers WR0-WR7, WR9-WR15 are accessed through this location. |
| $6 | **Port A Data Register (Read access).** SCC register RR8 is accessed through this location. |
| $7 | **Port A Data Register (Write access).** SCC register WR8 is accessed through this location. |
| $0 | **Port B Command Register (Read access).** SCC registers RR0-RR3, RR10, RR12, RR13, and RR15 are accessed through this address. |
| $1 | **Port B Command Register (Write access).** SCC registers WR0-WR7, WR9-WR15 are accessed through this location. |
| $4 | **Port B Data Register (Read access).** SCC register RR8 is accessed through this location. |
| $5 | **Port B Data Register (Write access).** SCC register WR8 is accessed through this location. |

## 4.2 MINI DIN-8 connectors

The serial interface is connected to the external world through two eight pin miniture DIN connectors. These connectors are interfaced through two 26LS30 and two 75175 chips to the SCC. Each signal pin passes through an RC filter network. The network is a low pass T configuration with two 39Ω resistors and a 200 pF capacitor to ground. Figure 4.1 shows the physical layout of the pins on the connector as seen from the back of the FS. Tables 4.2 and 4.3 indicate how each of the eight pins are connected to SCC.

When the Receive Data is RS-232 single ended input the RxD- pin should be tied to the SG pin.

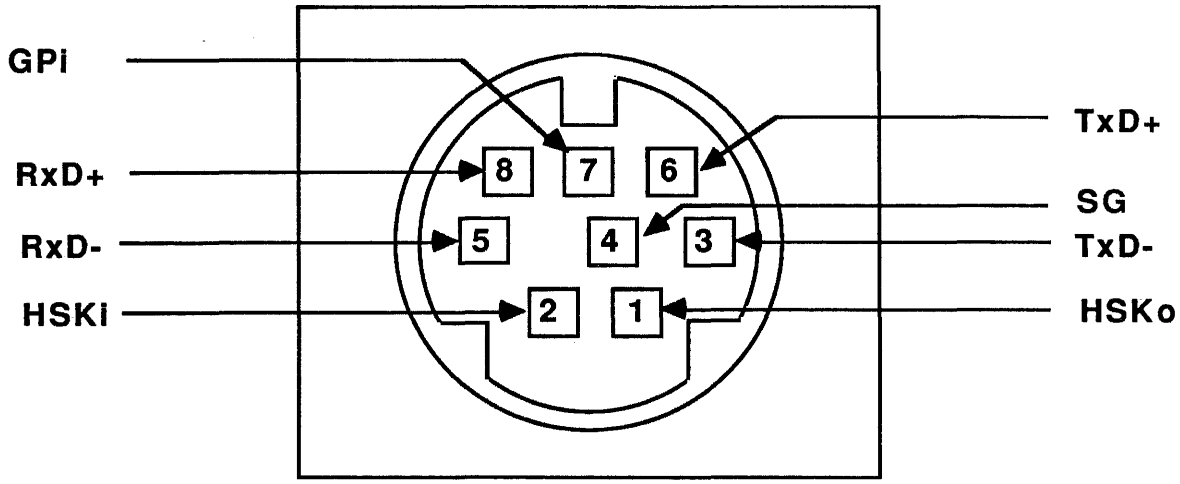


**Figure 4.1 MINI DIN-8 connector**

## Table 4.2 MINI DIN-8 (PORT A)

| Pin # | Name | Comments |
|---|---|---|
| 1 | HSKo | **Hankshake output.** Connected to SCC Request To Send (RTS). Tri-stated when Data Terminal Ready (DTR) deasserted. $V_{oh}$ = 3.6V, $V_{ol}$ = -3.6V, $R_l$ = 450Ω. |
| 2 | HSKi | **Handshake input.** Connected to SCC Clear To Send (CTS) and Transmit /Receive Clock (TRxC). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |
| 3 | TxD- | **Transmit Data (inverted).** Connected to SCC Transmit Data (TxD). Tri-stated when Data Terminal Ready (DTR) deasserted. $V_{oh}$ = 3.6V, $V_{ol}$ = -3.6V, $R_l$ = 450Ω. |
| 4 | SG | **Signal Ground.** Connected to logic and chassis ground. |
| 5 | RxD- | **Receive Data (inverted).** Connected to SCC Receive Data (RxD). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |
| 6 | TxD+ | **Transmit Data.** Connected to the SCC Transmit Data (TxD). Tri-stated when Data Terminal Ready (DTR) deasserted. $V_{oh}$ = 3.6V, $V_{ol}$ = -3.6V, $R_l$ = 450Ω. |
| 7 | GPi | **General Purpose Input.** Connected to the SCC Data Carrier Detect (DCD). Also connected to Receive/Transmit Clock (RTxC) if /APTKA deasserted. Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |
| 8 | RxD+ | **Receive Data.** Connected to the SCC Receive Data (RxD). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |

## Table 4.3 MINI DIN-8 (PORT B)

| Pin # | Name | Comments |
|---|---|---|
| 1 | HSKo | **Hankshake output.** Connected to SCC Request To Send (RTS). Tri-stated when Data Terminal Ready (DTR) deasserted. $V_{oh}$ = 3.6V, $V_{ol}$ = -3.6V, $R_l$ = 450Ω. |
| 2 | HSKi | **Handshake Input.** Connected to SCC Clear To Send (CTS) and Transmit /Receive Clock (TRxC). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |
| 3 | TxD- | **Transmit Data (Inverted).** Connected to SCC Transmit Data (TxD). Tri-stated when Data Terminal Ready (DTR) deasserted. $V_{oh}$ = 3.6V, $V_{ol}$ = -3.6V, $R_l$ = 450Ω. |
| 4 | SG | **Signal Ground.** Connected to logic and chassis ground. |
| 5 | RxD- | **Receive Data (Inverted).** Connected to SCC Receive Data (RxD). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |
| 6 | TxD+ | **Transmit Data.** Connected to the SCC Transmit Data (TxD). Tri-stated when Data Terminal Ready (DTR) deasserted. $V_{oh}$ = 3.6V, $V_{ol}$ = -3.6V, $R_l$ = 450Ω. |
| 7 | GPi | **General Purpose Input.** Connected to the SCC Data Carrier Detect (DCD). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |
| 8 | RxD+ | **Receive Data.** Connected to the SCC Receive Data (RxD). Input resistance = 12 KΩ, $V_{ih}$ = 0.2V, $V_{il}$ = -0.2V. |

## 5.0 Small Computer Standard Interface (5380)

The Small Computer Standard Interface (SCSI) consists of the NCR 5380 chip and 50 pin shielded ribbon connector.

The NCR5380 is a 40 pin NMOS device designed to support the SCSI as defined by the American National Standards Institute (ANSI) X3T9.2 committee. This device supports arbitration of the SCSI bus, including reselection. The chip is controlled through a set of read and write registers located at $C0 0000 with the offset addresses in Table 5.1. Note that the DMA registers are in a different address space as they perform a psuedo DMA access with the 68000 access generating a DMA handshake. The psuedo DMA mode lets the 5380 perform the SCSI bus handshake so the 68000 merely has to poll the chip for the completion of this handshake.

The NCR5380 is connected directly to the 50 pin shielded ribbon connector. The chip is capable of sinking 48 mA through each of the pins connected to the bus. The data and control lines on the SCSI bus are active low signals driven by open drain outputs. Pin 24 is connected through a diode to the POWER pin of the power supply connector. Applying 15 mA at 2V to pin 24 will power up the power supply only as long as this current is supplied. A 47Ω resistor to +5V will work fine.

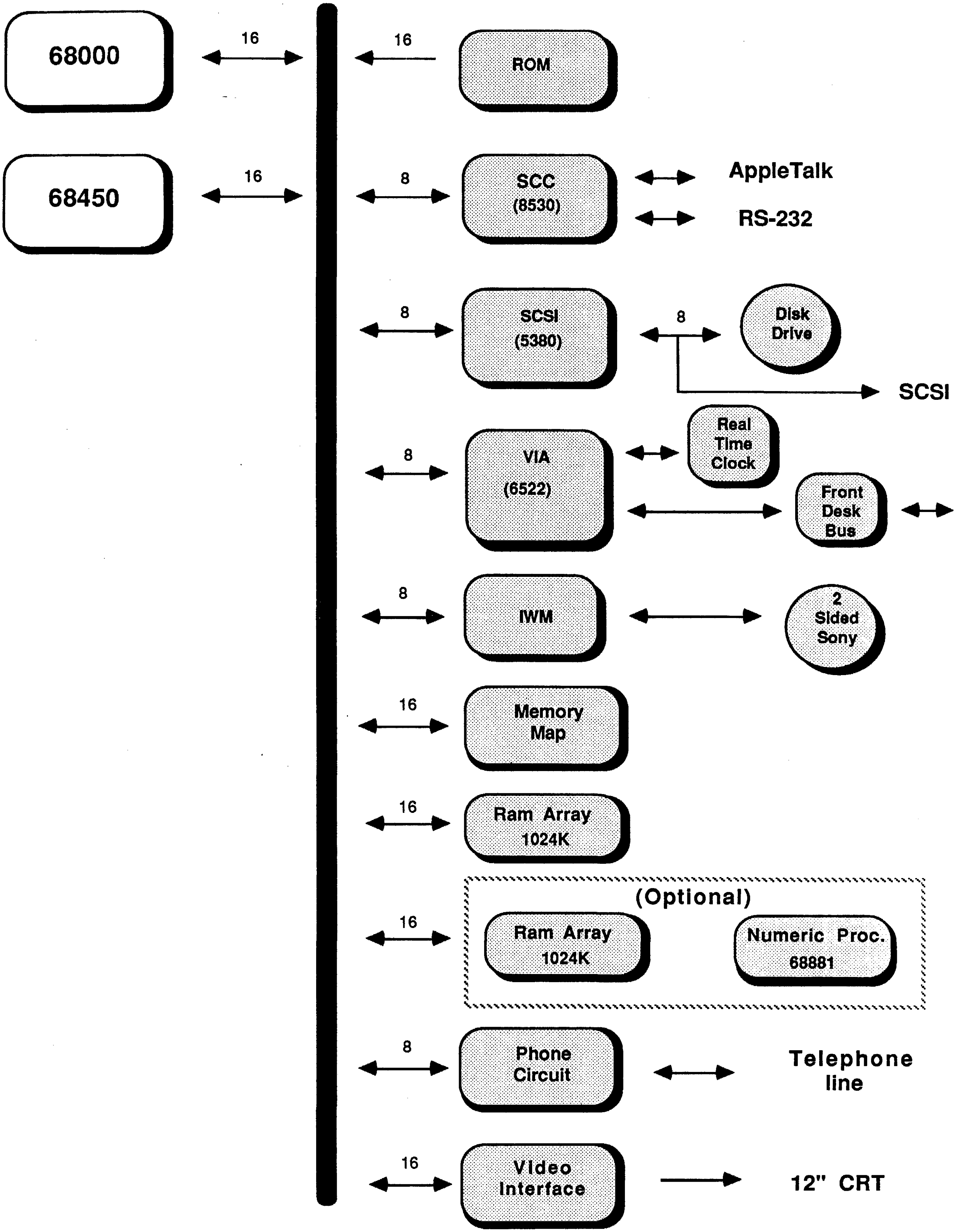
The bus can be externally pulled up using the +5V power on pin 26. The +5V is connected with a 1N4001 diode in series so the actual voltage will be 4.3V typically. The normal pullup connection is a 220Ω resistor to +5V and a 330Ω resistor to ground on each of the active pins. The pinout of the SCSI bus and its connection to the 5380 is shown in Table 5.2. Pin 1 of the 50 pin connector is the lower left pin with pin 2 directly above.

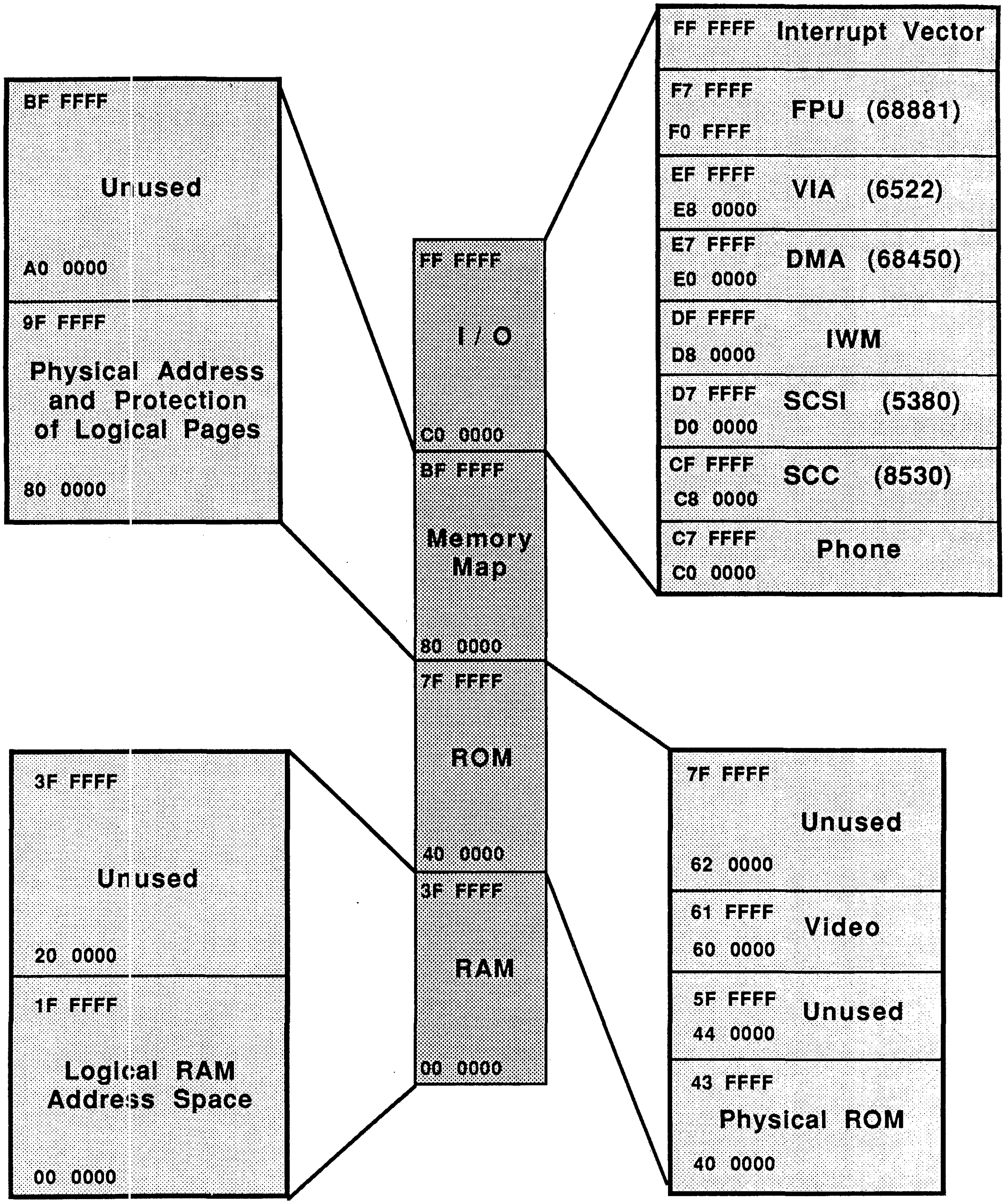
## Table 5.1 SCSI (5380) register offsets

| Offset Address | Name |
|---|---|
| $01 | **Current SCSI Data.** Read only. |
| $11 | **Initiator Command Register.** Read only. |
| $21 | **Mode Register.** Read only. |
| $31 | **Target Command Register.** Read only. |
| $41 | **SCSI Bus Status.** Read only. |
| $51 | **Bus and status register.** Read only. |
| $61 | **Input Data Register.** Read only. |
| $71 | **Reset Parity/Interrupts.** Read only. |
| $00 | **Output Data Register.** Write only. |
| $10 | **Initiator Command Register.** Write only. |
| $20 | **Mode register.** Write only. |
| $30 | **Target Command Register.** Write only. |
| $40 | **Select Enable Register.** Write only. |
| $50 | **Start DMA Send.** Write only. |
| $60 | **Start Target Receive DMA.** Write only. |
| $70 | **Start Initiator Receive DMA.** Write only. |
| $8 0000 | **DMA Read.** Read DMA data byte during DMA mode. Read only. |
| $8 0001 | **DMA Write.** Write DMA data byte during DMA mode. Write only. |

## Table 5.2 SCSI bus pinout

| Pin # | SCSI bus name | 5380 pin name |
|---|---|---|
| 1 | GND | |
| 2 | /DATA0 | /DB0 |
| 3 | GND | |
| 4 | /DATA1 | /DB1 |
| 5 | GND | |
| 6 | /DATA2 | /DB2 |
| 7 | GND | |
| 8 | /DATA3 | /DB3 |
| 9 | GND | |
| 10 | /DATA4 | /DB4 |
| 11 | GND | |
| 12 | /DATA5 | /DB5 |
| 13 | GND | |
| 14 | /DATA6 | /DB6 |
| 15 | GND | |
| 16 | /DATA7 | /DB7 |
| 17 | GND | |
| 18 | /PARITY | /DBP |
| 19 | GND | |
| 20 | no connect | |
| 21 | GND | |
| 22 | no connect | |
| 23 | GND | |
| 24 | PWRON | |
| 25 | no connect | |
| 26 | TERMPWR (+5V) | |
| 27 | GND | |
| 28 | no connect | |
| 29 | GND | |
| 30 | no connect | |
| 31 | GND | |
| 32 | /ATN | |
| 33 | GND | |
| 34 | GND | |
| 35 | GND | |
| 36 | /BUSY | /BSY |
| 37 | GND | |
| 38 | /ACK | /ACK |
| 39 | GND | |
| 40 | /RST | /RST |
| 41 | GND | |
| 42 | /MSG | /MSG |
| 43 | GND | |
| 44 | /SEL | /SEL |
| 45 | GND | |
| 46 | C/D | /C/D |
| 47 | GND | |
| 48 | REQ | /REQ |
| 49 | GND | |
| 50 | I/O | /I/O |


68000
16
16
ROM
68450
16
8
SCC
(8530)
AppleTalk
RS-232
8
SCSI
(5380)
8
Disk
Drive
SCSI
8
VIA
(6522)
Real
Time
Clock
Front
Desk
Bus
8
IWM
2
Sided
Sony
16
Memory
Map
16
Ram Array
1024K
(Optional)
16
Ram Array
1024K
Numeric Proc.
68881
8
Phone
Circuit
Telephone
line
16
Video
Interface
12" CRT


BF FFFF
Unused
A0 0000
9F FFFF
Physical Address and Protection of Logical Pages
80 0000
3F FFFF
Unused
20 0000
1F FFFF
Logical RAM Address Space
00 0000
FF FFFF
I / O
C0 0000
BF FFFF
Memory Map
80 0000
7F FFFF
ROM
40 0000
3F FFFF
RAM
00 0000
FF FFFF
Interrupt Vector
F7 FFFF
FPU (68881)
F0 FFFF
EF FFFF
VIA (6522)
E8 0000
E7 FFFF
DMA (68450)
E0 0000
DF FFFF
IWM
D8 0000
D7 FFFF
SCSI (5380)
D0 0000
CF FFFF
SCC (8530)
C8 0000
C7 FFFF
Phone
C0 0000
7F FFFF
Unused
62 0000
61 FFFF
Video
60 0000
5F FFFF
Unused
44 0000
43 FFFF
Physical ROM
40 0000