MOTOROLA

MC68HC11A4

Product Preview

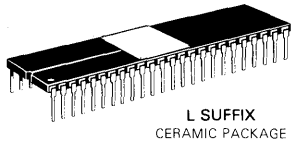
MC68HC11A4 8-BIT MICROCOMPUTER

The HCMOS MC68HC11A4 is an advanced microcomputer (MCU) containing highly sophisticated on-chip peripheral functions. An improved instruction set provides additional capability while maintaining compatibility with the other members of the M6801 Family. The fully static design allows operation at frequencies down to dc, further reducing its already low power consumption. Features available in addition to the normal M6801 features include:

- 4K Bytes of ROM
- 512 Bytes of EEPROM
- 256 Bytes of RAM (All Saved During Standby)
- Enhanced 16-Bit Timer System
  - Four Stage Programmable Prescaler
  - Three Input Capture Functions
  - Five Output Compare Functions
- An 8-Bit Pulse Accumulator Circuit
- An Enhanced Non-Return-To-Zero Serial Communications Interface (SCI)
- A New Serial Peripheral Interface
- Eight Channel 8-Bit A/D Converter
- A Real Time Interrupt Circuit
- A Computer Operating Properly (COP) Watchdog System

HCMOS
(HIGH-DENSITY HIGH-PERFORMANCE SILICON GATE)

MICROCOMPUTER

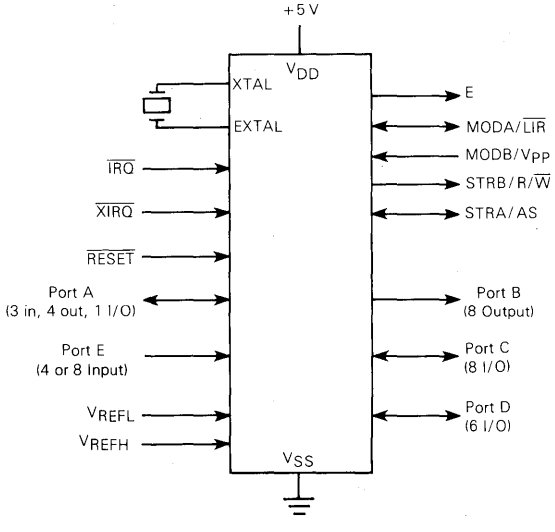


L SUFFIX
CERAMIC PACKAGE

Also Available In
52-Pin Quad Pack

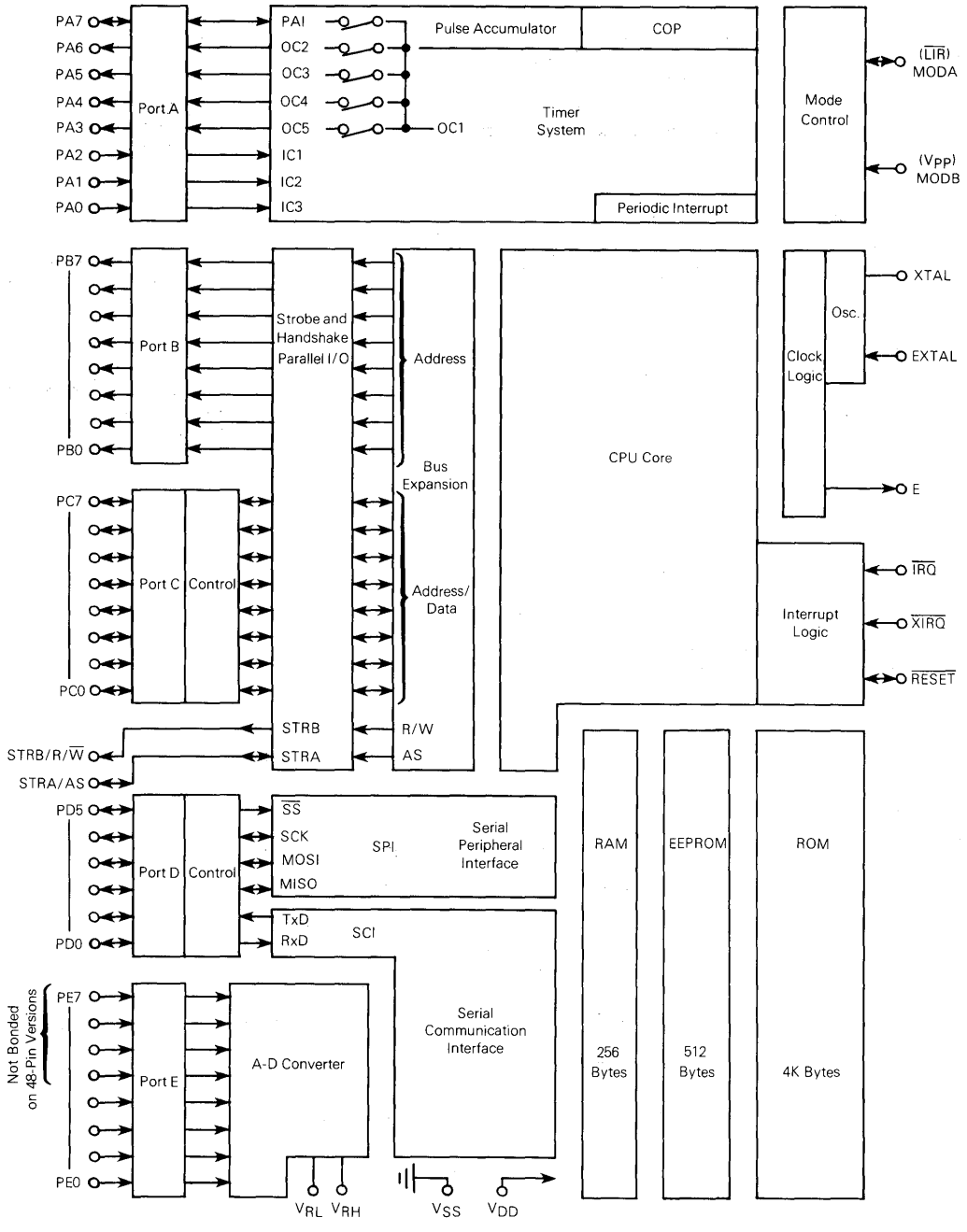
3

PIN ASSIGNMENT



This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

FIGURE 1 — MC68HC11A4 BLOCK DIAGRAM



GENERAL DESCRIPTION

The MC68HC11A4 is a single-chip microcomputer that utilizes HCMOS techniques to provide the low-power characteristics and high noise immunity of CMOS plus the high-speed operation of HMOS. On chip memory systems include a 4K byte ROM, 512 bytes of electrically erasable programmable ROM (EEPROM), and 256 bytes of static RAM. The MC68HC11A4 microcomputer also provides highly sophisticated, on-chip peripheral functions including: an 8-channel analog-to-digital converter, a serial communications interface (SCI) subsystem, and a serial peripheral interface (SPI) subsystem.

New design techniques are used to provide a 2 MHz nominal bus rate. The timer system is expanded to provide three input capture lines, five output compare lines, and a real time interrupt circuit. This gives the MC68HC11A4 one of the most comprehensive timer systems found on a single-chip microcomputer. Other features of the MC68HC11A4 include: a pulse accumulator which can be used to count external events (event counting mode) or measure an external period (input gates accumulation of internal clock — E/64); a computer operating properly (COP) watchdog system which helps protect against software failures; a programmable clock monitor system which causes generation of a system reset in case the clock is lost or running too slow; and an illegal opcode detection circuit which provides an unmaskable interrupt if an illegal opcode fetch is detected.

OPERATING MODES

The MC68HC11A4 MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip (mode 0) and expanded multiplexed (mode 1), and the special operating modes are bootstrap and special test. The levels required on the MODA and MODB pins for mode selection are discussed in **FUNCTIONAL PIN DESCRIPTION.** The characteristics of the operating modes are discussed below.

SINGLE-CHIP MODE (MODE 0)

In the single-chip mode the MCU functions as a self-contained microcomputer and has no external address or data bus. In this mode the MCU provides maximum use of its pins for on-chip peripheral functions, and all address and data activity occurs within the MCU. As discussed in **FUNCTIONAL PIN DESCRIPTION,** when MODA = 0 and MODB = V_{DD} the single chip mode is selected during reset.

EXPANDED MULTIPLEXED MODE (MODE 1)

In this mode, two I/O ports plus two additional I/O lines become address, data, and control (AS and R/$\overline{W}$) to allow the MCU to address up to 64K bytes of address space. High order address bits are output on the port B pins. Low order address bits and the data bus are time multiplexed on the eight port C pins. Port D bit 6 becomes the address strobe (AS) control output which is used in demultiplexing the low order address from the data at port C. The R/$\overline{W}$ control pin (port D, pin 7) is used to control the direction of data transfers on the port C bus. Refer to **FUNCTIONAL PIN DESCRIPTION** and **INPUT/OUTPUT PORTS** for additional information regarding address strobe, read/write, port B, and port C.

BOOTSTRAP MODE

The bootstrap mode is considered a special mode as distinguished from the normal operating single-chip mode. In the bootstrap mode, all vectors are fetched from the 128 byte on-chip boot loader ROM. This is a very versatile mode since there are essentially no limitations on the special purpose program that is boot loaded into the internal RAM. The boot loader is contained in 128 bytes of ROM which is enabled as internal memory space at $BF80-$BFFF. The boot loader contains a small program which reads a 256 byte program into on-chip RAM ($0000-$00FF). After the character for address $00FF is received, control is automatically passed to that program at memory address $0000 and the MCU operates in the single-chip mode.

In the bootstrap mode, the serial receive logic is initialized by software in the boot loader ROM. This allows the program control of the serial communications interface (SCI) baud and word format.

During initialization of the special bootstrap mode, a special control bit is configured to permit access to a number of special test control bits which allows for self testing of the MCU in the bootstrap mode. Also, since the mode control bits can be written to, the operating mode of the MCU may be changed from the special bootstrap mode (which is a single-chip mode by default) to expanded multiplexed mode under program control.

TEST MODE

The test mode is considered a special mode as distinguished from the normal operating expanded multiplexed mode; however, it is considered as operating in the expanded multiplexed mode since external memory may be addressed. The reset vector is fetched from external memory space $BFFE-$BFFF; therefore, program control may be vectored to an external test program.

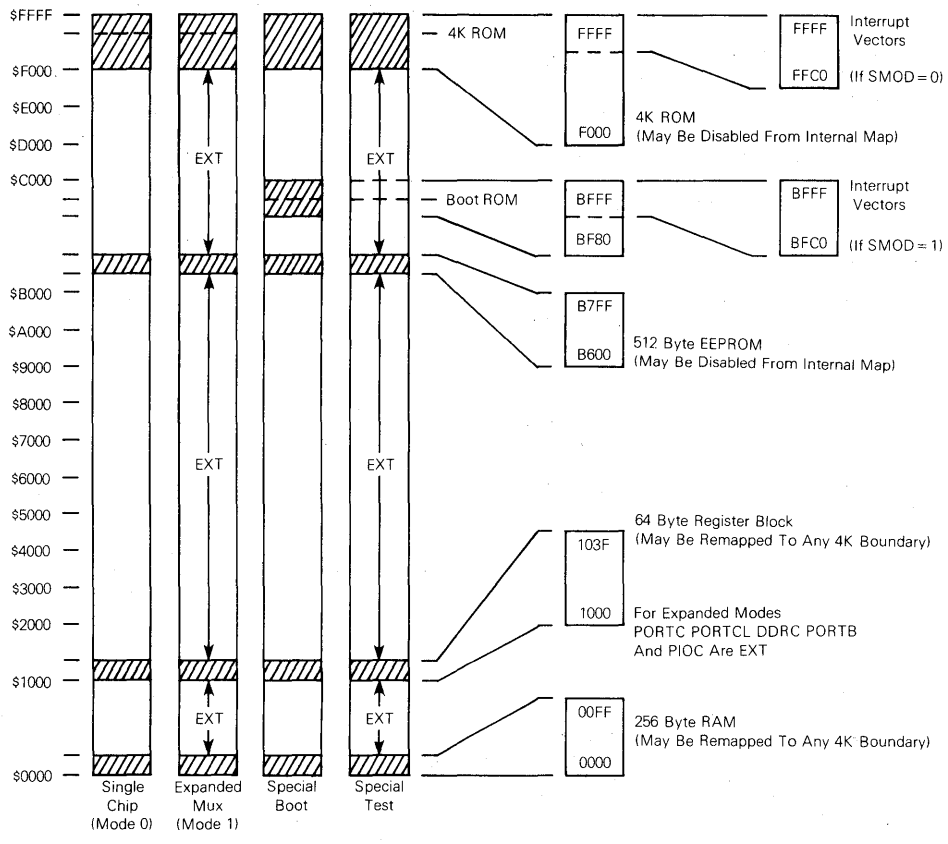
The test mode is primarily intended as the main production test mode at the time of manufacture; however, it may also be used to program calibration or personality data into the internal EEPROM (electrically erasable programmable read only memory) of the MC68HC11A4. During initialization of the test mode, a special control bit is configured to permit access to a number of special test control bits. Also, since the mode control bits can be written to in the test mode, the operating mode of the MCU may be changed from the special test mode (which is an expanded multiplexed mode by default) to the single-chip mode under program control.

MEMORY

Composite memory maps for each MC68HC11A4 mode of operation are shown in Figure 2. These modes include single-chip, expanded multiplexed, special boot, and special test.

In the single-chip mode (mode 0) of Figure 2, the MC68HC11A4 does not generate external addresses. The actual internal memory locations are shown in the shaded areas of Figure 2 and the contents of these shaded areas are

FIGURE 2 — MC68HC11A4 MEMORY MAPS



shown on the right side of the diagram. For example: memory locations $0000 through $00FF contain the 256 bytes allocated to RAM; memory locations $1000 through $103F are allocated for a 64-byte register block; memory locations $B600 through $B7FF are allocated for a 512-byte EEPROM (electrically eraseable programmable read only memory); and memory locations $F000 through $FFFF are allocated for 4K bytes of ROM (memory locations $FFC0 through $FFFF are reserved for the interrupt and reset vectors).

The expanded multiplexed mode (mode 1) memory locations shown in Figure 2 are basically the same as for the single-chip mode; however, the memory locations between the shaded areas (designated EXT) are for externally addressed memory and I/O.

The special bootstrap mode memory locations are similar to the single-chip memory locations except that a special bootstrap program is addressed at memory locations $BF80 through $BFFF. The special bootstrap program controls the process of boot loading a 256 byte program through a serial port into internal RAM.

The special test mode memory locations are similar to the expanded multiplex mode except the interrupt vectors are at external memory locations.

CENTRAL PROCESSING UNIT

The central processing unit (CPU) of the MC68HC11A4 is basically an extension of the MC6801 CPU. In addition to being able to execute all MC6800 and MC6801 instructions, the MC68HC11A4 uses a 4-page opcode map to allow execution of 91 new opcodes. As in the MC6801, the CPU of the MC68HC11A4 is implemented independently from the I/O,

memory, or on-chip peripheral configurations. Consequently, this CPU can be treated as an independent processor communicating with these internal subsystems when operating in the single-chip mode. However, when the MC68HC11A4 is operating in the extended multiplexed mode, it is capable of addressing external memory and peripherals in addition to communicating with the on-chip subsystems.

The MC68HC11A4 CPU has seven registers available to the programmers as shown in Figure 3. The interrupt stacking order is shown in Figure 4. The seven registers are discussed below.

FIGURE 3 – MC68HC11A4 PROGRAMMING MODEL

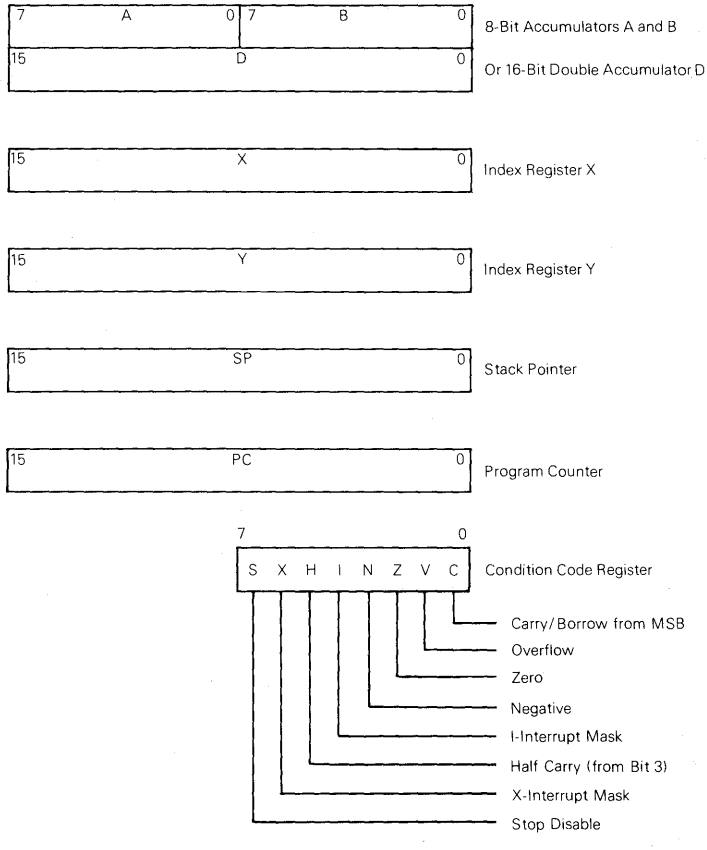
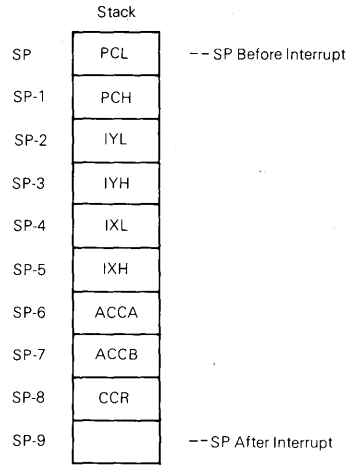


FIGURE 4 — INTERRUPT STACKING ORDER



ACCUMULATOR A AND B

Accumulator A and accumulator B are general purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. As in the MC6801, these two accumulators can be concatenated into a single double-byte accumulator called the D accumulator.

INDEX REGISTER X (IX)

The 16-bit IX register is used during indexed modes of addressing. It provides a 16-bit indexing value which may be added to an 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage area.

INDEX REGISTER Y (IY)

The 16-bit IY register is also used during indexed modes of addressing similar to the IX register; however, most instructions using the IY register require an extra byte of machine code and a cycle of execution time since they are two byte opcodes. The IY register can also be used as a counter or as a temporary storage in the same manner as the IX register.

STACK POINTER (SP)

The stack pointer (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 last-in-first-out read/write registers which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack (a push), the SP is decremented; whereas, each time a byte is removed from the stack (a pull) the SP is incremented. The address contained in the SP also indicates the location at which the accumulators (A and B), IX, and IY can be stored during certain instructions.

PROGRAM COUNTER (PC)

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

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit signifies the results of the instruction just executed. These bits can be individually tested by a program and a specific action can be taken as a result of the test. Each individual condition code register bit is explained below.

Carry/Borrow (C)

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

Overflow (V)

The overflow bit is set if there was an arithmetic overflow as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

The zero bit is set if the result of the last arithmetic, logic, or data manipulation was zero; otherwise, the Z bit is cleared.

Negative (N)

The negative bit is set if the result of the last arithmetic, logic, or data manipulation was negative (b7 of result equal to a logic one); otherwise, the N bit is cleared.

I Interrupt Mask (I)

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

Half Carry (H)

The half carry bit is set to a logic one when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction; otherwise, the H bit is cleared.

X Interrupt Mask (X)

The X interrupt mask bit is set only by hardware (Reset or $\overline{\text{XIRQ}}$); and it is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

The stop disable bit is set to disable the STOP instruction, and cleared to enable the STOP instruction. The S bit is program controlled. The STOP instruction is treated as no operation (NOP) if the S bit is set.

FUNCTIONAL PIN DESCRIPTION

The below pin descriptions do not include the I/O ports. They are discussed separately under **INPUT/OUTPUT PORTS.**

V_{DD} AND V_{SS}

Power is supplied to the MC68HC11A4 using these two pins. V_{DD} is power input (+5 V) and V_{SS} is the power return path.

$\overline{RESET}$

This active low bi-directional control pin is used as an input to initialize the MC68HC11A4 to a known start-up state, and as an open-drain output to indicate an internal failure has been detected in either the clock monitor or computer operating properly (COP) circuit.

XTAL, EXTAL

These two inputs provide for an interface with either a crystal input or a CMOS compatible clock to control the MC68HC11A4 internal clock generator circuitry. The frequency applied to these pins should be four times the desired internal clock rate since an internal divide-by-four circuit determines the actual E-clock rate. When a crystal is used, a 25 picofarad capacitor should be connected between V_{SS} and each of these two pins (XTAL and EXTAL); however, if a CMOS compatible external clock is used, the signal should be connected to the EXTAL pin and the XTAL pin should be left disconnected.

E

The E pin provides an output for the internally generated E-clock which can be used as a timing reference. The frequency of the E output is actually one fourth that of the input frequency at the XTAL and EXTAL pins. In general when the E pin is low, an internal process is taking place and, when high, data is being accessed. This output becomes inactive during the power-saving wait mode if the MC68HC11A4 is operating in the single-chip or bootstrap modes (see **MODA/$\overline{LIR}$, MODB/V_{PP}** description below).

$\overline{IRQ}$

The $\overline{IRQ}$ pin provides a means for requesting asynchronous interrupts to the MC68HC11A4. The $\overline{IRQ}$ interrupt input is program selectable with a choice of either negative edge-sensitive or level-sensitive triggering. The $\overline{IRQ}$ interrupt input is always configured to level-sensitive triggering during reset. The $\overline{IRQ}$ pin requires an external resistor to V_{DD}. The MCU completes the current instruction before responding to an interrupt request on the $\overline{IRQ}$ pin.

If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

$\overline{XIRQ}$

The $\overline{XIRQ}$ pin provides a means for requesting asynchronous non-maskable interrupts to the MC68HC11A4, after a power-on reset. During reset (including power-on reset), the X bit in the condition code register is set and the $\overline{XIRQ}$ interrupt is masked to preclude interrupts on this line until MCU operation is stabilized. The X bit may then be cleared by program control (using the transfer accumulator A instruction, TAP); however, the X bit can only be set again by reset or by recognition of a hardware $\overline{XIRQ}$ interrupt. Once the X bit in the condition code register is cleared, an interrupt on the $\overline{XIRQ}$ pin will be serviced as soon as the MCU completes the current instruction. When an $\overline{XIRQ}$ interrupt is recognized, on-chip hardware automatically sets the X bit. The X bit can be cleared either as part of interrupt routine by the TAP instruction (nested interrupt) or by the return from interrupt instruction. The $\overline{XIRQ}$ pin requires an external resistor to V_{DD}.

The $\overline{XIRQ}$ input may also be used to return the MCU to normal operation from the low-power stop mode by applying a low level to the $\overline{XIRQ}$ pin. If the X bit in the condition code register is cleared and the MCU is in the stop mode, a low input on the $\overline{XIRQ}$ brings the MCU out of the stop mode and operation resumes with the stacking operation leading to service of the $\overline{XIRQ}$ request. If the X bit is set and the MCU is in the stop mode, a low input on the $\overline{XIRQ}$ brings the MCU out of the stop mode and operation resumes with the program instruction following the STOP instruction.

MODA/$\overline{LIR}$, MODB/V_{PP}

These pins have alternate functions, MODA and MODB controlling one function, V_{PP} controlling an alternate function, and $\overline{LIR}$ used for an alternate function.

MODA, MODB

During reset these two pins are used to control the two basic operating modes of the MC68HC11A4 plus two special operating modes. The modes versus MODA and MODB inputs are shown in the table below.

MODB	MODA	Mode Selected
V_{DD}	0	Single-Chip (Mode 0)
V_{DD}	1	Expanded Multiplexed (Mode 1)
#	0	Special Bootstrap
#	1	Special Test

1 = Logic High
0 = Logic Low
# = 1.4 Times V_{DD} (or Higher)

V_{PP}

In addition to the MODA function, the MODA/V_{PP} pin is also used to supply the programming voltage for programming the internal EEPROM. Changing the voltage applied to this pin after reset has no affect on mode selection.

$\overline{LIR}$

In addition to the MODA function, the MODA/$\overline{LIR}$ pin provides an output as an aid in debugging once reset is completed. The LIR pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle (opcode fetch). Some MC68HC11A4 opcodes are two consecutive bytes long including a page 2 (PG2), page 3 (PG3), or page 4 (PG4) prebyte. For these instructions $\overline{LIR}$ goes low for only the first (prebyte) opcode byte fetch.

NOTE

The $\overline{LIR}$ output will not go low for at least two E-clock cycles after reset because of the reset vector fetch.

V_{REFL}, V_{REFH}

These two pins provide the reference voltage for the analog-to-digital converter.

R/$\overline{W}$/STRB

This pin provides two different functions depending on the operating mode. In single-chip mode the pin provides the STRB (output strobe) function and in the expanded multiplexed mode it provides the R/$\overline{W}$ (read-write) function.

In the single-chip mode the STRB pin acts as a programmable strobe. This strobe can also be used to provide a data acknowledge (handshake) to a parallel I/O device.

In the expanded multiplexed mode the R/$\overline{W}$ (read/write) is used to control the direction of transfers on the external data bus. A low level (write) on the R/$\overline{W}$ pin enables the data bus output drivers to the external data bus. A high level (read) on this pin forces the output drivers to a high-impedance state and data is read from the external bus.

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides the STRA (input strobe) function and in the expanded multiplexed mode it provides the AS (address strobe) function.

In the single-chip mode, the STRA pin acts as a programmable input strobe. This input is also used with STRB and port C for full handshake modes of parallel I/O.

In the expanded multiplexed mode the AS (address strobe) output may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT PORTS

There are five 8-bit ports on the MC68HC11A4 MCU. All of these ports serve more than one purpose depending on the mode configuration of the MCU. A summary of the pins versus function and mode is provided in Table 1 and discussed in the following paragraphs. Because the port functions are controlled by the particular mode selected, each port is discussed for its function(s) during the mode of operation.

SINGLE-CHIP MODE

In the single-chip mode the MC68HC11A4 functions as a monolithic microcomputer without external address or data buses. In this mode, four of these ports (A, B, C, D) are configured as parallel I/O data ports. Port E can be used for general purpose static inputs and/or analog-to-digital converter channel inputs.

Port A

In all operating modes (including the single-chip mode) port A may be configured for: three input capture functions (IC1, IC2, IC3), four output compare functions (OC2, OC3, OC4, OC5), and a pulse accumulator input (PAI) or a fifth output compare function (OC1).

Each of the input capture pins provide for a transitional input which is used to latch a timer value into a 16-bit read-only register (input capture register). The value latched by an input capture corresponds to the value of the free running counter which is part of the timer system. External devices provide the transitional inputs and internal decoders determine which input transition edge (rising, falling, or either) is sensed.

Each of the output compare pins provide for an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. The outputs can be used externally to indicate that a certain period of time has elapsed.

When port A pin 7 (PA7) is configured as a pulse accumulator input (PAI), the external input pulses are applied to a pulse accumulator register within the MC68HC11A4.

Each port A pin that is not used for its alternate timer function, as described above, may be used as a general purpose input or output line.

Port B

In the single-chip mode, all of the port B pins are general purpose output pins. During MCU read cycles the levels sensed at the input side of the port B output drivers is read. Port B may also be used in a simple strobed output mode where the STRB (port D bit 7) pulses each time port B is written.

Port C

In the single-chip mode, all port C pins are general purpose input/output pins. Port C inputs can be latched by the STRA input (at port D bit 6). Port C may also be used in full handshake modes of parallel I/O where the STRA input and STRB output act as handshake control lines.

Port D

In the single-chip mode port D bits 0-5 may be used for general I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bits 6 and 7 are used as handshake control signals for ports B and C.

Bit 0 is the receive data input (RxD) for the serial communication interface (SCI).

Bit 1 is the transmit data output (TxD) for the SCI.

Bits 2 through 5 are dedicated to the serial peripheral interface (SPI). Bit 2 is the master-in-slave-out (MISO) line; this pin is an input when the SPI is configured as a master device and an output when configured as a slave device. Bit 3 is the master-out-slave-in (MOSI) line; this pin is an output when the SPI is configured as a master device and an input when configured as a slave device. Bit 4 is the serial clock (SCK) and is an output when the SPI is configured as a master and an input when configured as a slave device. Bit 5 is the slave select ($\overline{SS}$) input which receives an active low signal to enable a slave device to accept SPI data.

Bit 6 (STRA) and 7 (STRB) are discussed in **FUNCTIONAL PIN DESCRIPTION.**

TABLE 1 — PORT SIGNAL SUMMARY

Port-Bit	Single-Chip Modes 0 and Bootstrap Mode	Expanded Multiplexed Mode 1 and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/and-or OC1	PA3/OC5/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/SS	PD5/SS
D-6	STRA	AS
D-7	STRB	R/$\overline{W}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4 ##	PE4/AN4 ##
E-5	PE5/AN5 ##	PE5/AN5 ##
E-6	PE6/AN6 ##	PE6/AN6 ##
E-7	PE7/AN7 ##	PE7/AN7 ##

- not bonded in 48-pin variations

Port E

In all operating modes (including the single-chip mode), port E is used for general purpose static inputs and/or analog-to-digital (A/D) channel inputs. Port E should not be read as static inputs while an A/D conversion is actually taking place.

NOTE

On 48-pin packaged versions of the MC68HC11A4, the four most significant bits of port E are not connected to pins.

EXPANDED MULTIPLEXED MODE

In the expanded multiplexed mode, the MC68HC11A4 has the capability of accessing a 64K byte address space. The total address space includes the same on-chip memory address as for single-chip mode plus external peripheral devices. In this mode ports B, C, and bits 6 and 7 of port D are configured as a memory expansion bus.

Port A

In all operating modes (including the expanded multiplexed mode), port A may be configured for: three input capture functions (IC1, IC2, IC3), four output compare functions (OC2, OC3, OC4, OC5), and a pulse accumulator input (PAI) or a fifth output compare functon (OC1).

Each of the input capture pins provide for a transitional input which is used to latch a timer value into a 16-bit read-only register (input capture register). The value latched by an input capture corresponds to the value of a free running

counter which is part of the timer system. External devices provide the transitional inputs and internal decoders determine which input transition edge (rising, falling, or either) is sensed.

Each of the output compare pins provide for an output whenever a match is made between the value in the free running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. The outputs can be used externally to indicate that a certain period of time has elapsed.

When port A pin 7 (PA7) is configured as a pulse accumulator input (PAI), the external input pulses are applied to a pulse accumulator register within the MC68HC11A4.

Each port A pin that is not used for its alternate timer function as described above, may be used as a general purpose input or output line.

Port B

In the expanded multiplexed mode, all of the port B pins act as high order address output pins. During each MCU cycle, bits 8 through 15 of the address are output on the PB0-PB7 lines respectively.

Port C

In the expanded multiplexed mode, all port C pins are configured as multiplexed address/data pins. During the address portion of each MCU cycle, bits 0 through 7 of the address are output on the PC0-PC7 lines. During the data portion of each MCU cycle (E high), bits 0 through 7 (D0-D7) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

Port D

In the expanded multiplexed mode port D bits 0-5 may be used for general I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bits 6 and 7 act as expansion bus control lines AS and R/W respectively.

Bit 0 is the receive data input (RxD) for the serial communications interface (SCI).

Bit 1 is the transmit data output (TxD) for the SCI.

Bits 2 through 5 are dedicated to the serial peripheral interface (SPI). Bit 2 is the master-in-slave-out (MISO) line; this pin is an input when the SPI is configured as a master device and an output when configured as a slave device. Bit 3 is the master-out-slave-in (MOSI) line; this pin is an output when the SPI is configured as a master device and an input when configured as a slave device. Bit 4 is the serial clock (SCK) and is an output when the SPI is configured as a master and an input when configured as a slave device. Bit 5 is the slave select ($\overline{SS}$) input which receives an active low signal to enable a slave device to accept SPI data.

Bit 6 (AS) and 7 (R/$\overline{W}$) are discussed in **FUNCTIONAL PIN DESCRIPTION.**

Port E

In all operating modes (including the expanded multiplexed mode), port E is used for general purpose static inputs and/or analog-to-digital (A/D) channel inputs. Port E should not be read as static inputs while an A/D conversion is actually taking place.

NOTE

On 48-pin packaged versions of the MC68HC11A4, the four most significant bits of port E are not connected to external pins.

BOOTSTRAP MODE

In the bootstrap mode all I/O port pins function the same as in the single-chip mode. Operational differences are discussed in **OPERATING MODES.**

TEST MODE

In the test mode all I/O port pins function the same as in the expanded multiplexed mode. Operational differences are discussed in **OPERATING MODES.**

INTERRUPTS

The MC68HC11A4 MCU interrupts can be generated by any of four different basic methods: (1) by presenting the appropriate external signal; (2) by enabling interrupts from the programmable timer output compare or input capture, serial communication interface, serial peripheral interface timer overflow, pulse accumulator, or parallel I/O; (3) by executing a software interrupt (SWI) instruction; or (4) by detection of an illegal opcode.

The program may also be interrupted by: (1) detection of a timeout in the computer operating properly (COP) circuit, (2) clock monitor detects loss of the E-clock or a low frequency E-clock, or (3) by a reset. The above three methods of interrupting the program result in fetching a reset vector rather than an interrupt vector; however, they do interrupt the program.

When an external or internal (hardware) interrupt occurs, the interrupt is not serviced until the current instruction being executed is completed. Until the current instruction is complete, the interrupt is considered pending. After completion of current instruction execution, unmasked interrupts may be serviced in accordance with an established fixed hardware priority circuit; however, one I bit related interrupt source may be elevated to the highest I bit priority position in the circuit.

Seventeen hardware interrupts and one software interrupt (excluding reset type interrupts) can be generated from all of the possible sources. The interrupts can be divided into two basic categories, maskable and non-maskable. In the MC68HC11A4 fifteen of the interrupts can be masked using the condition code register I bit. In addition to being maskable by the I bit in the condition code register, all of the on-chip interrupt sources are individually maskable by control bits. The software interrupt (SWI instruction) is a non-maskable instruction rather than a maskable interrupt source. The last interrupt (external input to the $\overline{XIRQ}$ pin) is considered as a non-maskable interrupt because once enabled, it cannot be masked by software; however it is masked during reset and upon receipt of an interrupt at the $\overline{XIRQ}$ pin. Table 2 provides a list of each interrupt, its vector location in ROM, and the actual condition code register bit that masks it. A discussion of the various interrupts is provided below.

TABLE 2 — INTERRUPT VECTOR ASSIGNMENTS

Vector Address	Interrupt Source	Masked By
FFC0, C1	Reserved	—
⋮	⋮	
FFD4, D5	Reserved	—
FFD6, D7	SCI Serial System	I Bit
FFD8, D9	SPI Serial Transfer Complete	I Bit
FFDA, DB	Pulse Accumulator Input Edge	I Bit
FFDC, DD	Pulse Accumulator Overflow	I Bit
FFDE, DF	Timer Overflow	I Bit
FFE0, E1	Timer Output Compare 5	I Bit
FFE2, E3	Timer Output Compare 4	I Bit
FFE4, E5	Timer Output Compare 3	I Bit
FFE6, E7	Timer Output Compare 2	I Bit
FFE8, E9	Timer Output Compare 1	I Bit
FFEA, EB	Timer Input Capture 3	I Bit
FFEC, ED	Timer Input Capture 2	I Bit
FFEE, EF	Timer Input Capture 1	I Bit
FFF0, F1	Real Time Interrupt	I Bit
FFF2, F3	$\overline{IRQ}$ (External Pin or Parallel I/O)	I Bit
FFF4, F5	$\overline{XIRQ}$ Pin (Pseudo Non-maskable Interrupt)	X Bit
FFF6, F7	SWI	None
FFF8, F9	Illegal Op-Code Trap	None
FFFA, FB	COP Failure (Reset)	None
FFFC, FD	COP Clock Monitor Fail (Reset)	None
FFFE, FF	$\overline{RESET}$	None

TIMER INTERRUPTS

The timer system provides nine of the fifteen interrupt possibilities: five output compare interrupts, three input capture interrupts, and a timer overflow interrupt.

The timer contains five 16-bit output compare registers which are program controlled and may be loaded with a number between $0000-$FFFF. The value in each output compare register is then compared to a 16-bit comparator, which is loaded from the timer free running counter, during each clock cycle. If a match is found between the 16-bit comparator value and the output compare register value, the corresponding output compare flag is set. When the output compare flag is set, a corresponding output compare interrupt may be generated and/or an external output may be generated at the corresponding port A pin(s). Port A outputs PA3 through PA7 are used as output pins for output compare functions OC1 through OC5.

In addition to the five output compare interrupts, the timer also provides for three input capture interrupts. The timer contains three 16-bit latch registers which are used to latch the value of the free running counter (in the timer) when an input capture edge is applied to the corresponding PA0-PA2 pin. The value of the free running counter is latched into the corresponding input capture register and an internal interrupt may be generated. The interrupt routine can then read the storage register and determine the time at which the input capture was detected.

The timer may also provide an interrupt when the free running counter changes value from $FFFF to $0000 (overflow). The 16-bit free running counter repeats this change once for every 65,536 inputs from a prescaler circuit. The prescaler is programmable for either divide-by-1, divide-by-4, divide-by-8, or divide-by-16 of the MCU E-clock. Thus, the prescaler extends the actual range of the free running counter and the time between timer overflow interrupts from 2^{16} to 2^{256} E-clock inputs to the prescaler.

REAL TIME INTERRUPTS

The real time interrupt is a maskable interrupt that occurs periodically at a rate of $E/2^{13}$, $E/2^{14}$, $E/2^{15}$, or $E/2^{16}$.

EXTERNAL INTERRUPTS

Two external interrupts are accessable using the $\overline{IRQ}$ and the $\overline{XIRQ}$ pins. The $\overline{IRQ}$ interrupt is a maskable interrupt while the $\overline{XIRQ}$ interrupt is considered a non-maskable interrupt; however, the $\overline{XIRQ}$ interrupt is masked during reset and immediately following receipt of an $\overline{XIRQ}$ interrupt signal. These interrupts are controlled by the I and X bits in the condition code register as discussed in **CENTRAL PROCESSING UNIT.**

SOFTWARE INTERRUPT (SWI)

The software interrupt is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the condition code register set). The SWI instruction is executed similar to other maskable interrupts in that it sets the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once it is fetched no other interrupt can be honored until the first instruction in the SWI service routine is completed.

SERIAL PERIPHERAL INTERFACE (SPI) INTERRUPT

A serial peripheral interface (SPI) interrupt is generated when a serial data transfer between the MC68HC11A4 and an external device has been completed. This interrupt is masked if the condition code register I bit is set.

SERIAL COMMUNICATIONS INTERFACE (SCI) INTERRUPT

A serial communications interface (SCI) interrupt is generated if any one of the following occurs in the SCI:

1. Transmit data register is empty
2. Transmission of data is complete
3. Receive data register is full or an overflow occurred in the receive data register
4. Idle line detected by receiver.

The SCI interrupt is masked if the condition code register I bit is set.

PULSE ACCUMULATOR INTERRUPT

The pulse accumulator contains an 8-bit counter which is program controlled to either count input pulses (event counting) at PA7 or to count internal E/64 clocks subject to an enable signal at PA7 (gated time accumulation). When the counter has an overflow from $FF to $00 a pulse accumulator overflow interrupt is generated provided the I bit in the condition code register is clear.

When the input to the pulse accumulator is a gate input at PA7 for counting internal E/64 clocks, the trailing edge of the gate signal (end of counting cycle) can generate an interrupt. This pulse accumulator input edge interrupt is generated provided the I bit in the condition code register is clear. Refer to **PULSE ACCUMULATOR** for more information.

PARALLEL I/O INTERRUPT

The parallel I/O subsystem can generate an interrupt which uses the same vector as the $\overline{IRQ}$ interrupt. The purpose of sharing the $\overline{IRQ}$ vector is to allow external emulation of the parallel I/O subsystem in expanded multiplexed modes.

RESETS

The MC68HC11A4 MCU has four possible types of reset: an active low external reset pin ($\overline{RESET}$), a power-on reset function, a computer operating properly (COP) watchdog timer reset, and a clock monitor reset.

$\overline{RESET}$ PIN

The $\overline{RESET}$ pin is used to reset the MCU to provide an orderly software startup procedure. To request an external reset, the $\overline{RESET}$ pin must be held low for eight E_{cyc} (two E_{cyc} if internal resets are not used).

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in power supply voltage. There is no provision for power-down reset. If the external $\overline{RESET}$ pin is low at the end of the power-on delay time, the processor remains in the reset condition until $\overline{RESET}$ goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

The MC68HC11A4 MCU contains a watchdog timer which will time itself out if not reset within a specific time by a program reset sequence. If for any reason the COP watchdog timer is allowed to timeout, it generates an MCU reset which is functionally similar to pulling the $\overline{RESET}$ pin low.

A control bit, which is implemented in an EEPROM cell of the system configuration register, is used to enable (or disable) the COP reset function. When this bit is clear, the COP reset function is disabled; if set, the COP reset is enabled.

CLOCK MONITOR RESET

The MC68HC11A4 MCU contains a clock monitor circuit which measures the E-clock input frequency. If the E clock input rate is high enough, then the clock monitor does not time out. However, if the E clock signal is lost, or its frequency falls below 200 kHz, then an MCU reset is generated which is functionally similar to pulling the $\overline{RESET}$ pin low.

A read-write control bit, which is implemented in the system configuration options register, is used to enable (or disable) the clock monitor reset. When this bit is clear, the clock monitor reset function is disabled; when set, the clock monitor reset is enabled.

STOP AND WAIT

The MC68HC11A4 MCU contains two programmable low-power operating modes; stop and wait. In the wait mode, the on-chip oscillator remains active together with other functions discussed below. In the stop mode, all clocks including the crystal oscillator are stopped.

WAI (WAIT) INSTRUCTION

The WAI instruction places the MC68HC11A4 MCU in a low power consumption (wait) mode. In the wait mode, the internal clock remains active, and the MCU enters one of four different variations of the wait mode. These variations, which depend upon the I bit in the condition register and whether or not the COP circuit is required in the system, include: (1) only the CPU turned off; (2) CPU and the E clock output buffer turned off; (3) CPU and timer system turned off; or (4) CPU, E output, and timer system all off.

During the wait mode, the CPU registers are stacked and processing is suspended until a qualified interrupt is detected. The actual qualified interrupt type is dependent upon which of the wait mode variations is selected. The qualified interrupt(s) required to bring the MCU out of the wait mode for each of the wait mode variations is shown below. In all cases, reset brings the MCU out of the wait mode; however, as in all resets, the system is reset and the start of MCU operation is determined by the reset vector.

Wait Mode Variation	Qualified Interrupt
Only CPU Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$, Any Internal Interrupt
CPU and E Clock Output Buffers Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$, Any Internal Interrupt
CPU and Timer System Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$
CPU, E Clock Output Buffers, and Timer System Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$

STOP INSTRUCTION

The STOP instruction places the MC68HC11A4 MCU in its lowest power consumption mode provided the S bit in the condition code register is clear. In the stop mode all clocks including the internal oscillator are stopped, causing all internal processing to be halted. To exit the stop mode and resume normal processing, a low level must be applied to one of the external interrupt pins ($\overline{IRQ}$ or $\overline{XIRQ}$) or to the $\overline{RESET}$ pin. If an external interrupt is used at the $\overline{IRQ}$ input, it is only effective if the I bit in the condition code register is clear. If an external interrupt is applied at the $\overline{XIRQ}$ input, the MCU exits from the stop mode regardless of the state of the X bit in condition code register; however, the actual recovery sequence differs depending on the X bit. If the X bit is clear, the MCU starts up with the stacking sequence leading to normal service of the XIRQ request. If the X bit is set, then processing will continue with the instruction immediately following the STOP instruction and no $\overline{XIRQ}$ interrupt service routine is requested. As in the wait mode, a low input to the $\overline{RESET}$ pin will always result in an exit from the stop mode and the start of MCU operation is determined by the reset vector.

Since the oscillator is stopped in the stop mode, a restart delay may be required to allow for oscillator stabilization when exiting from the stop mode. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, a control bit within the MCU may be used (cleared) to bypass the delay. If the delay bypass control bit is clear then the $\overline{RESET}$ pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit and the restart delay will be imposed.

PROGRAMMABLE TIMER SYSTEM

The timer system in the MC68HC11A4 uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free running counter. The free running counter is clocked by the output of a programmable prescaler (divide-by-1, 4, 8, or 16) which is in turn clocked by the MCU E clock. Functions available within the MC68HC11A4 timer include: three input capture functions and five output compare functions.

The capabilities of the programmable timer are obtained using the following registers:

1. Prescaler (divide-by-1, 4, 8, or 16)
2. Free Running Counter (16-bit)
3. Input Capture (three 16-bit registers)
4. Output Compare (five 16-bit registers)
5. Main Timer Control and Status Registers

PRESCALER AND FREE RUNNING COUNTER

The key element in the timer system is a 16-bit free running counter with its associated programmable prescaler (divide-by-1, 4, 8, or 16). The free running counter is clocked by the output of the prescaler which is in turn clocked by the E clock. The free running counter can be read by software at any time without affecting its value since it is clocked and read on opposite half cycles of the MPU E clock. The free running counter is cleared to $0000 during reset and is a read-only register (except in the test or bootstrap mode where this feature is used in factory testing).

The 16-bit free running counter repeats every 65,536 counts (prescaler output) and when the count changes from $FFFF to $0000 a timer overflow flag bit is set. Setting the timer overflow flag bit also generates an internal interrupt if the overflow interrupt enable bit is set.

Input Capture Functions

There are three separate 16-bit read-only input capture registers which are not affected by reset. Each of these registers is used to latch the value of the free running counter when a selected transition at an external pin is detected. External devices provide the inputs on the PA0-PA2 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

Output Compare Functions

There are five separate 16-bit read/write output compare registers which are initialized to $FFFF at reset. The value written into the output compare register is compared to the free running counter value during each MCU E clock cycle. If a match is found between the two values, the particular output compare flag bit is set and an interrupt is generated provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For OC1, the output action to be taken, when a match is found, is controlled by a 5-bit mask register and a 5-bit data register. The 5-bit mask register specifies which timer port outputs are to be affected and the 5-bit data register specifies the data to be placed on the affected output pins. For OC2 through OC5, one specific timer output is affected as controlled by four 2-bit fields in a timer control register. Specific actions include: (1) timer disconnect from output pin logic, (2) toggle output compare line, (3) clear output compare line to zero, or (4) set output compare line to one.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes depending on the state of a control bit. These include the event counting mode or the gated time accumulation mode.

The pulse accumulator control register contains four bits which enable and configure the pulse accumulator system. One bit enables the counter. One bit determines whether the

PA7/PAI pin will be an input or an output. A third bit specifies the event counting mode or the gated time accumulation mode, and the fourth bit determines which edge of the PAI input is the active one. The 8-bit counter counts from $00 to $FF and when it overflows from $FF to $00 a flag bit is set. This results in a hardware interrupt provided the pulse accumulator overflow interrupt enable bit is set.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external (PAI) pin input (PA7). In the gated time accumulation mode, the 8-bit counter is clocked to increasing values by the MCU E clock (divided-by-64) provided the proper gating signal is applied to an external (PAI) pin input (PA7).

SERIAL COMMUNICATIONS INTERFACE (SCI)

The serial communications interface (SCI) allows the MC68HC11A4 to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI in the MC68HC11A4 is provided with a standard NRZ format with a variety of baud rates. The baud rate is derived from the crystal clock circuit and interface with peripheral devices is accomplished using port D pins. PD0 for receive data (RxD) and PD1 for transmit data (TxD).

BAUD RATE GENERATION

The actual baud rate generation circuit contains a programmable prescaler and divider which is clocked by the MCU E clock. A programmable baud rate register is used to provide the various divide ratios used in the baud rate generator prescaler and divider. This scheme of baud rate generation allows for selection of many different standard baud rates, all of which are controlled by the crystal oscillator.

DATA FORMAT

Receive data (RxD) in or transmit data (TxD) out is the serial data which is presented between the input pin (PD0) and the internal data bus, and between the internal data bus and the output pin (PD1). The data format requires:

1. An idle line which is in the high state (logic one) prior to transmission/reception of a message.
2. A start bit (logic zero) which is transmitted/received indicating the start of a message.
3. Data is transmitted and received least-significant bit first.
4. A stop bit (logic one in the tenth or eleventh bit position) indicates the byte is complete.
5. A break is defined as the transmission or reception of a logic zero for some multiple of the data format.

The data format word length may consist of either ten or eleven bits. Selection of the word length is controlled by a single bit in a control register within the SCI. If this control bit is clear, the data contains a start bit, eight data bits, and a stop bit. If this control bit is set, there is a start bit, nine data bits, and a stop bit.

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This is referred to as a double buffered system in that besides the character being shifted out serially, another character is already waiting to be loaded into the serial shift register. The output of the transmit serial shift-register is applied to the TxD output pin (PD1) as long as a transmit enable bit is set.

RECEIVE OPERATION

Receive data in (RxD) is serial data which is presented to the input pin (PD0). An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. In this manner the data input can be selectively sampled to detect receive data and then verify that the data is valid. Data is received in a serial shift register and is transferred to a parallel register as a complete byte. This is referred to as a double buffered system in that besides the character already in the parallel register, another is being shifted in serially.

WAKE-UP FEATURE

The wake-up feature allows a receiver(s) to "sleep" until a specific action takes place. In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of a message. This wake-up feature allows uninterested MPUs to ignore incoming messages. The MC68HC11A4 SCI permits this wake-up feature by either of two methods: idle line wake-up or address mark wake-up.

In idle line wake-up, all receivers wake up whenever an idle line is detected; however, if a receiver does not recognize its address in the first frame of a message it may ignore the rest of the message by invoking the wake-up feature. In this wake-up method, transmitter software must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

In the address mark wake-up, all serial frames consist of seven (or eight) information bits plus a most-significant bit (MSB) which is used to indicate an address frame if the MSB is a logic one. The first frame of each message is an address frame which wakes up all receivers in the system. All receivers evaluate this marked address frame to determine which receiver(s) the message is intended for. If a receiver determines that a message is not intended for it, it invokes the receiver wake-up function so that no additional program overhead is required for the rest of the message.

INTERRUPT FLAGS

The serial communications interface (SCI) generates a hardware interrupt (SCI interrupt) whenever any one of several flags is set and its corresponding interrupt enable bit is also set. These flags which are discussed below include:

1. Transmit Data register empty
2. Transmission complete
3. Idle line detected
4. Receive data register full or overrun error detected.

The transmit data register empty (TDRE) bit is set to indicate that the transmit parallel data register contents have been transferred to the transmit serial shift register. If the corresponding interrupt enable bit (transmit interrupt enable) is set then an SCI interrupt is generated.

The transmission complete (TC) bit is set when the transmitter no longer has any meaningful information to transmit; i.e., no data in the serial shifter, no queued preamble, and no queued break. If the transmitter is enabled when TC is set, the serial line will go idle (continuous mark).

The idle line detected (IDLE) bit is set whenever a receiver detects a receiver idle line. This could indicate the end of a message, the preamble of a new message, or resynchronization with the transmitter. If the corresponding interrupt enable bit (idle line interrupt enable) is set then an SCI interrupt is generated.

The receiver data register full (RDRF) bit is set whenever the receiver serial shift register contents are transferred to the serial communications data register. If the corresponding interrupt enable bit (receive interrupt enable) is set then an SCI interrupt is generated.

The overrun error bit is set to indicate that the next byte is ready for transfer from the receive shift register to the receive data register but that register is already full (RDRF bit set). Data transfer is then inhibited until the OR (overrun) bit is cleared. As with the RDRF bit, an SCI interrupt is generated if the corresponding interrupt enable bit is set.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) allows several MC68HC11A4 MCUs, or MC68HC11A4 MCUs plus peripheral devices, to be interconnected within a single "black box", on the same printed circuit board. In a serial peripheral interface, the MC68HC11A4 provides such features as:

- Full Duplex, Two, Three, or Four Wire Synchronous Transfers
- Master or Slave Operation
- Interface With Low Cost "Dumb" Peripherals
- Interface With Intelligent Peripherals on Master/Slave Basis
- Four Programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End of Transmission Interrupt Flag
- Write Collision Error Detection
- Master-Master Mode Fault Error Detection

Four basic signal lines are associated with the SPI system. These include a master-out-slave-in (MOSI) line; a master-in-slave-out (MISO) line; a serial clock (SCK) line; and a slave select ($\overline{SS}$) line. Two master-slave system configurations are shown in Figure 5 and the basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described below.

MASTER OUT SLAVE IN (MOSI)

The MOSI pin is configured as a data output in a master (mode) device and as a data input in a slave (mode) device. In this manner data is transferred serially from a master to a slave on this line; most significant bit first, least significant last.

MASTER IN SLAVE OUT (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line; most significant bit first, least significant last.

SLAVE SELECT ($\overline{SS}$)

The slave select ($\overline{SS}$) is a fixed input which receives an active low signal that is generated by a master device to enable slave devices to accept data.

SERIAL CLOCK (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI or MISO pins. The master and slave devices can exchange a byte of information during a sequence of eight clock pulses. The SCK is generated by the controlling master device and becomes an input on all slave devices to synchronize slave data transfer.

ANALOG-TO-DIGITAL (A/D) CONVERTER

The MC68HC11A4 contains an 8-channel, multiplexed input, successive approximation analog-to-digital converter with sample and hold. Two dedicated pins (V_{REFL}, V_{REFH}) are provided for the reference supply voltage input. These dedicated pins are used instead of the device power pins to increase accuracy of the A/D conversion.

The 8-bit A/D conversions of the MC68HC11A4 are accurate to within ± one LSB (± ½ LSB quantizing error and ± ½ LSB non-linearity error). Each conversion is accomplished in 50 MCU E clock cycles or less. An internal control bit allows selection of an internal conversion clock oscillator which allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 25 to 50 microseconds to complete.

NOTE

In the 48-pin dual in-line package, four conversion channels are not implemented. These include channels four through seven.

ADDRESSING MODES

Six addressing modes can be used to reference memory; they include: immediate, direct, extended, indexed (with either of two 16-bit index registers and an 8-bit offset), inherent, and relative. Some instructions require an additional byte before the opcode to accommodate a multi-page 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 is to proceed.

FIGURE 5 - MASTER-SLAVE SYSTEM CONFIGURATION
(Sheet 1 of 2)

a. Single Master, Four Slaves

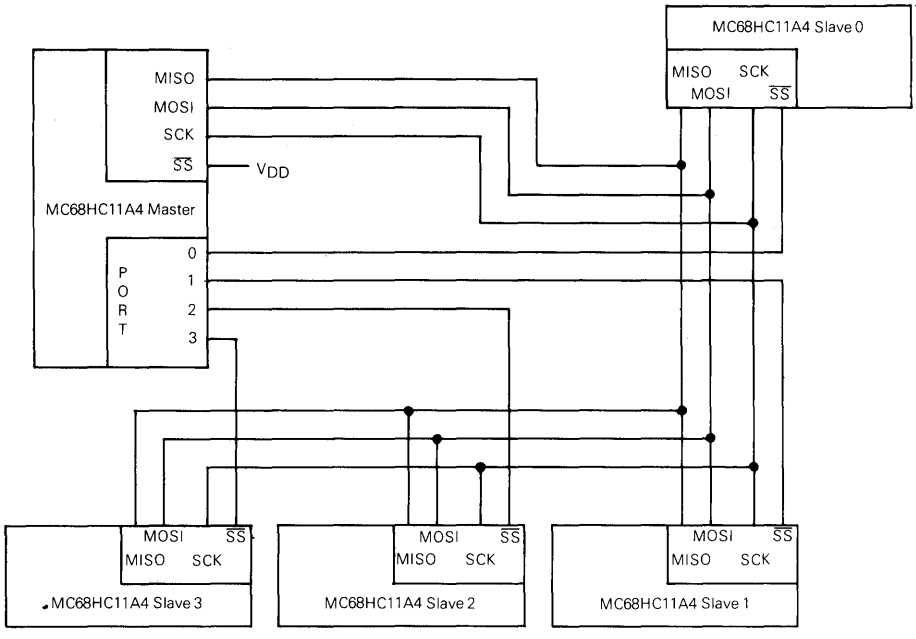
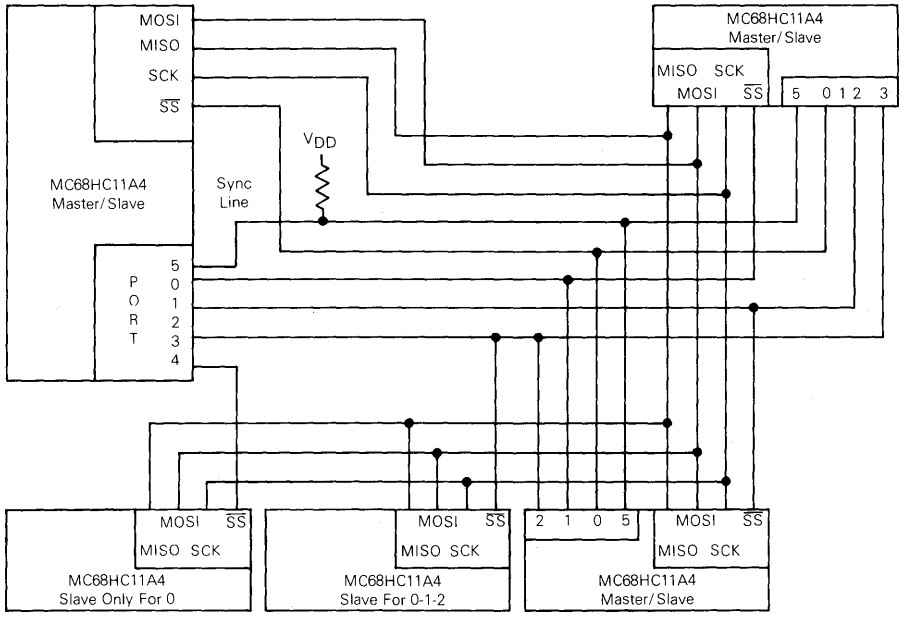


FIGURE 5 - MASTER-SLAVE SYSTEM CONFIGURATION
(Sheet 2 of 2)

b. Three Master/Slave, Two Slaves



3

IMMEDIATE ADDRESSING

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

DIRECT ADDRESSING

In the direct addressing mode, the least significant byte of the operand address is contained in a single byte following the opcode and the most significant byte is assumed to be $00. Direct addressing allows the user to access addresses $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. These are usually two or three (if prebyte is required) byte instructions.

EXTENDED ADDRESSING

In the extended addressing mode, the second and third bytes following the opcode contain the absolute address of the operand. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED ADDRESSING

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case the effective address is variable and depends on two factors: (1) the 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. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

INHERENT ADDRESSING

In the inherent addressing mode, all of the information to execute the instruction is contained in the opcode. The 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 and contents of the 8-bit signed byte following the opcode (the offset) is added to the contents of the program counter 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 used in the MC68HC11A4, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. The 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. Refer to **INSTRUCTION SUMMARY** for more detail.

INSTRUCTION SET

The central processing unit (CPU) in the MC68HC11A4 is basically a proper extension of the MC6801 CPU. In addition to its ability to execute all M6800 and M6801 instructions, the MC68HC11A4 CPU has a paged operation code (opcode) map with a total of 91 new opcodes. Major functional additions include a second 16-bit index register (Y register), two types of 16-by-16 divide instructions, a STOP instruction, and bit manipulation instructions.

Table 3 shows all MC68HC11A4 instructions in all possible addressing modes. For each instruction the operand construction is shown as well as the total number of machine code bytes and execution time in CPU E-clock cycles. Notes are provided at the end of Table 3 which explain the letters in the Operand and Execution Time columns of some instructions.

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
ABA	INH	1B		1	2
ABX	INH	3A		1	3
ABY	INH	18 3A		2	4
ADCA (opr)	A IMM	89	ii	2	2
	A DIR	99	dd	2	3
	A EXT	B9	hh ll	3	4
	A IND, X	A9	ff	2	4
	A IND, Y	18 A9	ff	3	5
ADCB (opr)	B IMM	C9	ii	2	2
	B DIR	D9	dd	2	3
	B EXT	F9	hh ll	3	4
	B IND, X	E9	ff	2	4
	B IND, Y	18 E9	ff	3	5
ADDA (opr)	A IMM	8B	ii	2	2
	A DIR	9B	dd	2	3
	A EXT	BB	hh ll	3	4
	A IND, X	AB	ff	2	4
	A IND, Y	18 AB	ff	3	5
ADDB (opr)	B IMM	CB	ii	2	2
	B DIR	DB	dd	2	3
	B EXT	FB	hh ll	3	4
	B IND, X	EB	ff	2	4
	B IND, Y	18 EB	ff	3	5
ADDD (opr)	IMM	C3	jj kk	3	4
	DIR	D3	dd	2	5
	EXT	F3	hh ll	3	6
	IND, X	E3	ff	2	6
	IND, Y	18 E3	ff	3	7
ANDA (opr)	A IMM	84	ii	2	2
	A DIR	94	dd	2	3
	A EXT	B4	hh ll	3	4
	A IND, X	A4	ff	2	4
	A IND, Y	18 A4	ff	3	5
ANDB (opr)	B IMM	C4	ii	2	2
	B DIR	D4	dd	2	3
	B EXT	F4	hh ll	3	4
	B IND, X	E4	ff	2	4
	B IND, Y	18 E4	ff	3	5
ASL (opr)	EXT	78	hh ll	3	6
	IND, X	68	ff	2	6
	IND, Y	18 68	ff	3	7
ASLA	A INH	48		1	2
ASLB	B INH	58		1	2
ASLD	INH	05		1	3
ASR (opr)	EXT	77	hh ll	3	6
	IND, X	67	ff	2	6
	IND, Y	18 67	ff	3	7
ASRA	A INH	47		1	2
ASRB	B INH	57		1	2
BCC (rel)	REL	24	rr	2	3
BCLR (opr) (msk)	DIR	15	dd mm	3	6
	IND, X	1D	ff mm	3	7
	IND, Y	18 1D	ff mm	4	8
BCS (rel)	REL	25	rr	2	3
BEQ (rel)	REL	27	rr	2	3
BGE (rel)	REL	2C	rr	2	3
BGT (rel)	REL	2E	rr	2	3

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
BHI (rel)	REL	22	rr	2	3
BHS (rel)	REL	24	rr	2	3
BITA (opr)	A IMM	85	ii	2	2
	A DIR	95	dd	2	3
	A EXT	B5	hh ll	3	4
	A IND, X	A5	ff	2	4
	A IND, Y	18 A5	ff	3	5
BITB (opr)	B IMM	C5	ii	2	2
	B DIR	D5	dd	2	3
	B EXT	F5	hh ll	3	4
	B IND, X	E5	ff	2	4
	B IND, Y	18 E5	ff	3	5
BLE (rel)	REL	2F	rr	2	3
BLO (rel)	REL	25	rr	2	3
BLS (rel)	REL	23	rr	2	3
BLT (rel)	REL	2D	rr	2	3
BMI (rel)	REL	2B	rr	2	3
BNE (rel)	REL	26	rr	2	3
BPL (rel)	REL	2A	rr	2	3
BRA (rel)	REL	20	rr	2	3
BRN (rel)	REL	21	rr	2	3
BRCLR (opr)	DIR	13	dd mm rr	4	6
(msk)	IND, X	1F	ff mm rr	4	7
(rel)	IND, Y	18 1F	ff mm rr	5	8
BRSET (opr)	DIR	12	dd mm rr	4	6
(msk)	IND, X	1E	ff mm rr	4	7
(rel)	IND, Y	18 1E	ff mm rr	5	8
BSET (opr) (msk)	DIR	14	dd mm	3	6
	IND, X	1C	ff mm	3	7
	IND, Y	18 1C	ff mm	4	8
BSR (rel)	REL	8D	rr	2	6
BVC (rel)	REL	28	rr	2	3
BVS (rel)	REL	29	rr	2	3
CBA	INH	11		1	2
CLC	INH	0C		1	2
CLI	INH	0E		1	2
CLR (opr)	EXT	7F	hh ll	3	6
	IND, X	6F	ff	2	6
	IND, Y	18 6F	ff	3	7
CLRA	A INH	4F		1	2
CLRB	B INH	5F		1	2
CLV	INH	0A		1	2
CMPA (opr)	A IMM	81	ii	2	2
	A DIR	91	dd	2	3
	A EXT	B1	hh ll	3	4
	A IND, X	A1	ff	2	4
	A IND, Y	18 A1	ff	3	5
CMPB (opr)	B IMM	C1	ii	2	2
	B DIR	D1	dd	2	3
	B EXT	F1	hh ll	3	4
	B IND, X	E1	ff	2	4
	B IND, Y	18 E1	ff	3	5

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
COM (opr)	EXT	73	hh ll	3	6
	IND, X	63	ff	2	6
	IND, Y	18 63	ff	3	7
COMA	A INH	43		1	2
COMB	B INH	53		1	2
CPD (opr)	IMM	1A 83	jj kk	4	5
	DIR	1A 93	dd	3	6
	EXT	1A B3	hh ll	4	7
	IND, X	1A A3	ff	3	7
	IND, Y	CD A3	ff	3	7
CPX (opr)	IMM	8C	jj kk	3	4
	DIR	9C	dd	2	5
	EXT	BC	hh ll	3	6
	IND, X	AC	ff	2	6
	IND, Y	CD AC	ff	3	7
CPY (opr)	IMM	18 8C	jj kk	4	5
	DIR	18 9C	dd	3	6
	EXT	18 BC	hh ll	4	7
	IND, X	1A AC	ff	3	7
	IND, Y	18 AC	ff	3	7
DAA	INH	19		1	2
DEC (opr)	EXT	7A	hh ll	3	6
	IND, X	6A	ff	2	6
	IND, Y	18 6A	ff	3	7
DECA	A INH	4A		1	2
DECB	B INH	5A		1	2
DES	INH	34		1	3
DEX	INH	09		1	3
DEY	INH	18 09		2	4
EORA (opr)	A IMM	88	ii	2	2
	A DIR	98	dd	2	3
	A EXT	B8	hh ll	3	4
	A IND, X	A8	ff	2	4
	A IND, Y	18 A8	ff	3	5
EORB (opr)	B IMM	C8	ii	2	2
	B DIR	D8	dd	2	3
	B EXT	F8	hh ll	3	4
	B IND, X	E8	ff	2	4
	B IND, Y	18 E8	ff	3	5
FDIV	INH	03		1	41
IDIV	INH	02		1	41
INC (opr)	EXT	7C	hh ll	3	6
	IND, X	6C	ff	2	6
	IND, Y	18 6C	ff	3	7
INCA	A INH	4C		1	2
INCB	B INH	5C		1	2
INS	INH	31		1	3
INX	INH	08		1	3
INY	INH	18 08		2	4
JMP (opr)	EXT	7E	hh ll	3	3
	IND, X	6E	ff	2	3
	IND, Y	18 6E	ff	3	4
JSR (opr)	DIR	9D	dd	2	5
	EXT	BD	hh ll	3	6
	IND, X	AD	ff	2	6
	IND, Y	18 AD	ff	3	7

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
LDAA (opr)	A IMM	86	ii	2	2
	A DIR	96	dd	2	3
	A EXT	B6	hh ll	3	4
	A IND, X	A6	ff	2	4
	A IND, Y	18 A6	ff	3	5
LDAB (opr)	B IMM	C6	ii	2	2
	B DIR	D6	dd	2	3
	B EXT	F6	hh ll	3	4
	B IND, X	E6	ff	2	4
	B IND, Y	18 E6	ff	3	5
LDD (opr)	IMM	CC	jj kk	3	3
	DIR	DC	dd	2	4
	EXT	FC	hh ll	3	5
	IND, X	EC	ff	2	5
	IND, Y	18 EC	ff	3	6
LDS (opr)	IMM	8E	jj kk	3	3
	DIR	9E	dd	2	4
	EXT	BE	hh ll	3	5
	IND, X	AE	ff	2	5
	IND, Y	18 AE	ff	3	6
LDX (opr)	IMM	CE	jj kk	3	3
	DIR	DE	dd	2	4
	EXT	FE	hh ll	3	5
	IND, X	EE	ff	2	5
	IND, Y	CD EE	ff	3	6
LDY (opr)	IMM	18 CE	jj kk	4	4
	DIR	18 DE	dd	3	5
	EXT	18 FE	hh ll	4	6
	IND, X	1A EE	ff	3	6
	IND, Y	18 EE	ff	3	6
LSL (opr)	EXT	78	hh ll	3	6
	IND, X	68	ff	2	6
	IND, Y	18 68	ff	3	7
LSLA	A INH	48		1	2
LSLB	B INH	58		1	2
LSLD	INH	05		1	3
LSR (opr)	EXT	74	hh ll	3	6
	IND, X	64	ff	2	6
	IND, Y	18 64	ff	3	7
LSRA	A INH	44		1	2
LSRB	B INH	54		1	2
LSRD	INH	04		1	3
MUL	INH	3D		1	10
NEG (opr)	EXT	70	hh ll	3	6
	IND, X	60	ff	2	6
	IND, Y	18 60	ff	3	7
NEGA	A INH	40		1	2
NEGB	B INH	50		1	2
NOP	INH	01		1	2
ORAA (opr)	A IMM	8A	ii	2	2
	A DIR	9A	dd	2	3
	A EXT	BA	hh ll	3	4
	A IND, X	AA	ff	2	4
	A IND, Y	18 AA	ff	3	5
ORAB (opr)	B IMM	CA	ii	2	2
	B DIR	DA	dd	2	3
	B EXT	FA	hh ll	3	4
	B IND, X	EA	ff	2	4
	B IND, Y	18 EA	ff	3	5

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
PSHA	A INH	36		1	3
PSHB	B INH	37		1	3
PSHX	INH	3C		1	4
PSHY	INH	18 3C		2	5
PULA	A INH	32		1	4
PULB	B INH	33		1	4
PULX	INH	38		1	5
PULY	INH	18 38		2	6
ROL (opr)	EXT	79	hh ll	3	6
	IND, X	69	ff	2	6
	IND, Y	18 69	ff	3	7
ROLA	A INH	49		1	2
ROLB	B INH	59		1	2
ROR (opr)	EXT	76	hh ll	3	6
	IND, X	66	ff	2	6
	IND, Y	18 66	ff	3	7
RORA	A INH	46		1	2
RORB	B INH	56		1	2
RTI	INH	3B		1	12
RTS	INH	39		1	5
SBA	INH	10		1	2
SBCA (opr)	A IMM	82	ii	2	2
	A DIR	92	dd	2	3
	A EXT	B2	hh ll	3	4
	A IND, X	A2	ff	2	4
	A IND, Y	18 A2	ff	3	5
SBCB (opr)	B IMM	C2	ii	2	2
	B DIR	D2	dd	2	3
	B EXT	F2	hh ll	3	4
	B IND, X	E2	ff	2	4
	B IND, Y	18 E2	ff	3	5
SEC	INH	0D		1	2
SEI	INH	0F		1	2
SEV	INH	0B		1	2
STAA (opr)	A DIR	97	dd	2	3
	A EXT	B7	hh ll	3	4
	A IND, X	A7	ff	2	4
	A IND, Y	18 A7	ff	3	5
STAB (opr)	B DIR	D7	dd	2	3
	B EXT	F7	hh ll	3	4
	B IND, X	E7	ff	2	4
	B IND, Y	18 E7	ff	3	5
STD (opr)	DIR	DD	dd	2	4
	EXT	FD	hh ll	3	5
	IND, X	ED	ff	2	5
	IND, Y	18 ED	ff	3	6

3

TABLE 3 – MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
STOP	INH	CF		1	2
STS (opr)	DIR	9F	dd	2	4
	EXT	BF	hh ll	3	5
	IND, X	AF	ff	2	5
	IND, Y	18 AF	ff	3	6
STX (opr)	DIR	DF	dd	2	4
	EXT	FF	hh ll	3	5
	IND, X	EF	ff	2	5
	IND, Y	CD EF	ff	3	6
STY (opr)	DIR	18 DF	dd	3	5
	EXT	18 FF	hh ll	4	6
	IND, X	1A EF	ff	3	6
	IND, Y	18 EF	ff	3	6
SUBA (opr)	A IMM	80	ii	2	2
	A DIR	90	dd	2	3
	A EXT	B0	hh ll	3	4
	A IND, X	A0	ff	2	4
	A IND, Y	18 A0	ff	3	5
SUBB (opr)	B IMM	C0	ii	2	2
	B DIR	D0	dd	2	3
	B EXT	F0	hh ll	3	4
	B IND, X	E0	ff	2	4
	B IND, Y	18 E0	ff	3	5
SUBD (opr)	IMM	83	jj kk	3	4
	DIR	93	dd	2	5
	EXT	B3	hh ll	3	6
	IND, X	A3	ff	2	6
	IND, Y	18 A3	ff	3	7
SWI	INH	3F		1	14
TAB	INH	16		1	2
TAP	INH	06		1	2
TBA	INH	17		1	2
TEST	INH	00		1	*
TPA	INH	07		1	2
TST (opr)	EXT	7D	hh ll	3	6
	IND, X	6D	ff	2	6
	IND, Y	18 6D	ff	3	7
TSTA	A INH	4D		1	2
TSTB	B INH	5D		1	2
TSX	INH	30		1	3
TSY	INH	18 30		2	4
TXS	INH	35		1	3
TYS	INH	18 35		2	4
WAI	INH	3E		2	14+n**
XGDX	INH	8F		1	3
XGDY	INH	18 8F		2	4

*-infinity or until reset occurs.

**-12 cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycles (n) until an interrupt is recognized. Finally two additional cycles are used to fetch the appropriate interrupt vector.

dd = 8-bit direct address ($0000-$00FF) (high byte assumed to be $00)
ff = 8-bit positive offset $00 (0) to $FF (255) (is added to index)
hh = high order byte of 16-bit extended address
ii = one byte of immediate data
jj = high order byte of 16-bit immediate data
kk = low order byte of 16-bit immediate data
ll = low order byte of 16-bit extended address
mm = 1-byte bit mask (set bits to be affected)
rr = signed relative offset $80 (– 128) to $7F (+ 127) (offset relative to the address following the machine code offset byte)