MC68HC99

Technical Summary

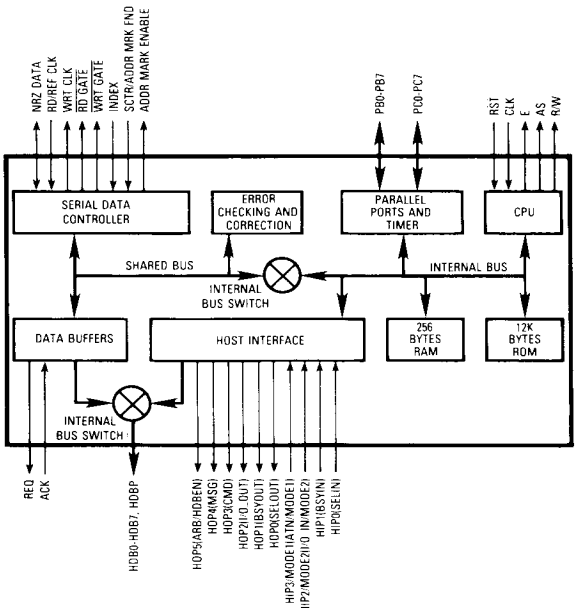
Motorola's MC68HC99 hard disk controller (HDC) provides an economical solution to the problem of controlling one or more hard disks. The high-speed, low-cost HCMOS technology of the MC68HC99 replaces the multiple-chip set small computer system interface (SCSI) controller solutions used in current board-level and embedded controller designs. The following block diagram shows how the HDC incorporates all the elements of a disk controller in a single package.

Some hardware and software features of the MC68HC99 are as follows:

- High-Speed Serial Data Controller
- Low-Power HCMOS Technology
- Reed-Solomon Error Detection/Correction
- Two On-Chip 528-Byte Rotating Data Buffers
- MC68HC11 Central Processing Unit (CPU) with 256 Bytes of Random-Access Memory (RAM)
- MC68HC11 Core with 12K Bytes of Read-Only Memory (ROM)
- 16-Bit timer with Input Capture and Output Compare Functions
- Programmable Host Interface
  - — Suitable for Both SCSI and Non-SCSI Applications
  - — Completely SCSI Compatible
- Four Operating Modes
  - — Single Chip
  - — Simple Expanded (for External Peripheral Devices)
  - — ROM Expanded (for Additional Firmware Space)
  - — ROM Expanded with Timer (Adds External Timer Functions)

3

BLOCK DIAGRAM



This document contains information on a new product. Specifications and information herein are subject to change without notice.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the HDC using these two pins. V_{DD} is the power input (+5.0 V), and V_{SS} is ground.

CLOCK INPUT (CLK)

Clock input must be a transistor-to-transistor logic (TTL) compatible signal that is internally buffered to develop the internal clocks needed by the HDC. This clock input should not be gated off at any time.

E-CLOCK (E)

This clock output signal provides a timing reference for CPU bus cycles on the expanded bus. When E is low, an internal cycle is taking place. High E indicates that an external bus access is taking place. E is valid regardless of which mode the HDC is configured to operate in.

RESET ($\overline{RST}$)

This input signal is used to reset the HDC. A logic low on this pin preemptively resets the HDC and drives all outputs to their quiescent state.

ADDRESS STROBE (AS)

This output signal is used to demultiplex data signals from ports B and C. AS is high one-eight of a cycle after the falling edge of E, and low one-eight of a cycle before the rising edge of E. AS is used to capture low-order address information with a transparent latch. AS is valid regardless of which mode the HDC is configured to operate in.

READ/WRITE (R/$\overline{W}$

This output signal controls the direction of transfers on the external data bus. A high on this pin indicates a read cycle; a low indicates a write cycle. R/$\overline{W}$ is valid regardless of which mode the HDC is configured to operate in.

HOST DATA BUS (HDB0-HDB7, HDBP)

The host data bus consists of eight data signals (HDB0-HDB7), and a parity signal (HDBP). This bus is used to transfer commands, status, and data between the HDC and its host computers. Internally, the host data bus connects with the host data buffers and with the host data register (HDR).

HOST-INTERFACE INPUT/OUTPUT PINS (HIP0-HIP3, HOP0-HOP5)

The HDC host interface includes four input signals (HIP0-HIP3) and six output signals (HOP0-HOP5). Each input signal has a corresponding bit in the host status register (HSR). Each output signal has a corresponding bit in the host pin control register (HPCR). When the HDC is used as an SCSI controller, the output signals drive the select (SEL), busy (BSY), input/output (I/O), command (CMD), and message (MSG) signals. Under the same conditions, the input signals monitor the SEL, BSY, I/O, and attention (ATN) signals.

TRANSFER REQUEST (REQ) AND ACKNOWLEDGE (ACK)

The output signal, REQ, and input signal, ACK, are used to transfer data asynchronously between the HDC data buffers and the host computer.

SERIAL INTERFACE SIGNALS

The following are the high-speed serial data controller signals. Three of these signals, NRZ DATA, WRT CLK, and ADDR MARK ENABLE must be of equal current and capacitance in order to maintain timing relationships.

Nonreturn To Zero Data (NRZ DATA)

This bidirectional signal transfers serial NRZ data between the HDC and the disk.

Read/Reference Clock (RD/REF CLK)

This input signal from the disk determines the data transfer rate.

Write Clock (WRT CLK)

This output signal is derived from the reference clock input. It is supplied prior to beginning a write data operation and exists for the duration of the write operation.

Read Gate ($\overline{RD}$ $\overline{GATE}$)

When asserted, this active-low output signal indicates a disk read operation.

Write Gate ($\overline{WRT}$ $\overline{GATE}$)

When asserted, this active-low output signal indicates a disk write operation.

Index Detect (INDEX)

This input signal, when asserted, indicates the beginning of a track on the disk.

Sector/Address Mark Found (SCTR/ADDR MRK FND)

This input signal is dependent on the type of disk attached to the HDC. With a hard-sectored disk, this signal indicates the beginning of a sector. With a soft-sectored disk, this signal is driven by an external ADDRESS MARK DETECT circuit.

Address Mark Enable (ADDR MARK ENABLE)

The function of this input signal is dependent on the type of disk attached to the HDC and the programmed operation of the HDC.

If the HDC is programmed for the soft-sectored enhanced small disk interface (ESDI) disk format and the operation is write format, then this signal, asserted with $\overline{WRT}$ $\overline{GATE}$, causes the ESDI disk to write an address mark. Negation of ADDR MARK ENABLE with $\overline{WRT}$ $\overline{GATE}$ asserted indicates the beginning of an ID PL0 sync field. In all functions other than write format, ADDR MARK ENABLE is asserted instructing the soft-sectored ESDI disk to search for an address mark field.

If the HDC is programmed for the hard-sectored ESDI disk format and the operation is write format, then ADDR MARK is asserted during the preamble field to indicate

the beginning of an ID PL0 sync field. ADDR MARK ENABLE is not asserted in operations other than write format.

If the HDC is programmed for the ST-506 disk format and the operation is write format, then ADDR MARK ENABLE is asserted with $\overline{\text{WRT GATE}}$ for one bit time during the sixth bit of the address mark field. This pulse is used externally to generate an illegal clock pattern that constitutes a unique address mark.

INPUT/OUTPUT PROGRAMMING

Three of the five I/O parallel port registers available with the MC68HC99 HDC are used in the serial data controller and the host interface. The function of the remaining parallel ports, B and C, is dependent on the operating mode selected.

SINGLE-CHIP MODE

The single-chip mode allows only the on-chip resource shown in the block diagram to be available to the HDC. The CPU fetches interrupt vectors and executes its code from the ROM.

In this mode, port B pins 0-7 are general-purpose I/O pins. Pins PB6 and PB7 can be programmed to function as the timer capture input signal (TCAP) and the timer compare output signal (TCMP), respectively. This reprogramming is accomplished by setting the TCAP/TCMP enable (TEN) bit of the highest priority interrupt register (HPIR).

All port C pins are general-purpose I/O pins in this mode.

EXPANDED MODE 1

Expanded mode 1 uses the internal ROM for user firmware and CPU fetches of all interrupt vectors. Additional peripheral devices or memory can be added via the expanded bus.

In this mode, all port B pins (0-7) act as high-order address output pins. During each MCU cycle, bits 8-15 of the address are output on PB0-PB7.

All port C pins are configured as multiplexed address/data pins. During the address portion of each microcomputer unit (MCU) cycle, bits 0-7 of the address are output on PC0-PC7. During the data portion of the MCU cycle (E high), bits PC0-PC7 are bidirectional data pins controlled by the R/$\overline{\text{W}}$ signal.

EXPANDED MODE 2

Expanded mode 2 uses external memory for CPU fetches of firmware and interrupt vectors. After reset, the internal ROM is left in the memory map. This ROM can be removed from the memory map by setting the remove internal ROM (REMROM) bit of the HPIR. This removal adds 12K to the external memory space.

All of the port B pins (0-7) act as high-order address output pins. During each MCU cycle, bits 8-15 of the address are output on PB0-PB7.

All port C pins are configured as multiplexed address/data pins. During the address portion of each MCU cycle, bits 0-7 of the address, are output on PC0-PC7. During the data portion of the MCU cycle (E high), bits PC0-PC7 are bidirectional data pins controlled by the R/$\overline{\text{W}}$ signal.

EXPANDED MODE 3

Expanded mode 3 is the same as expanded mode 2, except for the configuration of parallel port B.

Port B pins 0-5 act as high-order address output pins. During each MCU cycle, bits 8-13 of the address, are output on PB0-PB5. PB6 and PB7 are configured as multiplexed, high-order address/timer pins. During the address portion of each MCU cycle, bits 14 and 15 of the address are output on PB6 and PB7, respectively. During the data portion of the MCU cycle (E high), PB6 is TCAP (input), and PB7 is TCMP (output) for the timer.

All port C pins function the same way as in expanded mode 2.

NOTE

When the HDC is configured to operate in any of the expanded modes, data written to the internal registers and internal RAM is also driven onto the external data bus. If data is written to the internal ROM while the REMROM bit in the HPIR is logic zero, the data is not driven onto the external bus.

MEMORY

The MC68HC99 HDC has 12K bytes of available ROM and 256K bytes of available RAM. The memory is mapped for the HDC single-chip mode as shown in Figure 1, for the expanded mode 1 as shown in Figure 2, and for expanded modes 2 and 3 as shown in Figure 3.

REGISTERS

The CPU of the HDC uses the basic core of the MC68HC11 microcomputer. Seven CPU registers are available to programmers, see Figure 4.

ACCUMULATORS A AND B

Accumulators A and B are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. The two accumulators can be concatenated into a single 16-bit accumulator called the D accumulator.

INDIRECT REGISTER X (IX)

The 16-bit IX register is used for indexed mode addressing. It provides a 16-bit indexing value that is added to an 8-bit offset in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

INDIRECT REGISTER Y (IY)

The 16-bit IY register is also used for indexed mode addressing, similar to the IX register. All instructions using the IY register require an extra byte of execution time because these instructions use off-page addressing.

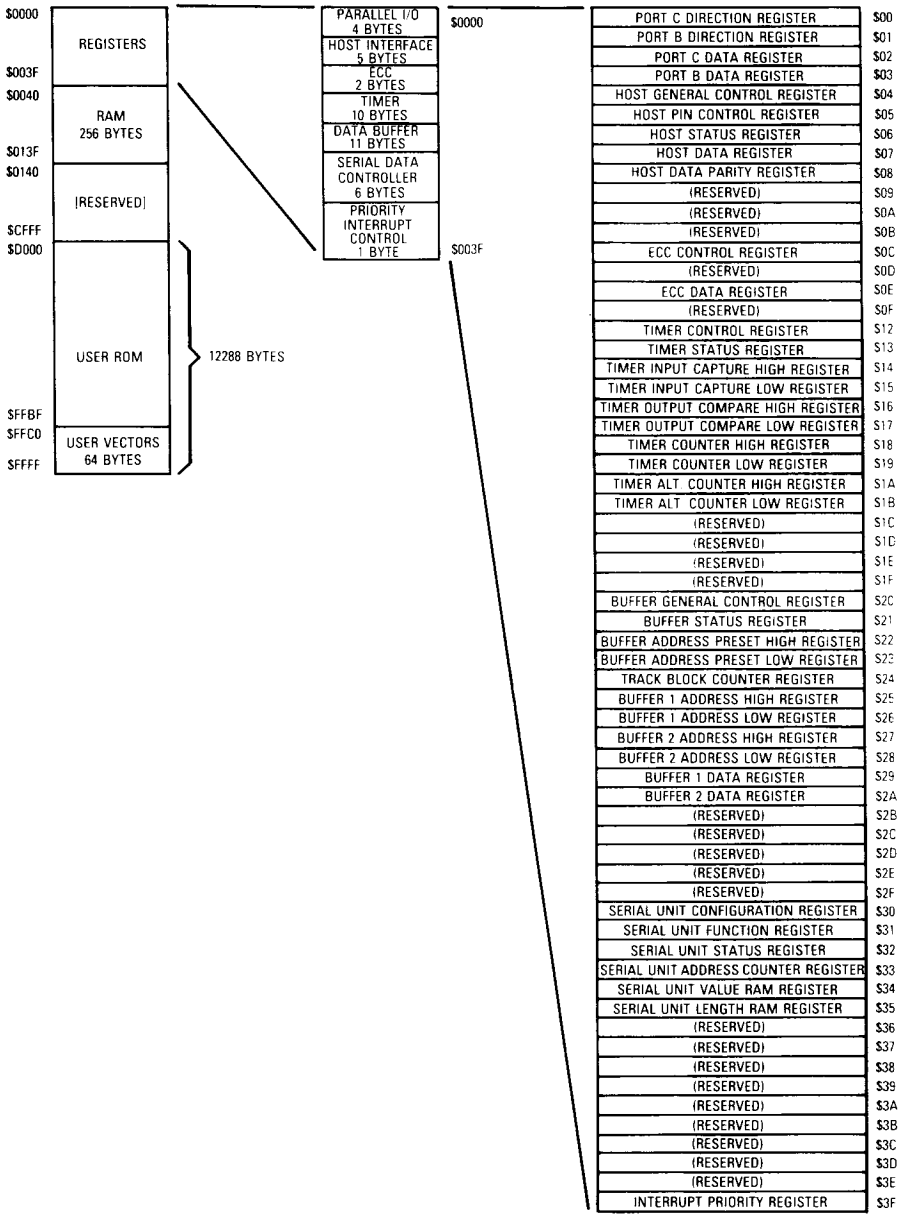


Figure 1. HDC Single-Chip Mode Memory Map

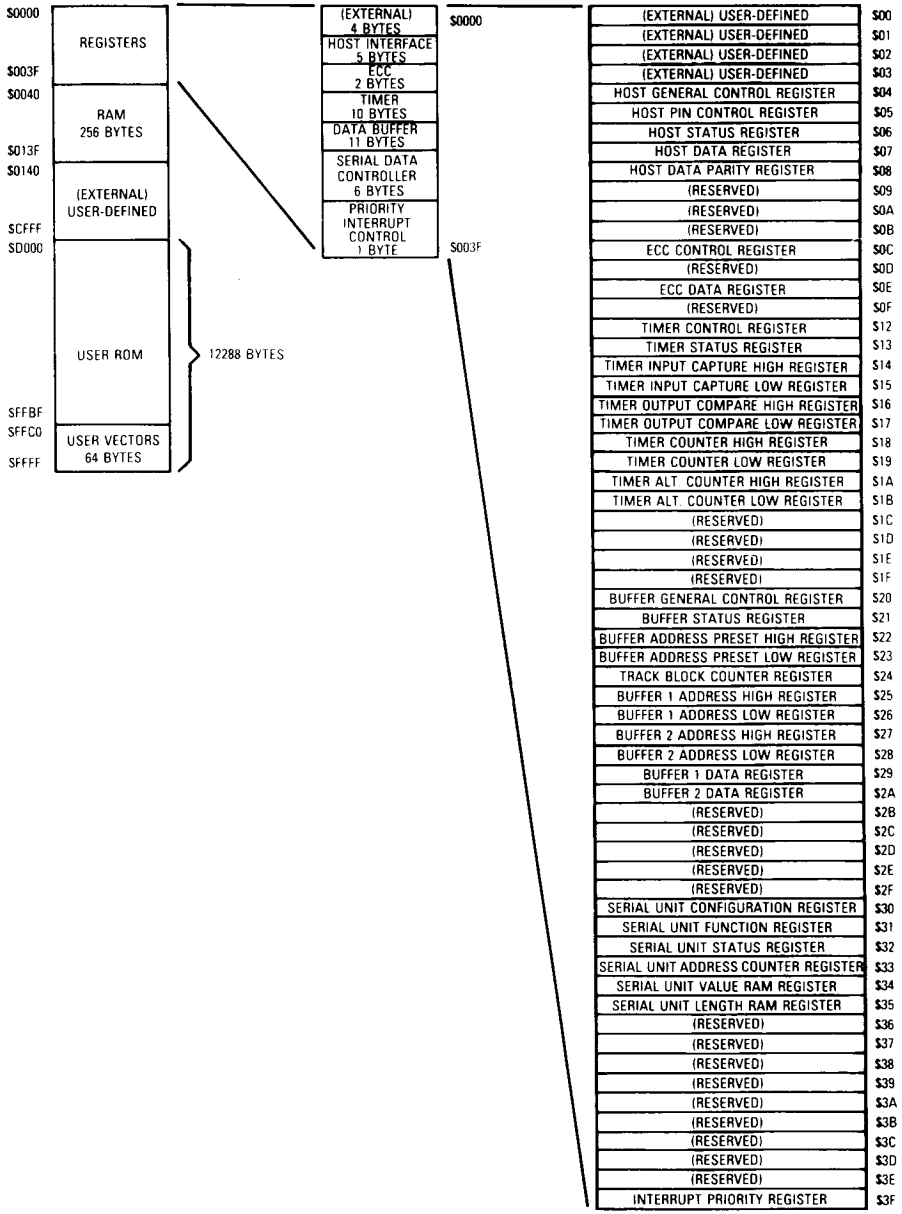


Figure 2. HDC Expanded Mode 1 Memory Map

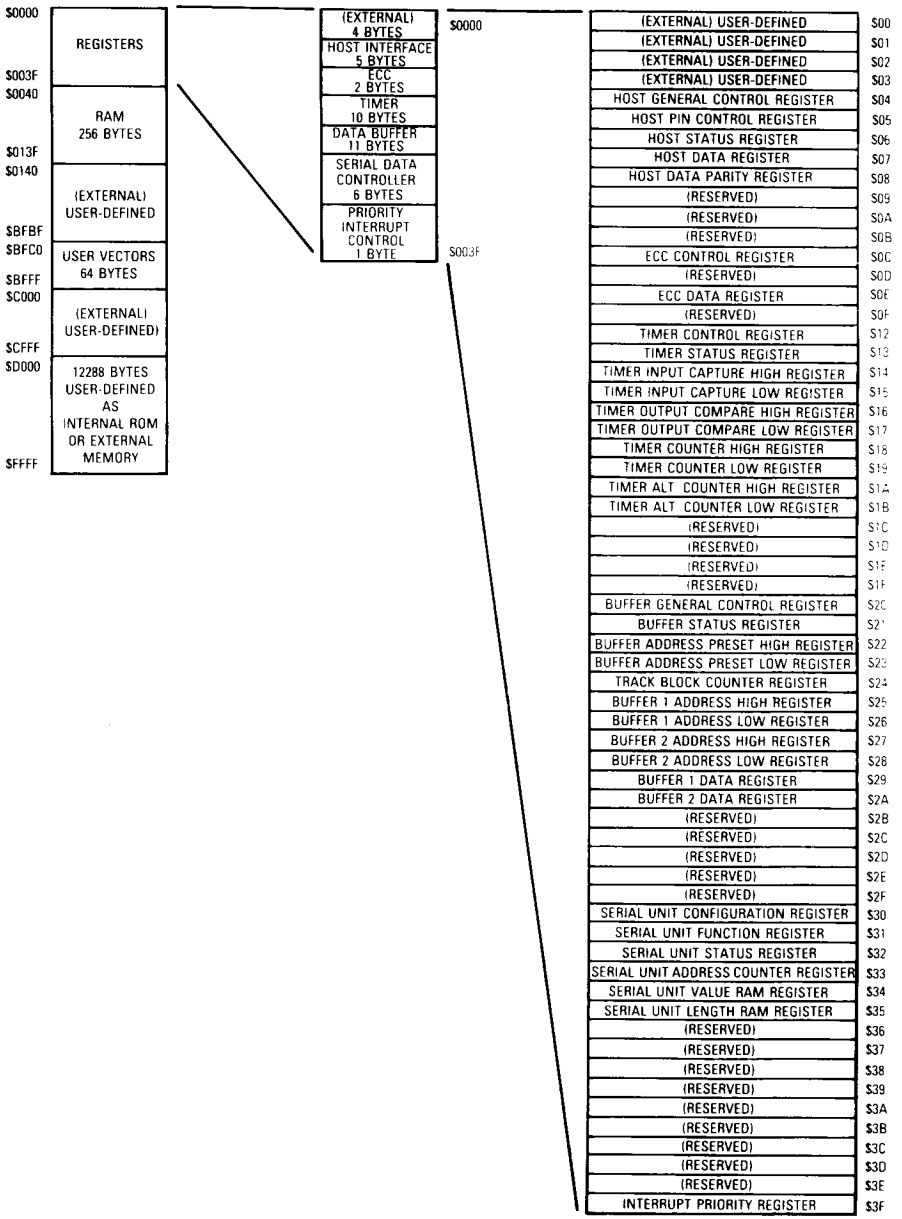


Figure 3. HDC Expanded Modes 2 and 3 Memory Map

3

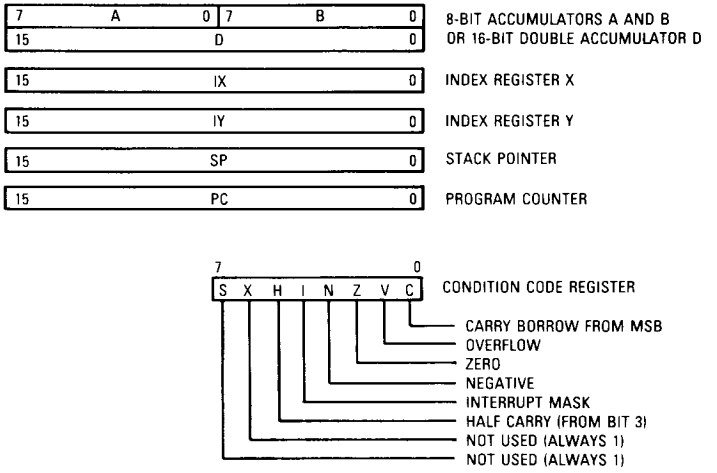


Figure 4. Programming Model

3

PROGRAM COUNTER (PC)

The PC is a 16-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP)

The SP is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of LIFO read/write registers that allow important data, such as the PC and CCR, to be stored during interrupts and subroutine calls. Each push decrements the stack; each pull increments it. Stacking order is shown in Figure 5.

CONDITION CODE REGISTER (CCR)

The CCR is an 8-bit register with each bit signifying the results of the instruction just executed. These bits are program testable, allowing specific action to be taken based on test results. The condition code register bits are given following.

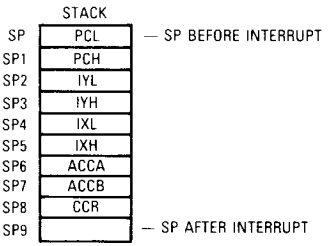


Figure 5. Interrupt Stacking Order

Carry/Borrow (C).

The C bit is set if a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last operation. The C bit is also effected during shift and rotate instructions.

Overflow (V)

The V bit is set if an arithmetic overflow resulted from the last operation. Otherwise, the V bit is cleared.

Zero (Z)

The Z bit is set if the result of the last arithmetic, logic, or data operation was zero. Otherwise, the Z bit is cleared.

Negative (N)

The N bit is set if the result of the last arithmetic, logic, or data operation was negative. Otherwise, the N bit is cleared.

INTERRUPT MASK (I)

The I bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources, (both external and internal).

Half Carry (H)

The H bit is set when a carry occurs between bits three and four of the ALU during an ADD, ABA, or ADC instruction. Otherwise, the H bit is cleared.

HOST INTERFACE

The host interface provides the connection between the HDC and one or more host computers. Although the host interface conforms to the proposed ANSI X3T9.2 SCSI bus specification, it can operate with other interfaces. The HDC host interface consists of five registers through which the firmware is able to:

- Control the host-interface operations,
- Monitor and manipulate the host-interface control signals, and
- Read or write the host-interface data bus.

In addition to these registers, the host interface contains special-purpose hardware that allows the HDC to function as an SCSI target device.

NOTE

As a convention, reset values for register bits in all MC68HC99 registers are defined as follows:

0 = Bit is reset to zero.
1 = Bit is reset to one.
U = Bit is not affected by reset.
R = Bit value is determined by an input pin at reset.

HOST GENERAL CONTROL REGISTER (HGCR) $0004

The HGCR controls the host-interface interrupt activity. Bits HGCR5-HGCR7 are used only when the HDC is functioning as an SCSI controller.

7	6	5	4	3	2	1	0
REC	ARBINT	SELINT	HSTINT (ATNINT)	0	0	0	0
RESET: 0	0	0	0	0	0	0	0

REC — Start Arbitration/Reconnection Operation Bit
When the operation is completed, REC is cleared by the host interface and cannot be reset to zero by a CPU write to the HGCR. REC should be used only when the HDC is an SCSI controller.
1 = Operation in progress
0 = Operation completed

ARBINT — Arbitration Begun Interrupt Enable Bit
ARBINT should be used only when the HDC is an SCSI controller.
1 = Enable
0 = Disable

SELINT — Selection Interrupt Enable Bit
SELINT should be used only when the HDC is an SCSI controller.
1 = Enable
0 = Disable

HSTINT — Host Interrupt Enable (Attention Interrupt Enable) Bit
1 = Enable
0 = Disable

Bits 3–0 — Not used.
Unused bits in the HGCR always read as zeros.

HOST PIN CONTROL REGISTER (HPCR) $0005

The HPCR controls the output levels of the host output port pins (HOP0–HOP5) as well as the direction of bits 0-7 of the host-interface data register and the data bus parity (DBP) bit in the host-interface data parity register.

7	6	5	4	3	2	1	0
DDIR	0	HOP5 (ARBEN/ HDBEN)	HOP4 (MSG)	HOP3 (CMD)	HOP2 (I/O OUT)	HOP1 (BSYOUT)	HOP0 (SELOUT)
RESET: 0	0	0	0	0	0	0	0

DDIR — Data Register Direction Bit
1 = The information in bits 0-7 of the host-interface data register and bit DBP of the host-interface data parity register is driven onto the HDB0–HDB7 and HDBP pins, respectively.
0 = The information in bits 0-7 of the host-interface data register and bit DBP of the host-interface data parity register reflects the levels on the HDB0–HDB7 and HDBP pins, respectively.

Bit 6 — Not Used.
Unused bits in the HPCR always read as zeros.

HOP5 — Host-Interface Output Control Bit 5
1 = Pin HOP5(ARB/HDBEN) is forced to a logic high level.
0 = Pin HOP5(ARB/HDBEN) is forced to a logic low level.

HOP4 — Host-Interface Output Control Bit 4
1 = Pin HOP4(MSG) is forced to a logic high level.
0 = Pin HOP4(MSG) is forced to a logic low level.
HOP3 — Host-Interface Output Control Bit 3
1 = Pin HOP3(CMD) is forced to a logic high level.
0 = Pin HOP3(CMD) is forced to a logic low level.
HOP2 — Host-Interface Output Control Bit 2
1 = Pin HOP2(I/O_OUT) is forced to a logic high level.
0 = Pin HOP2(I/O_OUT) is forced to a logic low level.
HOP1 — Host-Interface Output Control Bit 1
1 = Pin HOP1(BSYOUT) is forced to a logic high level.
0 = Pin HOP1(BSYOUT) is forced to a logic low level.
HOP0 — Host-Interface Output Control Bit 0
1 = Pin HOP0(SELOUT) is forced to a logic high level.
0 = Pin HOP0(SELOUT) is forced to a logic low level.

HOST STATUS REGISTER (HSR) $0006

The HSR provides the status of the host input port pins (HIP0-HIP3).

7	6	5	4	3	2	1	0
LOST	ARB	SELPH	HIP3E	HIP3	HIP2	HIP1	HIP0
			(ATNE)	(ATN)	(I/O_IN)	(BSYIN)	(SELIN)
RESET:							
0	0	0	0	U	U	U	U

LOST — Arbitration Lost Status Bit
This bit denotes the failure of a host-interface arbitration and reselection operation due to successful arbitration by a faster or higher priority SCSI device.

LOST should be used only when the HDC is an SCSI controller.
1 = Arbitration loss occured.
0 = Arbitration loss did not occur.
ARB — Arbitration Begun Bit
ARB should be used only when the HDC is an SCSI controller.
1 = Detection of an SCSI bus free phase by the host-interface arbitration and reselection operation drives the ARB/SDBEN and BSYOUT pins to logic high levels.
0 = Bus free phase not detected.
SELPH = Selection Phase Detected Bit
SELPH should be used only when the HDC is an SCSI controller.
1 = Host interface detects an SCSI selection phase.
0 = No SCSI selection phase detected.
HIP3E — Host Input Port Bit 3 Edge Detected Bit
1 = Host interface hardware detects a rising edge on the HIP3/MODE1(ATN/MODE1) pin.
0 = No rising edge detected on the HIP3 MODE1(ATN/MODE1) pin.
HIP3 — Host Input Port Bit 3
HIP3 (ATN) is a read-only status bit.
1 = Pin HIP3/MODE1 (ATN/MODE1) is at a logic high level.
0 = Pin HIP3/MODE1 (ATN/MODE1) is at a logic low level.
HIP2 — Host Input Port Bit 2
HIP2(I/O IN) is a read-only status bit.
1 = Pin HIP2/MODE2(I/O IN/MODE2) is at a logic high level.
0 = Pin HIP2/MODE2(I/O IN/MODE2) is at a logic low level.
HIP1 — Host Input Port Bit 1
HIP1(BSYIN) is a read-only status bit.
1 = Pin HIP1(BSYIN) is at a logic high level.
0 = Pin HIP1(BSYIN) is at a logic low level.
HIP0 — Host Input Port Bit 0
HIP0(SELIN) is a read-only status bit.
1 = Pin HIP0(SELIN) is at a logic high level.
0 = Pin HIP0(SELIN) is at a logic low level.

The LOST, ARB, SELPH, and HIP3E(ATNE) bits are cleared by a read of the HSR followed by a write of zero to the desired bit(s).

HOST DATA REGISTER (HDR) $0007

The HDR provides the means to place data on the HDB0–HDB7 pins and to read data off the HDB0–HDB7 pins. The data direction bit (DDIR) in the host pin control register (HPCR) determines whether the data in the HDR is driven on the HDB0–HDB7 pins or whether the data in the HDR reflects the levels on the HDB0–HDB7 pins. Bit 7 in the HDR corresponds to HDB7 and so on. A one in any bit indicates the level on the corresponding pin is a logic high, while a zero in any bit indicates the level on the corresponding pin is a logic low.

HOST DATA PARITY REGISTER (HDPR) $0008

The HDPR provides the means to place the correct parity on the HDBP pin in conjunction with the data in the HDR. Likewise, it can be used to monitor the level on the HDBP pin. It also provides information on whether or not parity is used on the host data bus, and it can store an encoded device ID.

7	6	5	4	3	2	1	0
DBP	HDBPEN	0	0	0	ID2	ID1	ID0
RESET:							
U	HDBP	0	0	0	HDB2	HDB1	HDB0

DBP — Data Bus Parity Bit
The DDIR bit in the HPCR determines whether the logic level on the DBP bit is driven onto the HDBP pin or reflects the logic level on the HDBP pin.
1 = HDBP is high.
0 = HDBP is low.
HDBPEN — Host Data Bus Parity Option Bit
This read-only bit reflects the value of the HDBP pin at the rising edge of $\overline{\text{RESET}}$.
1 = Host data bus parity is enabled.
0 = Host data bus parity is disabled.
Bits 5–3 — Not used.
These bits always read as zero.
ID2–0 — SCSI Bus Device Identification Bits
These are read-only bits that reflect the state of the HDB2–0 pins at the rising edge of $\overline{\text{RESET}}$.

HDC SCSI OPERATIONS

The following paragraphs provide information on how to use the HDC host-interface as an SCSI target. The HDC host-interface is capable of disconnecting and reconnecting and of detecting the SCSI selection, bus free, and attention phases. It can also perform the initial steps of the SCSI bus arbitration and reselection phases. The host interface provides the means:

1) To determine the levels of the SCSI BSY, SEL, I/O, and ATN signals and
2) To drive the SCSI BSY, SEL, I/O, CMD, and MSG signals.

CONTROLLING THE SCSI BUS

Once selected by an SCSI host, the HDC, as an SCSI target, controls the various types of information transferred on the SCSI data bus. The SCSI standard supports six information phases as follows:

Data out (from the host)	Data in (to the host)
Command out (from the host)	Status in (to the host)
Message out (from the host)	Message in (to the host)

The HDC host interface selects the information phase with the arbitration enabled/host data bus enabled (ARB/HDBEN), CMD, MSG, and I/O OUT signals as shown in Table 1. When the HDC is not communicating with a host, the APB/HDBEN, CMD, MSG, and I/O_OUT signals should be driven to a logic low level. All HDC host-interface signals are active high and should be inverted by the external hardware that drives the SCSI bus.

Table 1. SCSI Bus Phase Control

ARB/HDBEN	CMD	MSG	I/O_OUT	SCSI BUS PHASE
0	0	0	0	Data Out
0	1	0	0	Command
0	1	1	0	Message Out
1	0	0	1	Data In
1	1	0	1	Status
1	1	1	1	Message In
1	0	0	0	*

NOTE:
1 = Logic High
0 = Logic Low

*When ARB/HDBEN is 1 and I/O_Out is 0, external hardware drives the HDC SCSI bus device ID onto the SCSI data bus.

All data is transferred between the HDC and the host using the HDC data buffers. Using the corresponding control bits in the HPCR, the ARB/HDBEN, CMD, MSG, and I/O OUT signals are driven to select the desired data phase. Then, the data buffers transfer the data.

SERIAL DATA CONTROLLER

The serial data controller provides a high-speed connection between the HDC and a disk drive. This controller is a programmable unit that can work with several driven interfaces, such as ST-506, ESDI, and storage module device (SMD) interfaces, without CPU intervention. Operations performed are read and write standard, read and write format, read and write long, and data search.

The serial data controller consists of six registers through which the CPU can control serial operations. Three of these registers program the serial data operation, while the other three establish format parameters.

SERIAL UNIT CONFIGURATION REGISTER (SUCR) $0030

The SUCR selects the disk interface type.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FIXED	ESDI
RESET:							
0	0	0	0	0	0	0	0

SUCR 7–2 — Not used.
Unused bits in the SUCR always read as zeros.

FIXED — Disk Sector Type Bit
The FIXED bit should not be set when the ESDI bit is cleared.
1 = Hard-sectored
0 = Soft-sectored

ESDI — Disk Data Format Type Bit
1 = ESDI compatible disk
0 = ST-506 compatible disk

SERIAL UNIT FUNCTION REGISTER $0031

The serial unit function register (SUFR) selects and starts a serial data operation.

7	6	5	4	3	2	1	0
START	ABORT	INT EN	ECC	DIS-CHCK	FOR-MAT	RD/WRT	SEARCH
RESET:							
0	0	0	0	0	0	0	0

START — Start Operation Bit
The START bit always reads as a zero.
1 = The serial data controller begins the operation defined by the ECC, DISCHCK, FORMAT, RD/WRT, and SEARCH bits.
0 = Immediately after START is set to one, the serial data controller resets START to zero.

ABORT — Abort Operation Bit
Usually, ABORT will read as a zero.
1 = The serial data controller aborts the current operation within one sector time cycle, and performs an orderly shutdown of all serial control signals.
0 = Immediately after the ABORT bit is set to one, the serial data controller resets it to zero. The time required by the controller to clear ABORT depends on the frequency of the read/reference clock signal from the disk.

INT EN — Interrupt Enable Bit
1 = The serial data controller interrupt is enabled.
0 = The serial data controller interrupt is disabled.

ECC — Error Checking and Correction Bit
- 1 = When this bit is set, the serial data controller performs the read or write operation without using the ECC unit. On a read operation, the data field is not checked for data errors. This function allows a host computer to manage the ECC.
- 0 = (Default) When this bit is clear, the serial data controller performs the read or write operation using the ECC unit. Check bytes from the ECC unit are written on the disk immediately following the data field when a write operation is executed. When a read operation is executed, the ECC unit checks for data errors.

DISCHCK — Disable Header ECC Check Bit
- 1 = The address field of each sector is not checked for errors.
- 0 = (Default) The address field of each sector is checked for errors.

FORMAT — Selects a Format Operation Bit
- 1 = Selects a format track operation
- 0 = Does not select a format track operation.

RD/WRT — Selects a Read or Write Operation Bit
- 1 = Selects a read operation
- 0 = Selects a write operation

SEARCH — Selects a Search Operation Bit
- 1 = Selects a data search operation
- 0 = Does not select a data search operation

The serial data controller supports eight legal combinations of the ECC, DISCHCK, FORMAT, RD/WRT, and SEARCH bits. If an illegal option is selected, the serial data controller will operate erratically and can destroy data on the disk. Table 2 is a list of legal values in the various types of serial data controller operations.

Table 2. List of Legal Values

Serial Data Controller Operation
0 0 0 0 0 = Write Standard
0 0 0 1 0 = Read Standard
0 0 1 0 0 = Write Format
0 0 1 1 0 = Read Format
1 0 0 0 0 = Write with ECC
1 0 0 1 0 = Read with ECC
0 0 0 1 1 = Search
1 1 0 1 0 = Read with ECC on Header Disabled
All Others = Illegal Operation

SERIAL UNIT STATUS REGISTER (SUSR) $0032

The SUSR indicates the status of the serial data operation.

7	6	5	4	3	2	1	0
ACT	DONE	NO-MATCH	SKERR	FLGERR	IXERR	ECCERR	DSNF
RESET: 0	0	0	0	0	0	0	0

ACT — Active Bit
One of the other status bits indicates whether or not it was a normal operation termination. ACT is a read-only status bit.
- 1 = The serial data controller operation is in progress.
- 0 = The operation terminated.

DONE — Operation Complete Bit
DONE is not valid while ACT is one.
- 1 = The serial data controller operation completed without error or was aborted by the ABORT bit in the SUFR.
- 0 = Error termination.

NOMATCH — No Match Bit
NOMATCH is not valid while ACT is one.
- 1 = The data search operation terminated because of a data miscompare.
- 0 = Data miscompare did not cause error termination.

SKERR — Seek Error Bit
SKERR is not valid while ACT is one.
- 1 = The serial data controller operation terminated because of a cylinder or head miscompare during an address field search.
- 0 = A cylinder or head miscompare did not cause error termination.

FLGERR — Flag Error Bit
FLGERR is not valid while ACT is one.
- 1 = The operation terminated because of a flag miscompare detected on the current track.
- 0 = A flag miscompare did not cause error termination.

IXERR — Index Error Bit
IXERR is not valid while ACT is one.
- 1 = Two index pulses were detected since the operation started without finding the indicated sector address. During a write format operation, format information was still being written when INDEX was detected, indicating the field lengths in the length RAM are too long.
- 0 = Index error did not cause error termination.

ECCERR — Error Checking and Correction Error Bit
ECCERR is not valid while ACT is one.
- 1 = A data field ECC error occurred.
- 0 = A data field error did not cause an error termination.

DSNF — Data Sync Not Found Bit
DSNF is valid only for ST-506 disks. DSNF is not valid when ACT is one.
- 1 = A data sync byte was never found during the operation.
- 0 = Data sync not found did not cause an error termination.

The DONE, NOMATCH, SKERR, FLGERR, IXERR, ECCERR, and IDSNF bits are cleared by a read of the SUSR followed by a write of zero to the desired bit(s).

SERIAL UNIT ADDRESS COUNTER REGISTER (SUAC)

The SUAC is a 5-bit register that points to two bytes: one in the value RAM and one in the length RAM. When

a serial data operation is in progress, the SUAC register is under the control of the serial data controller and should not be accessed.

SERIAL UNIT VALUE RAM DATA REGISTER (SUVR)

The SUVR accesses the data in the value RAM.

SERIAL UNIT LENGTH RAM DATA REGISTER (SULR)

The SULR accesses the data in the length RAM.

VALUE AND LENGTH RAM ARRAYS

The serial data controller uses two arrays of RAM to define the length and data content of each field in one sector. The first array is the length RAM. The second array is the value RAM. A byte in the length RAM corresponds to a byte in the value RAM. Each RAM array contains 24 bytes, which are not directly mapped in the CPU's address space. As a result, the length and value RAM arrays are accessed via the SULR and SUVR, respectively. Each register is a window providing access to the corresponding bytes addressed by the SUAC.

During a serial data controller write format operation, the information in a value RAM byte is duplicated on the disk the number of times specified by the corresponding length RAM byte. Normal serial-data-controller read/write operations use the information in the value and length RAMS to assert/negate the serial-data-controller signals ($\overline{RD}$ $\overline{GATE}$, $\overline{WRT}$ $\overline{GATE}$, etc.). The serial-data-controller takes these steps to locate the data fields within the sectors on a track.

The value and length RAM arrays must be initialized by the CPU in the HDC startup procedure. Once the arrays are initialized, only the cylinder, head, sector, and flag fields need to be changed prior to starting each serial data operation.

BUFFER UNIT

The data buffer controller provides a high-speed connection between the HDC and a host computer using the host interface data bus. In a standard read operation, parallel data from the serial data controller is shipped through the data buffers to the host computer. A standard write operation ships data from the host through the data buffer to the serial data controller. When performing a standard read or write operation, the data buffer controller fills one data buffer while the other is being emptied. The data buffers are also used to send or receive command and status information between the HDC and a host computer and to send format information and verification patterns to the disk.

The data buffer controller consists of eight registers, through which the CPU can control data operations, and two data buffers. Two registers are used to program the operation while the remaining registers address and access the data buffers.

BUFFER GENERAL-CONTROL REGISTER (BGCR) $0020

The BGCR is used to select and start a data buffer operation.

7	6	5	4	3	2	1	0
START	ABORT	DNINT	ERRINT	DEFALT	XFER1	XFER0	DIR
RESET:							
0	0	0	0	0	0	0	0

START — Start Operation Bit
The START bit always reads logic zero.
1 = The buffer controller begins the operation defined by the XFER1, XFER0, and DIR bits.
0 = Immediately after START is set to one, the data buffer controller resets START to zero.

ABORT — Abort Operation Bit
The ABORT bit always reads logic zero.
1 = The data buffer controller aborts the current operation as soon as the byte transfer to/from the host and/or the byte transfer to/from the serial data controller is finished. The data buffer controller negates REQ and drives HDB0–HDB7 and HDBP to a quiescent state.
0 = Immediately after ABORT is set to one, the data buffer controller resets ABORT to zero.

DNINT — Done Interrupt Bit
1 = The buffer done interrupt is enabled.
0 = The buffer done interrupt is disabled.

ERRINT — Error Interrupt Bit
1 = The buffer error interrupt is enabled.
0 = The buffer error interrupt is disabled.

DEFALT — Default Bit
The default buffer is the first buffer to receive or send data.
1 = The programmed operation uses BUFFER2.
0 = The programmed operation uses BUFFER1.

XFER0–XFER1 — Transfer Zero–Transfer One Bits
These bits contain a binary number that selects the data buffer controller operation.

DIR — Direction Bit
1 = Direction of data flow through the data buffer is from controller to host.
0 = Direction of data flow through the data buffer is from host to serial data controller.

In this register, the DEFALT, XFER1, XFER0, and DIR bits are write protected when the ACT bit of the BUSR is logic one. Table 3 lists the data buffer controller transfer types and their coding.

Table 3. Transfer Types and Coding

Transfer Type	
0 0 0	= Write Disk
0 0 1	= Read Disk
0 1 0	= Read Host
0 1 1	= Write Host
1 0 0	= Write Format
1 0 1	= Read Format
1 1 0	= Write Standard
1 1 1	= Read Standard

BUFFER STATUS REGISTER (BUSR) $0021

The BUSR provides the status of the data buffer operation.

7	6	5	4	3	2	1	0
ACT	DONE	HPRTY1	DPRTY1	HPRTY2	DPRTY2	CNTERR	BUFX
RESET:							
0	0	0	0	0	0	0	0

ACT — Operation Active Bit
ACT is a read-only status bit.
1 = A data bufer controller operation is in progress.
0 = The operation has terminated.
The other BUSR status bits indicate whether the termination was a normal or error termination.

DONE — Operation Done Bit
DONE is not valid while ACT is one.
1 = The operation completed without error or was aborted by setting the ABORT bit in the BGCR.
0 = The operation completed with error.

HPRTY1 — Parity Error Between Host and BUFFER1 Bit
HPRTY1 is not valid while ACT is one.
1 = The data buffer operation on BUFFER1 terminated because of a parity error on the host-interface data bus. If parity is not enabled on the host-interface data bus, then HPRTY1 is set to one only for a parity error on data transferred from BUFFER1 to the host computer.
0 = No parity error existed on data transferred between BUFFER1 and the host.

DPRTY1 — Parity Error Between Disk and BUFFER1 Bit
DPRTY1 is not valid while ACT is one.
1 = The data buffer operation on BUFFER1 terminated because of a parity error on data transferred from BUFFER1 to the serial data controller.
0 = No parity error existed on data transferred between BUFFER1 and the serial data controller.

HPRTY2 — Parity Error Between Host and BUFFER2 Bit
HPRTY2 is not valid while ACT is one.
1 = The data buffer operation on BUFFER2 terminated because of a parity error on the host interface data bus. If parity is not enabled on the host-interface data bus, then HPRTY2 is set to one only for a parity error on data transferred from BUFFER2 to the host computer.
0 = No parity error existed on data transferred between BUFFER2 and the host.

DPRTY2 — Parity Error Between Disk and BUFFER2 Bit
DPRTY2 is not valid while ACT is one.
1 = The data buffer operation on BUFFER2 terminated because of a parity error on data transferred from BUFFER2 to the serial data controller.
0 = No parity error existed on data transferred between BUFFER2 and disk.

CNTERR — Count Error Bit
CNTERR is not valid while ACT is one.
1 = The data buffer controller sent or received more bytes than the serial data controller requested.
0 = No count error occurred.

BUFX — Buffer Last Connected to Disk Bit
This information is used when an error occurred during a standard operation, and the data is to be corrected.
BUFX is a read-only status bit.
1 = The last buffer to transfer data to or from the serial data controller was BUFFER 2.
0 = The last buffer to transfer data to or from the serial data controller was BUFFER 1.

The DONE, HPRTY1, DPRTY1, HPRTY2, DPRTY2, and CNTERR bits are cleared by a read of the BUSR followed by a write to the desired bit(s).

BUFFER ADDRESS REGISTERS (BAR1, BAR2)

The buffer address registers (BAR1 and BAR2) are 16-bit registers used to address a byte in the corresponding data bufer. BAR1 and BAR2 are decremented by the data buffer controller when the corresponding data buffer is accessed. When a data buffer operation is in progress, BAR1 and BAR2 are controlled by the data buffer controller and should not be accessed.

The four most significant bits of both BAR1 and BAR2 are not used and always read as zeros. The address of the most significant byte (MSB) of BAR1 is $0025, and the address of the least significant byte (LSB) of BAR1 is $0026. The address of the MSB of BAR2 is $0027, and the address of the LSB of BAR2 is $0028.

BUFFER DATA REGISTERS (BDR1, BDR2)

Data in BUFFER1 and BUFFER2 is accessed via BDR1 and BDR2, respectively. When a data buffer operation is in progress, BDR1 and BDR2 are controlled by the data buffer controller and should not be accessed. The address of BDR1 is $0029, and the address of BDR2 is $002A.

BUFFER ADDRESS PRESET REGISTER (BAPR)

The BAPR is a 16-bit register used to load a preset address into BAR1 and/or BAR2 during a data buffer operation. The value in this register addresses the first byte in either buffer to send or receive data. The BAPR must be initialized prior to starting the data buffer operation. For example, when a 10-byte command packet is sent from the host computer into BUFFER1, BAPR is set to 9. After the data buffer operation, the first byte of the command is in BUFFER1 at address $009, and the last byte is at address $000. For a multiblock disk read with a block size of 256 bytes, the BAPR is set to $0FF (255). When a data buffer operation is in progress, BAPR is controlled by the buffer data controller and should not be accessed. The four most significant bits of the BAPR are not used and always read as zeros. The address of the MSB of the BAPR is $0022, and the address of the LSB is $0023.

TRACK BLOCK COUNTER REGISTER (TBCR)

The TBCR specifies the number of blocks of data to be transferred through the data buffers. For example, when a command packet is to be received from the host, TBCR is set to one. When a multiblock disk read is to be done, TBCR is set to the number of blocks to be sent to the host. During a data buffer operation, TBCR is controlled by the data buffer controller and should not be accessed. The address of the TBCR is $0024.

DATA BUFFERS

The HDC has two data buffers, BUFFER1 and BUFFER2, which are each large enough to hold a 512-byte sector of data plus 16 ECC check byte (a total of 528 bytes). During a standard read or write data operation, only one sector of data is held in a data buffer, and the two data buffers cannot be concatenated for larger sector sizes. Because the data buffers are not directly mapped in the CPU's address space, each buffer is accessed via the buffer data registers (BDR1 and BDR2) and the buffer address registers (BAR1 and BAR2). To read or write in BUFFERx, where x is 1 or 2, BARx must be set to the address of the byte ($000 through $20F). BDRx must then be read or written to access the data byte. After BDRx is read or written, BARx is automatically decremented by one. Figure 6 is a diagram of the data buffer.

If BARx is set to a value greater than $20F and BDRx is read or written, then BUFFERx is not accessed. In the case of a read, the value returned is undefined. The flowchart in Figure 7 shows how to access a data buffer.

DATA BUFFER PARITY

Each byte in both data buffers has a corresponding parity bit. When a data byte is written to either data buffer, a parity bit is generated and stored with the byte. When a data byte is read from a buffer, the byte is checked for a parity error. If an error is detected on a byte being sent to a host computer, then either the HPRTY1 or HPRTY2 bit in the BUSR is set, depending on which data buffer sent the byte. If an error is detected on a byte being sent to the serial data controller, then either the DPRTY1 or DPRTY2 bit in the BUSR is set, depending on which data buffer sent the byte.

When a data byte is received from the host computer with parity on the host interface data bus enabled, the byte is checked for a parity error. If an error is detected, then either the HPRTY1 and HPRTY2 bit in the BUSR is set, depending on which data buffer sent the byte.

When a data byte is received from the host computer with parity on the host interface data bus enabled, the byte is checked for a parity error. If an error is detected, then either the HPRTY1 and HPRTY2 bit in the BUSR is set, depending on which data buffer received the byte.

No indication of a parity error when the CPU reads data from the data buffers is given.

ECC UNIT

The serial data controller manages the error checking and correction unit (ECC) to provide Reed-Solomon based error checking and correction without host intervention. When an error is detected on a data field, the CPU interrogates the ECC unit for information to correct erroneous data held in the data buffers. The ECC unit uses interleaving to increase protection for data fields by breaking the data field into smaller logical blocks and protecting each block separately. The interleave factor is selectable between one, two, three, and five. A recommended interleave of two should be used for block sizes of 256 or 128, and an interleave of three is recommended for a block size of 512. The ECC unit interleave should not be confused with the sector interleave on the disk. The ECC unit interleave, which is invisible, does not affect the way data is written on the disk.

The ECC unit consists of two registers through which the CPU can select an interleave factor and obtain correction information in case of error.

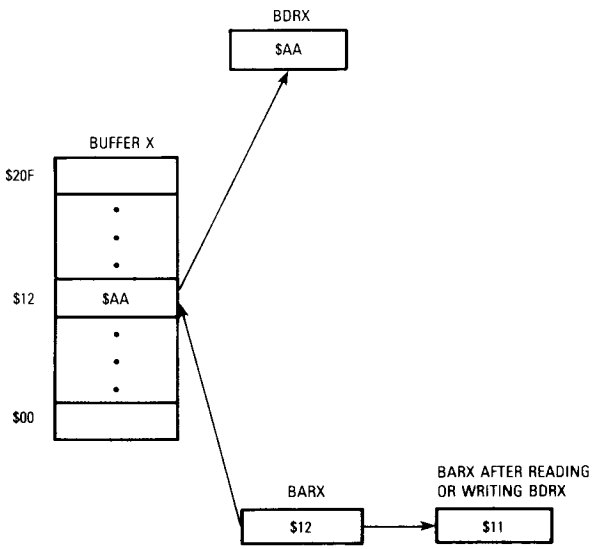


Figure 6. Data Buffer Diagram

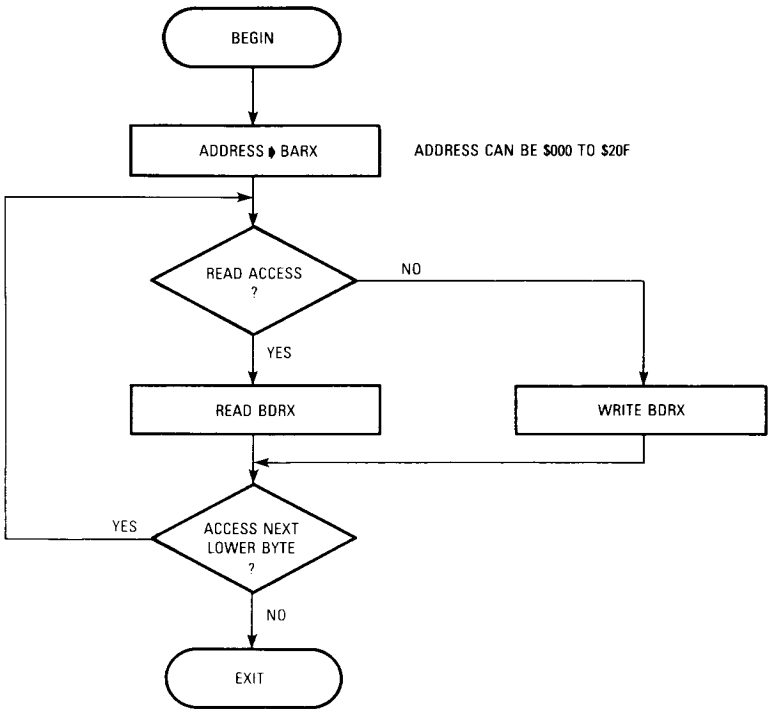


Figure 7. MPU Access of Buffer X

ECC CONTROL REGISTER (ECR) $000C

The ECR selects the interleave and interrogates the ECC unit for correction information if an error is detected.

7	6	5	4	3	2	1	0
EM	IPT	ILV1	ILV0	CS	CP	CD	CR
RESET: U	U	U	U	U	U	U	U

EM — Enable Monitor Bit
This bit is used by Motorola for testing the ECC unit and should not be written.

IP — Increment Pointer Bit
This bit is used during a correction to increment the interleave pointer.

ILV1–0 — Interleave Factor Bits
The ILV1 and ILV0 bits select the ECC interleave factor to be used on a block of data, as shown in Table 4.

Table 4. Interleave Selection

ILV1	ILV0	Interleave Factor
0	0	1
0	1	2
1	0	3
1	0	4

CS — Clear Syndrome Bit
This bit is used by Motorola for testing the ECC unit and should not be written.

CP — Clear Pointer Bit
This bit is used during a correction to clear an internal data pointer.

CD — Clear Data Bit
This bit is used during a correction to clear the ECC data register.

CR — Clear RAM Bit
This bit is used by Motorola for testing the ECC unit and should not be written.

ECC DATA REGISTER (EDCR)

The EDCR provides the CPU with an error byte mask and is used only during a correction.

RESET

The $\overline{\text{RESET}}$ signal resets the CPU core and on-chip devices to provide an orderly startup procedure and to ensure that all peripherals and output signals are in a quiescent state.

CPU AFTER RESET

After reset, if the operating mode is single-chip or expanded mode 1, the CPU fetches its restart vector from internal ROM at $FFFE and $FFFF. The I bit in the condition code register is set to mask interrupt requests.

PORTS B AND C AFTER RESET

After reset, if the operating mode is single-chip, then the port B and C pins are inputs. If the operating mode is one of the expanded modes, then the port B and C pins begin driving the expansion bus.

TIMER AFTER RESET

After reset, the timer counter is initialized to $FFF9. If the operating mode is single-chip, then timer TCAP and TCMP pin functions are disabled. The user can elect to enable these functions by setting the TEN bit of the HPIR.

INTERRUPTS

The HDC supports both maskable and nonmaskable interrupts with a prioritization scheme similar to the MC68HC11.

Maskable interrupts originate only from the on-chip peripheral devices and can be masked by the I bit of the condition code register. The interrupt signals coming from the various devices are prioritized. The highest priority interrupt is selected by changing the value of the PSEL0–3 bits in the highest priority interrupt register (HPIR). The timer interrupt defaults to the highest priority after reset.

Nonmaskable interrupts include $\overline{\text{RESET}}$, illegal opcode trap, and software interrupt (SWI). $\overline{\text{RESET}}$ has the highest priority, followed by illegal opcode trap. SWI is actually an instruction having the highest priority after $\overline{\text{RESET}}$. This statement is essentially true since, once the SWI opcode has been fetched, no other interrupt will be honored until after the SWI vector is fetched.

Table 5 shows the interrupt sources and their corresponding vectors. Each vector requires two bytes. The first vector byte contains the eight most significant bits of vector address; the second byte contains the eight least significant bits of vector address. Figure 8 shows the interrupt handler flowchart.

Table 5. Interrupt Vectors

Vector Address	Interrupt Source	Masked By
XXC0, XXC1 To XXE4, XXE5	Reserved	
XXE6, XXE7	Serial Unit Termination	I Bit
XXE8, XXE9 XXEA, XXEB	Buffer Done Buffer Error	I Bit I Bit
XXEC, XXED XXEE, XXEF XXF0, XXF1	SCSI Attention SCSI Arbitration Begun SCSI Selection Phase	I Bit I Bit I Bit
XXF2, XXF3	Timer	I Bit
XXF4, XXF5 XXF6, XXF7 XXF8, XXF9 XXFA, XXFB XXFC, XXFD XXFE, XXFF	Reserved SWI Illegal Opcode Trap Reserved Reserved Reset	 None None None

NOTE:
XX = FF if mode is single-chip or expanded 1
XX = BF if mode is expanded 2 or expanded 3

HIGHEST PRIORITY INTERRUPT REGISTER (HPIR) $003F

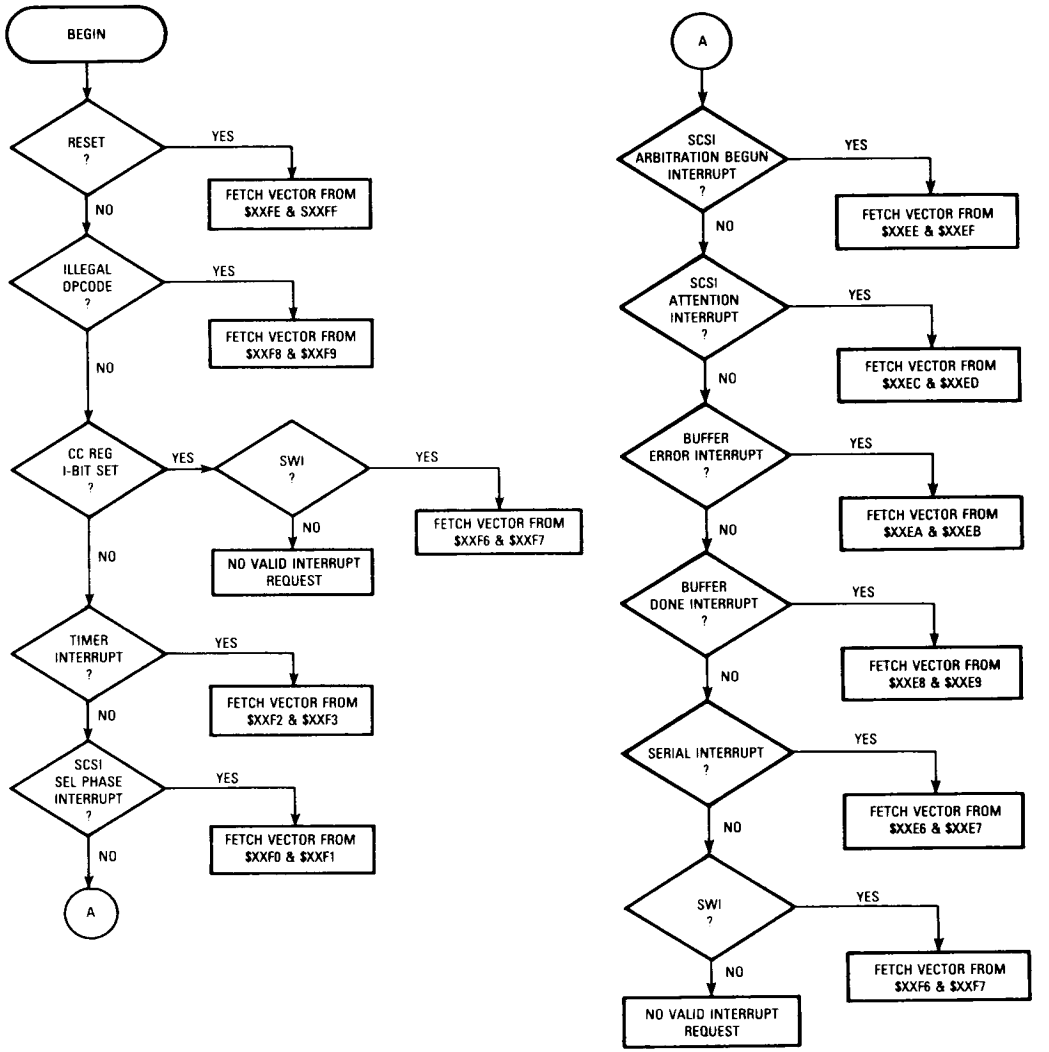
The HPIR is an 8-bit read/write register that contains seven control bits and one read-only status bit.

7	6	5	4	3	2	1	0
TEN	REM-ROM	IRV	MODE	PSEL3	PSEL2	PSEL1	PSEL0
RESET:							
0	0	0	R	0	1	0	1

TEN — TCAP/TCMP Enable Bit
In single-chip mode:
1 = TCAP and TCMP are enabled on port B pins 6 and 7, respectively.
0 = TCAP and TCMP are disabled, and port B pins 6 and 7 are dedicated to general-purpose I/O.
In expanded modes:
TEN is a read-only bit that always reads logic zero.

REMROM — Remove Internal ROM Bit
In expanded 2 and 3 modes:
1 = Internal ROM is disabled, addresses from $D000 to $FFFF must be accessed externally.
0 = Internal ROM is enabled.
In expanded 1 and single-chip modes:
REMROM is a read-only bit that always reads logic zero.

IRV — Internal Read View Bit
In single-chip mode:
IRV is a read-only bit that always reads logic zero.
In expanded modes:
Once set, IRV can only be cleared by reset.
1 = Data read from internal registers, ROM, and RAM is driven onto the external multiplexed address/data bus.
0 = Data is not driven onto external bus.



NOTE:
XX is FF if mode is single-chip or expanded 1;
XX is BF if mode is expanded 2 or expanded 3.

Figure 8. Interrupt Handler Flowchart

MODE — Read-Only Status Bit Indicating Current Operating Mode
1 = Expanded 2 or expanded 3 mode
0 = Single-chip or expanded 1 mode
PSEL3–0 — Priority Select
These bits contain a binary number indicating the I-interrupt source with the highest priority. This binary code and its meaning are shown in Table 6.

Table 6. Interrupt Source and Priority

0110 — Timer	1010 — Buffer Error
0111 — SCSI Selection	1011 — Buffer Done
1000 — SCSI Arbitration	1100 — Serial Unit
1001 — SCSI Attention	

All other codes are reserved.

LOW POWER MODE: WAIT

For low power operation, this HDC implements the WAIT instruction. WAIT causes the CPU core of the HDC to assume a low power state, or wait mode, with the internal clocks still active.

In the wait mode, the machine state is stacked and program execution stops. The serial data controller, ECC unit, data buffer unit, host interface and timer are not affected. $\overline{\text{RESET}}$ and the I interrupt cause an exit from the wait mode if the I bit in the condition code register is clear.

TIMER

Preceded by a fixed divide-by-four prescaler, the programmable timer can be used for many purposes since its pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 9.

Because the timer has a 16-bit architecture, each specific functional segment, or capability, is represented by two 8-bit registers. These registers contain the high or low byte of that functional segment and are called the high-byte register and low-byte register, respectively. Generally, accessing the low-byte register of a specific timer function allows full control of that function. However, an access of the high-byte register inhibits that specific timer function until the low-byte register is also accessed.

NOTE

The I bit in the condition code register should be set while manipulating both the high-byte and the low-byte registers of a specific timer function to ensure that an interrupt does not occur. Setting this bit prevents interrupts from occurring between the time that the high- and low-byte registers are accessed.

TIMER CONTROL REGISTER (TCR) $0012

The TCR is an 8-bit read/write register which contains five control bits. Three of these bits, ICIE, OCIE, and TOIE, control the interrupts associated with each of the three flag bits found in the timer status register (TSR): ICF, OCF, and TOF, respectively. The IEDG bit controls whether the negative or positive edge of the signal is significant to the input capture edge detector. The OLVL bit controls the next value to be clocked to the output level register in response to a successful output compare.

The TCR and the free-running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and remains low until raised high by a valid compare. The TCR is shown below:

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET:							
0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable Bit
1 = The timer interrupt is enabled when the input capture flag (ICF) status flag is set.
0 = Interrupt is inhibited.
OCIE — Output Compare Interrupt Enable Bit
1 = The timer interrupt is enabled when the output compare flag (OCF) status flag is set.
0 = Interrupt is inhibited.
TOIE — Timer Overflow Interrupt Enable Bit
1 = The timer interrupt is enabled when the timer overflow flag (TOF) status flag is set.
0 = Interrupt is inhibited.
Bits 4–2 — Not Used.
IEDG — Input Edge Bit
The input edge bit determines which level transition will trigger a free-running counter transfer to the input capture register.
1 = Positive edge
0 = Negative edge
OLVL — Output Level Bit
The output level bit is clocked into the output level register by the next successful output compare. This bit and the output level register are cleared by reset.
1 = High output
0 = Low output

TIMER STATUS REGISTER (TSR) $0013

The TSR is an 8-bit register of which the three most significant bits contain read-only status information. These three bits indicate the following:
— A proper transition has taken place with an accompanying transfer of the free-running counter contents to the input capture register.
— A match has been found between the free-runnign counter and the output compare register.
— A free-running counter transition from $FFFF to $0000 has been sensed (timer overflow).

7	6	5	4	3	2	1	0
ICF	OCF	TOF	0	0	0	0	0
RESET:							
U	U	U	0	0	0	0	0

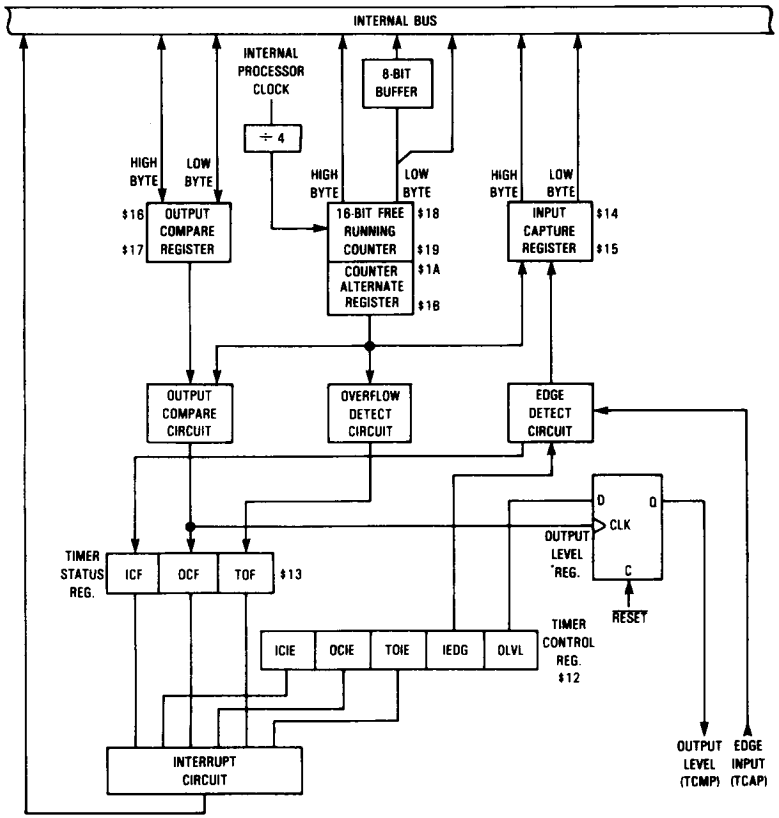


Figure 9. Timer Block Diagram

ICF — Input Capture Flag
- 1 = A proper edge has been sensed by the input capture edge detector.
- 0 = The flag is cleared by a processor access of the TSR (with the ICF set) followed by accessing the low byte ($15) of the input capture register.

OCF — Output Compare Flag
- 1 = The output compare register contents match the contents of the free-running counter.
- 0 = The flag is cleared by accessing the timer status register (with OCF set) and then accessing the low byte ($17) of the output compare register.

TOF — Timer Overflow Flag
- 1 = A transition of the free-running counter from $FFFF to $0000 occurred.
- 0 = The flag is cleared by accessing the timer status register (with TOF set) and then accessing the low byte of the free-running counter $19).

Bits 4–0 — Not used.

Accessing the TSR satisfies the first step required to clear any status bits that happen to be set during the access. The next step is to access the register associated with the status bit. Typically, this procedure is accomplished using the input capture and output compare functions.

A problem can occur when the user is using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the TOF could unintentionally be cleared if:

1) The TSR is read or written when TOF is set and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

Since the counter alternate register contains the same value as the counter, this alternate register can be read at any time without affecting the TOF.

INPUT CAPTURE REGISTER (ICR)

The two 8-bit registers, which comprise the 16-bit ICR, are read-only registers. After the corresponding input-capture edge detector senses a defined transition, these registers are used to latch the value of the free-running counter. The level transition that triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the ICR.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal processor clock preceding the external transition. This delay is required for internal synchronization. Resolution is affected by the prescaler allowing the timer to increment only every four internal processor clock cycles.

The free-running counter contents are transferred to the ICR on each proper signal transition regardless of whether the ICF is set or clear. The ICR always contains the free-running counter value that corresponds to the most recent input capture.

After reading the ICRs MSB ($14), counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the input-capture software routine and its interaction with the main program to determine the minimum pulse period attainable. The free-running counter increments every four internal processor clock cycles due to the prescaler.

A read of the LSB ($15) of the ICR does not inhibit the running-counter transfer since the read and the transfer occur on opposite edges of the internal processor clock. Minimum pulse periods are ones which allow software to read the LSB ($15) and perform needed operations.

OUTPUT COMPARE REGISTER (OCR)

The OCR is a 16-bit register, which is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The OCR can be used for such pruposes as indicating when a period of time has elapsed. The OCR is unique in that all bits are readable and writable and are not altered by the timer hardware. Reset does not affect the contents of this register. If the compare function is not needed, the two bytes of the OCR can be used as storage locations.

OCR contents are compared with the contents of the free-running counter every four internal processor clocks. If a match is found, the corresponding OCF bit of the TCR is set, and the corresponding OLVL bit is clocked to an output level register by the output compare circuit pulse. The values in the OCR and the OLVL should be changed after each successful comparison in order to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the OCR containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal processor clock cycles due to the internal prescaler. The minimum time required to update the OCR is a function of the program rather than the internal hardware.

A processor write may be made to either byte of the OCR without affecting the other byte. The OLVL is clocked to the output level register regardless of whether the OCF is set or clear.

Because neither the OCF nor OCR is affected by reset, care must be exercised when initializing the output compare function. The following procedure is recommended:

1) Write to the high byte of the OCR to inhibit further compares until the low byte is written,
2) Read the timer status register to arm the OCF if it is already set, and
3) Write to the low byte of the OCR to enable the output compare function with the flag clear.

This procedure is advantageous because it prevents the OCF bit from being set between the time OCF is read and OCR is written. A software example is as follows:

```
B7  16  STA  OCMPHI   Inhibit Output Compare
B6  13  LDA  TSTAT    Arm OCF Bit If Set
BF  17  STX  OCMPLD   Ready for Next Compare
```

COUNTER

The key element in the programmable timer is a 16-bit free-running counter, or counter register, preceded by a prescaler which divides the internal procesor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal processor clock is 2.0 megahertz. The counter is clocked to increasing values during the low portion of the internal processor clock. Software can read the counter at any time without affecting the value.

The double-byte free-running counter can be read from either the two locations $18–$19 (called counter register at this location), or $1A–$1B (counter alternate register at this location). A read from only the LSB of the free-running counter ($19, $1B) will receive the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the MSB ($18, $1A), then the read causes the LSB ($19, $1B) to be transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B). This access completes a read sequence of the total counter value. If either the free-running counter or counter alternate register MSB is read, the LSB must also be read in order to complete the sequence.

The free-running counter is configured to $FFF9 during reset and is always a read-only register. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 processor internal clock cycles. When the counter rolls over from $FFFF to $0000, TOF is set. An interrupt can also be enabled when counter rollover occurs by setting TOIE.

INSTRUCTION SET

The CPU of the MC68HC99 HDC is an MC68HC11 microprocessor. In addition to its ability to execute all MC6800 and MC6801 instructions, the HDC has an opcode map with a total of 91 new opcodes. Major func-

3

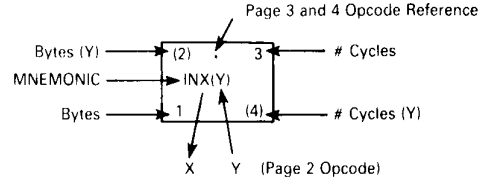
Table 4. Opcode Map

HI / LOW	INH 0 0000	INH 1 0001	REL 2 0010	INH 3 0011	ACCA 4 0100	ACCB 5 0101	(Y) INDX 6 0110	EXT 7 0111	ACCA IMM 8 1000	ACCA DIR 9 1001	ACCA (Y) INDX A 1010	ACCA EXT B 1011	ACCB IMM C 1100	ACCB DIR D 1101	ACCB (Y) INDX E 1110	ACCB EXT F 1111	HI / LOW
0 0000	TEST	SBA	BRA	TSX(Y) (4)	NEGA	NEGB	NEG (7)	NEG	SUBA	SUBA	SUBA (5)	SUBA	SUBB	SUBB	SUBB (5)	SUBB	0
1 0001	NOP	CBA	BRN	INS					CMPA	CMPA	CMPA (5)	CMPA	CMPB	CMPB	CMPB (5)	CMPB	1
2 0010	IDIV	BRSET (8)	BHI	PULA					SBCA	SBCA	SBCA (5)	SBCA	SBCB	SBCB	SBCB (5)	SBCB	2
3 0011	FDIV	BRCLR (8)	BLS	PULB	COMA	COMB	COM (7)	COM	* SUBD	* SUBD	* SUBD (7)	* SUBD	ADDD	ADDD	ADDD (7)	ADDD	3
4 0100	LSRD	BSET (8)	(BHS) BCC	DES	LSRA	LSRB	LSR (7)	LSR	ANDA	ANDA	ANDA (5)	ANDA	ANDB	ANDB	ANDB (5)	ANDB	4
5 0101	(LSLD) ASLD	BCLR (8)	(BLO) BCS	TX(Y)S (4)					BITA	BITA	BITA (5)	BITA	BITB	BITB	BITB (5)	BITB	5
6 0110	TAP	TAB	BNE	PSHA	RORA	RORB	ROR (7)	ROR	LDAA	LDAA	LDAA (5)	LDAA	LDBB	LDBB	LDBB (5)	LDBB	6
7 0111	TPA	TBA	BEQ	PSHB	ASRA	ASRB	ASR (7)	ASR		STAA	STAA (5)	STAA		STBB	STBB (5)	STBB	7
8 1000	INX(Y) (4)	PAGE 2	BVC	PULX(Y) (6)	ASLA	ASLB	ASL (7)	ASL	EORA	EORA	EORA (5)	EORA	EORB	EORB	EORB (5)	EORB	8
9 1001	DEX(Y) (4)	DAA	BVS	RTS	ROLA	ROLB	ROL (7)	ROL	ADCA	ADCA	ADCA (5)	ADCA	ADCB	ADCB	ADCB (5)	ADCB	9
A 1010	CLV	PAGE 3	BPL	ABX(Y) (4)	DECA	DECB	DEC (7)	DEC	ORAA	ORAA	ORAA (5)	ORAA	ORAB	ORAB	ORAB (5)	ORAB	A
B 1011	SEV	ABA	BMI	RTI					ADDA	ADDA	ADDA (5)	ADDA	ADDB	ADDB	ADDB (5)	ADDB	B
C 1100	CLC	BSET (8)	BGE	PSHX(Y) (5)	INCA	INCB	INC (7)	INC	CPX(Y) (6)	CPX(Y) (6)	* CPX(Y) (7)	CPX(Y) (7)	LDD	LDD	LDD (6)	LDD	C
D 1101	SEC	BCLR (8)	BLT	MUL	TSTA	TSTB	TST (7)	TST	BSR	JSR	JSR (7)	JSR	PAGE 4	STD	STD (6)	STD	D
E 1110	CLI	BRSET (8)	BGT	WAI			JMP (4)	JMP	LDS	LDS	LDS (6)	LDS	LDX(Y) (4)	LDX(Y) (5)	* LDX(Y) (6)		E
F 1111	SEI	BRCLR (8)	BLE	SWI	CLRA	CLRB	CLR (7)	CLR	XGDX(Y) (4)	STS	STS (6)	STS		STX(Y) (5)	* STX(Y) (6)		F

INH	Inherent
REL	Relative
IMM	Immediate
EXT	Extended
DIR	Direct
INDX(Y)	Index X(Y)

*Page 3 and 4 Opcode Reference

Mnemonic	Page	Opcode	Bytes	Cycles
CPD	3	83	4	5
	3	93	3	6
	3	B3	4	7
	3	A3	3	7
	4	A3	3	7
CPY	3	AC	3	7
CPX	4	AC	3	7
LDY	3	EE	3	6
LDX	4	EE	3	6
STY	3	EF	3	6
STX	4	EF	3	6



tional additions include a second 16-bit index register, two types of 16-by-16 divide instructions, bit manipulation instructions, and a WAIT instruction.

ADDRESSING MODES

Six addressing modes can be used to reference memory. Some instructions require an additional byte before the opcode to accomodate a multipage opcode map. This byte is called a prebyte. The following paragraphs provide a description of each addressing mode plus a discussion of the prebyte. In these descriptions, the term "effective address" is used to indicate the address in memory from which the argument is fetched or stored or from which execution proceeds.

IMMEDIATE ADDRESSING

In the immediate addressing mode, the actual argument is contained in byte(s) immediately following the instruction, where the number of bytes matches the size of the register. If prebyte is required, then these are two-, three-, or four-byte instructions.

DIRECT ADDRESSING

In the direct addressing mode, the LSB of operand address is contained in a single byte following the opcode, and the MSB is assumed to be $00. Direct addressing allows the user to access $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the HDC, the first 192 bytes of the internal RAM and registers are fixed in page zero of memory, with the exception of the bytes at $0000 through $0003 for the parallel ports. These bytes can be either internal or external depending on the operating mode.

EXTENDED ADDRESSING

In the extended addressing mode, the second and third bytes following the opcode contain the absolute address of the operand. If prebyte is required, then these are three- or four-byte instructions. One or two bytes are required for the opcode and two bytes for the effective address.

INDEXED ADDRESSING

In the indexed addressing mode, one index register (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors:

1) Current contents of the index register (X or Y) being used and
2) The 8-bit unsigned offset contained in the instruction.

This addressing mode allows referencing any memory location in the 64K byte address space. If prebyte is required, then these are usually two- or three-byte instructions, to accomodate the opcode plus an 8-bit offset.

INHERENT ADDRESSING

In the inherent addressing mode, all information to execute an instruction is contained in the opcode. Operands, if any, are registers, and no memory reference is required. These are usually one- or two-byte instructions.

RELATIVE ADDRESSING

The relative addressing mode is used for branch instructions. If the branch condition is true, the contents of the eight-bit signed byte following the opcode, the offset, is added to the contents of the PC to form the effective branch address. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

PREBYTE

In order to expand the number of instructions available in the HDC, a prebyte instruction is used with certain instructions. The instructions affected are usually associated with the index register Y. Opcode instructions which do not require a prebyte could be considered as page 1 of the overall opcode map. The remaining opcodes could be considered as pages 2, 3, and 4 of the opcode map and would require a prebyte: $18 for page 2, $1A for page 3, and $CD for page 4.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to V_{DD} +0.5	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain per pin* Excluding V_{DD}, V_{SS}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance PLCC	θ_{JA}	50	°C/W

3

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 5%; V_{SS} = 0 Vdc; T_A = 0°C to 70°C)

Characteristic	Symbol	Min	Max	Unit
Supply Voltage Range	V_{DD}	4.75	5.25	V
Output High Voltage I_{Load} = 0.8 mA	V_{OH}	2.4	—	V
Output Low Voltage				
SELOUT, BSYOUT, I/O/OUT, CMD, MSG, ARB/SDBEN, REQ, AMEN, WRCLK, NRZ I_{Load} = 1.6 mA	V_{OL}	—	0.4	V
All Other Outputs I_{Load} = 3.7 mA	V_{OL}	—	0.4	V
Input High Voltage	V_{IH}	—	2.0	V
Input Low Voltage	V_{IL}	0.8	—	V
Three-State Leakage $V_{in} = V_{IH}$ or V_{IL} (0.0 − 5.25 V)	I_{OZ}	—	±10	μA
Input Leakage Current	I_{IN}	—	±1	μA
Input Capacitance	C_{IN}	—	15	pF
Power Dissipation	P_D	—	TBD	mW
Total Current Supply				
CPU Only	C_{IDD}	TBD	TBD	V
CPU In Wait Mode	W_{IDD}	TBD	TBD	V
Everything Running	R_{IDD}	TBD	TBD	V

NOTES:
1. No DC loads, V_{IL} = 0.0 V, V_{IH} = 5.0 V, CLK = 10 MHz
2. REFCLK = 0.0 V, serial port and buffer are not active, buffer is refreshing DRAMs.
3. REFCLK = 25 MHz

EXPANSION BUS TIMING (see Figure 10)

Num	Characteristic	Formula		Symbol	Min	Max	Unit
	Frequency of Operation (E Clock Frequency)			f_o	2.5	2.5	MHz
1	Cycle Time			t_{cyc}	400	—	ns
2	Pulse Width, E Low	$1/2\ t_{cyc} - 32$		PW_{EL}	168	—	ns
3	Pulse Width, E High	$1/2\ t_{cyc} - 24$		PW_{EH}	176	—	ns
4	E and AS Rise and Fall Time			t_r, t_f	—	15	ns
9	Address Hold Time	$1\,8\ t_{cyc} - 25$	(a)	t_{AH}	2.5	—	ns
12	Non-Muxed Address Valid Time to E Rise	$PW_{EL}-(t_{ASD} + 78)$	(b)	t_{AV}	60	—	ns
17	Read Data Setup Time			t_{DSR}	50	—	ns
18	Read Data Hold Time	$Max = t_{MAD}$		t_{DHR}	10	60	ns
19	Write Data Delay Time	$1/8\ t_{cyc} + 70$	(a)	t_{DDW}	—	120	ns
21	Write Data Hold Time	$1/8\ t_{cyc} - 30$	(a)	t_{DHW}	20	—	ns
22	Muxed Address Valid Time to E Rise	$PW_{EL} - (t_{ASD} + 78)$	(b)	t_{AVM}	60	—	ns
24	Muxed Address Valid Time to AS Fall	$1\,8\ t_{cyc} - 35$	(a)	t_{ASL}	15	—	ns
25	Muxed Address Hold Time	$1\,8\ t_{cyc} - 25$	(b)	t_{HL}	25	—	ns
26	Delay Time, E to AS Rise	$1\,8\ t_{cyc} - 20$	(a)	t_{ASD}	30	—	ns
27	Pulse Width, AS High	$1\,4\ t_{cyc} - 25$		PW_{ASH}	75	—	ns
28	Delay Time, AS to E Rise	$1\,8\ t_{cyc} - 20$	(b)	t_{ASED}	30	—	ns
29	MPU Address Access Time	$t_{AVM} + t_r + PW_{EH} - t_{DSR}$	(b)	t_{ACCA}	201	—	ns
35	MPU Access Time	$PW_{EH}-t_{DSR}$		t_{ACCE}	—	126	ns
36	Muxed Address Delay (Previous Cycle MPU Read)	$t_{ASD} + 30$	(a)	t_{MAD}	60	—	ns

NOTE:
Input clocks with duty cycles other than 50% affect bus performance. Timing parameters affected by the input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions for $1\,8\ t_{cyc}$ in the formulas the expansion bus timing table where applicable:
(a) $(1\text{-}DC) \times 1/4\ t_{cyc}$
(b) $DC \times 1/4\ t_{cyc}$
where:
DC is the decimal value of the duty cycle percentage, (high time).

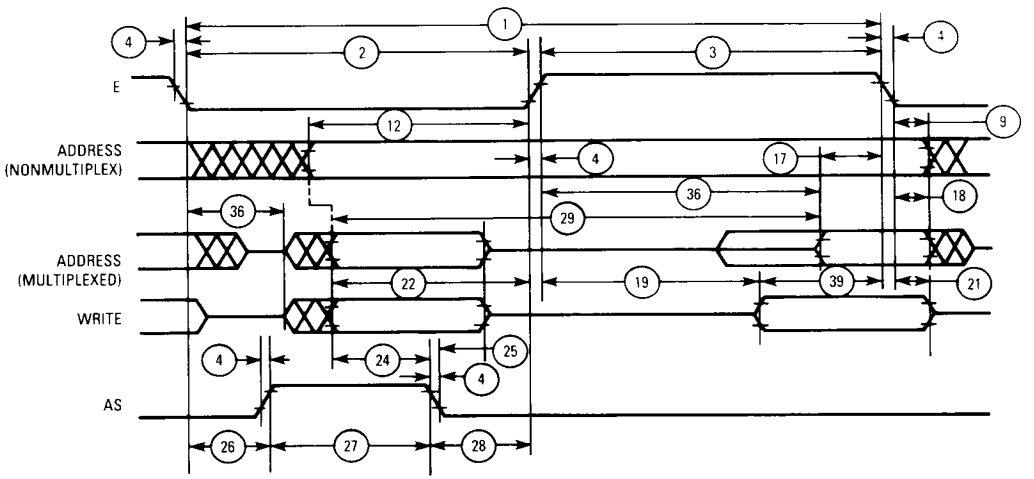
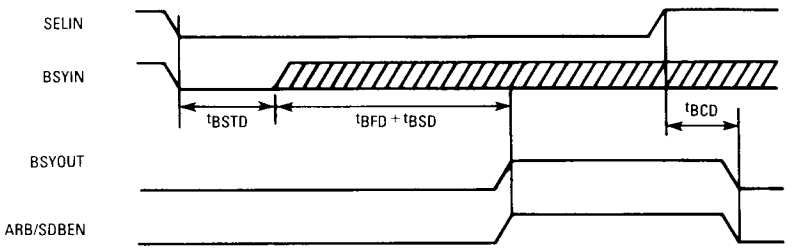


Figure 10. Expansion Bus Timing Diagram

SCSI SELECTION AND RESELECTION TIMING (see Figures 11 and 12)

Characteristic	Symbol	Min	Max	Unit
Selection Phase Detect Time	t_{SPD}	1	1.25	t_{cyc}
Bus Settle Delay (BSYIN and SELIN Negated)	t_{BSTD}	t_{cyc}	—	ns
Bus Free Delay	t_{BFD}	2 t_{cyc}	—	ns
Bus Set Delay	t_{BSD}	1/2	2.0	t_{cyc}
Bus Clear Delay (ARB/SDBEN, BSYOUT Negated)	t_{BCD}	0	1.5 t_{cyc}	ns



NOTE:
The assertion of SELIN during the arbitration phase will asynchronously terminate the arbitration phase and cause the HDC to negate BSYOUT and ARB/SDBEN. The assertion of SELOUT terminates the arbitration phase.

Figure 11. SCSI Arbitration Timing Diagram

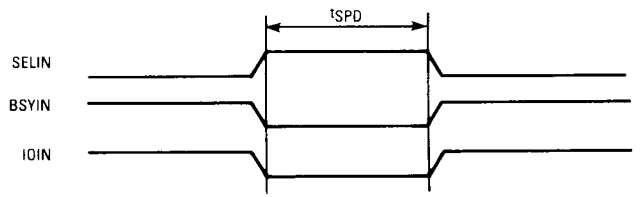


Figure 12. HDC SCSI Selection Phase Timing Diagram

HOST INTERFACE TRANSFER TIMING (see Figures 13 and 14)

Characteristic	Symbol	Min	Max	Unit
Request asserted to ACK Asserted	t_{RAAA}	0	—	ns
ACK Asserted to REQ Negated (Note 1)	t_{AARN}	0	1/8 t_{cyc}	ns
REQ Negated to ACK Negated	t_{RNAN}	0	—	ns
ACK Negated to REQ Asserted (Note 2)	t_{ANRA}	1/2	1.5	t_{cyc}
Data Valid to REQ Asserted (Transmit)	t_{DVR}	55	—	ns
Data Hold from ACK Asserted (Transmit)	t_{DHAA}	0	—	ns
Data Valid to ACK Asserted (Receive)	t_{DVAA}	55	—	ns
REQ Negated to Data Invalid (Hold Time, Receive)	t_{RNDI}	0	—	ns

NOTES:
1. See note (a) in **EXPANSION BUS TIMING** table.
2. Both edges of ACK can affect the host transfer speed. Slow ACK timing causes the transfer speed to be degraded in integer multiples of t_{cyc}.

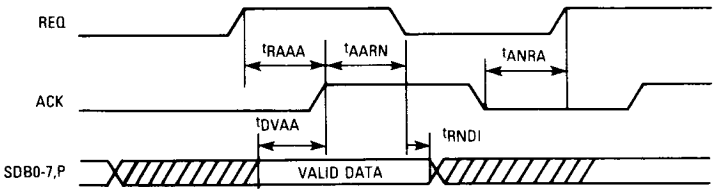


Figure 13. HDC SCSI Receive Timing Diagram

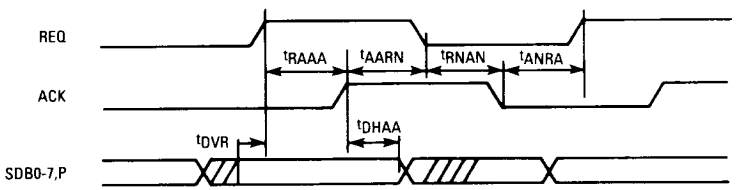


Figure 14. HDC SCSI Transmit Timing Diagram

DISK READ/WRITE TIMING (see Figures 15 and 16)

Parameter	Symbol	10 MHz		16 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
REFCLK Period	t_{RFCP}	100	4000	62	4000	40	4000	ns
REFCLK Pulse Width (50 ± 9.2% Duty Cycle)	PW_{RCL}	41	59	25	37	16	24	ns
REFCLK Rise/Fall Time	t_{RCRF}	—	10	—	10	—	10	ns
NRZ Read Data Setup Time (20 ns for TRFCP ≤ 100 ns, else 13 ns)	t_{RDS}	20	—	13	—	13	—	ns
NRZ Read Data Hold Time (20 ns for TRFCP ≤ 100 ns, else 13 ns)	t_{RDH}	20	—	13	—	13	—	ns
ST-506 AMF Setup Time (20 ns for TRFCP ≤ 100 ns, else 13 ns)	t_{AMFS}	20	—	13	—	13	—	ns
WRCLK Pulse Width (50 ± 18.8% Duty Cycle)	PW_{WCL}	31	69	19	43	12	28	ns
NRZ Write Data Setup Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{WDS}	14	—	8	—	8	—	ns
NRZ Write Data Hold Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{WDH}	14	—	8	—	8	—	ns
ST-506 AMEN Setup Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{AMES}	14	—	8	—	8	—	ns
ST-506 AMEN Hold Time (14 ns for TRFCP ≤ 100 ns, else 8 ns)	t_{AMEF}	14	—	8	—	8	—	ns

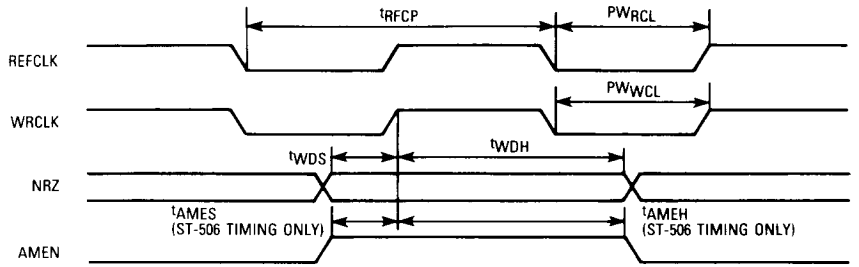


Figure 15. Disk Read Timing Diagram

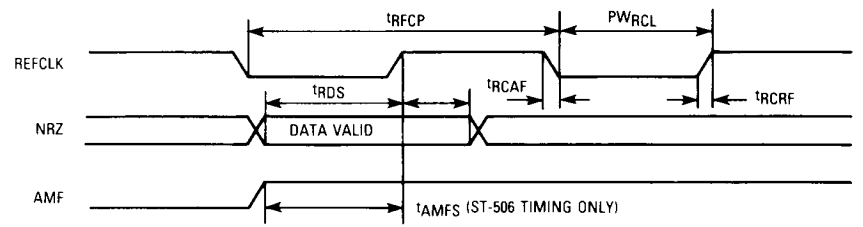


Figure 16. Disk Write Timing Diagram

SPECIAL TIMING (see Figures 17 and 18)

Characteristic	Symbol	Min	Max	Unit
Reset Low Input Pulse Width	PW_{RST}	4	—	t_{cyc}
Mode/Parity/Unit Programming Setup	t_{MPS}	2	—	t_{cyc}
Mode/Parity/Unit Programming Hold	t_{MPH}	0	1	t_{cyc}
Timer Input Capture PW (Single-Chip Mode)	PW_{TIM}	200	—	ns

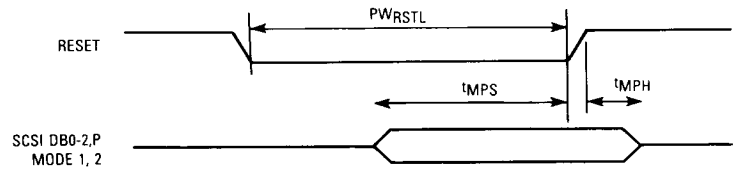


Figure 17. Reset Timing Diagram

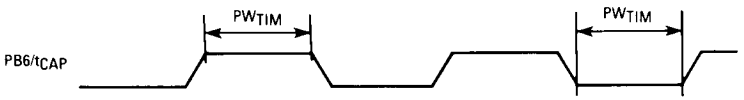


Figure 18. Input Capture Timing Diagram (Single-Chip Mode)

PORT READ AND WRITE TIMING (see Figures 19 and 20)

Parameter	Symbol	Min	Max	Unit
Delay Time, Peripheral Data Write	t_{PWD}	—	75	ns
Peripheral Read Data Setup Time PB0-PB7, PC0-PC7 SDB0-SDB7, P SCSI Inputs	 t_{PDSU1} t_{PDSU2} t_{PDSU3}	 50 0 125	 — — —	ns
Peripheral Read Data Hold Time PB0-PB7, PC0-PC7 SDB0-SDB7, P SCSI Inputs	 t_{PDH1} t_{PDH2} t_{PDH3}	 25 0 125	 — — —	ns

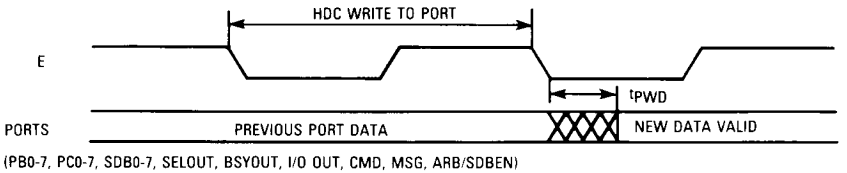


Figure 19. Port Write Timing Diagram

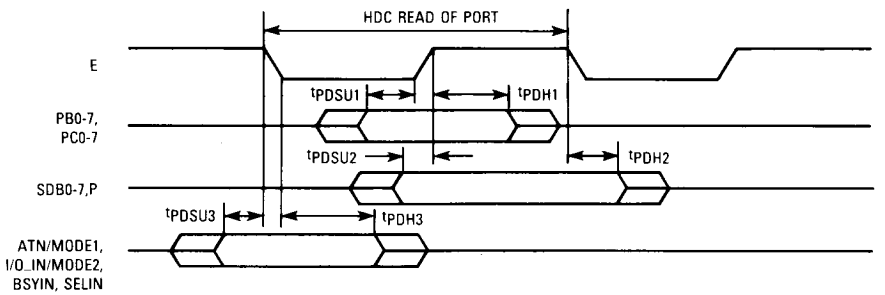


Figure 20. Port Read Timing Diagram

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola using the following media:

MDOS, disk file
MS-DOS disk file (360K)
EPROM(s) 2516, 2716, 2532, 2732

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, sales person, or a Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MDOS™ or MS™-DOS disk file) may be submitted for pattern generation. Disk should be programmed with the customer's program, using positive logic sense for address and data. The diskette should be clearly labeled with the customer's name, date, project or product name, and the filename containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MDOS Disk File

MDOS is Motorola's Disk Operating System available on the EXORciser® development system. The disk media submitted must be a single-sided, single-density, 8-inch MDOS compatible floppy diskette. The diskette must contain the minimum set of MDOS system files in addition to the pattern file.

The .LO output of the M68HC11 cross assembler should be furnished. In addition, the file must be produced using the ROLLOUT command, so that it contains the absolute image of the M68HC99 memory. It is necessary to include the entire memory image of both program and data space. All unused bytes, including those in the user space, must be set to logic zero.

MS-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. Disk media submitted must be a standard density (360K), double-sided 5-1/4 inch compatible floppy diskette. The diskette must contain the object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by cross assemblers and linkers on IBM PC style machines.

To contain the entire MC68HC99 program, 12K bytes of EPROM are necessary. Six 2516 or 2716 type EPROMs or three 2532 or 2732 type EPROMs can be submitted for pattern generation. The EPROM is programmed with the customer's program using positive logic sense for address and data. Submissions on two EPROMs must be clearly marked. All unused bytes, including the user's space, must be set to zero.

If the MC68HC99 HDC ROM pattern is submitted on three 2532 or 2732 EPROMs, or on six 2516 or 2716 type EPROMs, memory map addressing is one-for-one. The data space ROM runs from EPROM address $018 to $05F, and program space ROM runs from EPROM address $960 to $FF7, with vectors from $FFC to $FFF.

For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.

Verification Media

All original pattern media, EPROMs or floppy disks, are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM Verification Units (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

Ordering Information

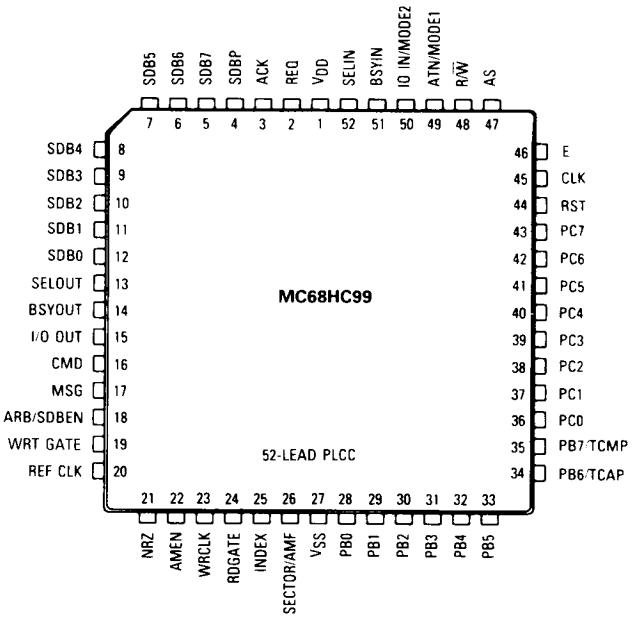
The following table provides generic information pertaining to the package type, temperature, and order numbers for the MC68HC99.

Package Type	Temperature	Order Number
PLCC	0°C to 70°C	MC68HC99FN
FN Suffix	−40°C to +85°C	MC68HC99CFN

3

MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
EXORciser is a registered trademark of Motorola Inc.
IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENT



3