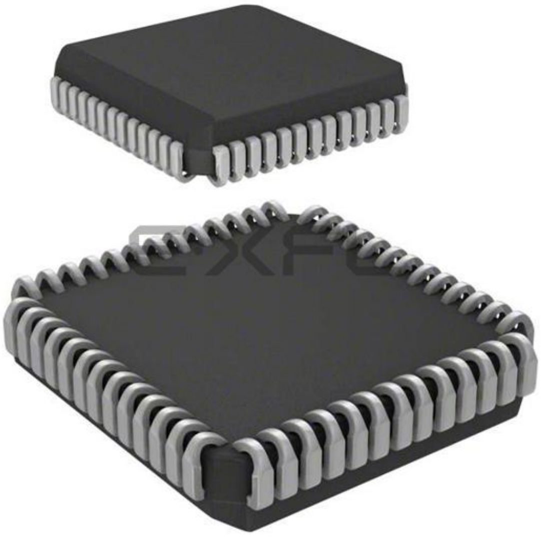
NXP USA Inc. - MC68HC11A0CFNE3 Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC11
Core Size	8-Bit
Speed	3MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	38
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc11a0cfne3

Table 2-2 Port Signal Summary

Port-Bit	Single Chip and Bootstrap Mode	Expanded Multiplexed 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/OC1	PA3/OC5/OC1
A-4	PA4/OC4/OC1	PA4/OC4/OC1
A-5	PA5/OC3/OC1	PA5/OC3/OC1
A-6	PA6/OC2/OC1	PA6/OC2/OC1
A-7	PA7/PAI/OC1	PA7/PAI/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
	STRA	AS
	STRB	R/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 versions

2.2.4 Additional Boot Loader Program Options

The user may transmit a $55 (only at E clock/16) as the first character rather than the normal $FF. This will cause the program to jump directly to location $0000, skipping the download.

The user may tie the receiver to the transmitter (with an external pull-up resistor). This will cause the program to jump directly to the beginning of EEPROM ($B600). Another way to cause the program to jump directly to EEPROM is to transmit either a break or $00 as the first character rather than the normal $FF.

Note that none of these options bypass the security check and so do not compromise those customers using security.

Keep in mind that upon entry to the downloaded program at location $0000, some registers have been changed from their reset states. The SCI transmitter and receiver are enabled which cause port D pins 0 and 1 to be dedicated to SCI use. Also port D is configured for wired-OR operation. It may be necessary for the user to write to the SCCR2 and SPCR registers to disable the SCI and/or port D wire-OR operation.

2.2.5 Special Test Operating Mode

The test mode is a special operating mode intended primarily for factory testing. This mode is very similar to the expanded multiplexed operating mode. In special test operating mode, the reset and interrupt vectors are fetched from external memory locations $BFC0–$BFFF rather than $FFC0–$FFFF. There are no time limits for protection of the TMSK2, OPTION, and INIT registers, so these registers may be written repeatedly. Also a special TEST1 register is enabled which allows several factory test functions to be invoked.

The special test operating mode is not recommended for use by an end user because of the reduced system security; however, an end user may wish to come out of reset in special test operating mode. Then, after some initialization, the SMOD and MDA bits could be rewritten to select a normal operating mode to re-enable the protection features.

Note that if the RAM is relocated to either $E000 or $F000, which is in conflict with the internal ROM, (no conflict if the ROMON bit in the configuration register is zero), RAM will take priority and the conflicting ROM will become inaccessible. Also, if the 64 control registers are relocated so that they conflict with the RAM and/or ROM, then the 64 control registers take priority and the RAM and/or ROM at those locations become inaccessible. No harmful conflicts result, the lower priority resources simply become inaccessible. Similarly, if an internal resource conflicts with an external device no harmful conflict results. Data from the external device will not be applied to the internal data bus and cannot interfere with the internal read.

Note that there are unused register locations in the 64 byte control register block. Reads of these unused registers will return data from the undriven internal data bus and not from another resource that happens to be located at the same address.

3.3 ROM

The internal 8K ROM occupies the highest 8K of the memory map ($E000–$FFFF). This ROM is disabled when the ROMON bit in the CONFIG register is clear. The ROMON bit is implemented with an EEPROM cell and is programmed using the same procedures for programming the on-chip EEPROM. For further information refer to **3.5.3 System Configuration Register (CONFIG)**.

In the single-chip operating mode, internal ROM is enabled regardless of the state of the ROMON bit.

There is also a 192 byte mask programmed boot ROM in the MC68HC11A8. This bootstrap program ROM controls the operation of the special bootstrap operating mode and is only enabled following reset in the special bootstrap operating mode. For more information refer to **2.2.3 Special Bootstrap Operating Mode.**

3.4 RAM

The 256 byte internal RAM may be relocated during initialization by writing to the INIT register. The reset default position is $0000 through $00FF. This RAM is implemented with static cells and retains its contents during the WAIT and STOP modes.

The contents of the 256-byte RAM can also be retained by supplying a low current backup power source to the MODB/V_{STBY} pin. When using a standby power source, V_{DD} may be removed; however, $\overline{\text{RESET}}$ must go low before V_{DD} is removed and remain low until V_{DD} has been restored.

3.5 EEPROM

The 512 bytes of EEPROM are located at $B600 through $B7FF and have the same read cycle time as the internal ROM. The write (or programming) mechanism for the EEPROM is controlled by the PPROG register. The EEPROM is disabled when the EEON bit in the CONFIG register is zero. The EEON bit is implemented with an EEPROM cell.

The erased state of an EEPROM byte is $FF. Programming changes ones to zeros. If any bit in a location needs to be changed from a zero to a one, the byte must be erased in a separate operation before it is reprogrammed. If a new data byte has no ones in bit positions which were already programmed to zero, it is acceptable to program the new data without erasing the EEPROM byte first. For example, programming $50 to a location which was already $55 would change the location to $50.

Programming and erasure of the EEPROM relies on an internal high-voltage charge pump. At E clock frequencies below 2 MHz the efficiency of this charge pump decreases which increases the time required to program or erase a location. The recommended program and erase time is 10 milliseconds when the E clock is 2 MHz and should be increased to as much as 20 milliseconds when E is between 1 MHz and 2 MHz. When the E clock is below 1 MHz, the clock source for the charge pump should be switched from the system clock to an on-chip R-C oscillator clock. This is done by setting the CSEL bit in the OPTION register. A 10 millisecond period should be allowed after setting the CSEL bit to allow the charge pump to stabilize. Note that the CSEL bit also controls a clock to the analog-to-digital converter subsystem.

3.5.1 EEPROM Programming Control Register (PPROG)

This 8-bit register is used to control programming and erasure of the 512-byte EEPROM. Reset clears this register so the EEPROM is configured for normal reads.

	7	6	5	4	3	2	1	0	
$103B	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
RESET	0	0	0	0	0	0	0	0	

ODD — Program Odd Rows (TEST)

EVEN — Program Even Rows (TEST)

Bit 5 — Not implemented.
This bit always reads zero.

BYTE — Byte Erase Select
This bit overrides the ROW bit.
0 = Row or Bulk Erase
1 = Erase Only One Byte

ROW — Row Erase Select
If the BYTE bit is 1, ROW has no meaning.
0 = Bulk Erase
1 = Row Erase

ERASE — Erase Mode Select
0 = Normal Read or Program
1 = Erase Mode

4.3 Simple Strobed I/O

The simple strobed mode of parallel I/O is invoked and controlled by the parallel I/O control register (PIOC). This mode is selected when the handshake bit (HNDS) in the PIOC register is clear. Port C becomes a strobed input port with the STRA line as the edge-detecting latch command input. Also, port B becomes a strobed output port with the STRB line as the output strobe. The logic sense of the STRB output is selected by the invert strobe B bit (INVB) in the PIOC register.

4.3.1 Strobed Input Port C

In this mode, there are two addresses where port C may be read, the PORTC data register and the alternate latched port C register (PORTCL). The data direction register still controls the data direction of all port C lines. Even when the strobed input mode is selected, any or all of the port C lines may still be used for general purpose I/O.

The STRA line is used as an edge-detecting input, and the edge-select for strobe A (EGA) bit in the PIOC register defines either falling or rising edge as the significant edge. Whenever the selected edge is detected at the STRA pin, the current logic levels at port C lines are latched into the PORTCL register and the strobe A flag (STAF) in the PIOC register is set. If the strobe A interrupt enable (STAI) bit in PIOC is also set, an internal interrupt sequence is requested. The strobe A flag (STAF) is automatically cleared by reading the PIOC register (with STAF set) followed by a read of the PORTCL register. Data is latched in the PORTCL register whether or not the STAF flag was previously clear.

4.3.2 Strobed Output Port B

In this mode, the STRB pin is a strobe output which is pulsed for two E clock periods each time there is a write to port B. The INVB bit in the PIOC register controls the polarity of the pulse on the STRB line.

4.4 Full Handshake I/O

The full handshake modes of parallel I/O involve port C and the STRA and STRB lines. There are two basic modes (input and output) and an additional variation on the output handshake mode that allows three-stated operation of port C. In all handshake modes, STRA is an edge-detecting input, and STRB is a handshake output line.

When full input handshake protocol is specified, both general purpose input and/or general purpose output can coexist at port C. When full output handshake protocol is specified, general purpose output can coexist with the handshake outputs at port C, but the three-state feature of the output handshake mode interferes with general purpose input in two ways. First, in full output handshake, the port C lines are outputs whenever STRA is at its active level regardless of the data direction register bits. This potentially conflicts with any external device trying to drive port C unless that external device has an open-drain type output driver. Second, the value returned on reads of port C is the state of the outputs of an internal port C output latch regardless of the states of the data direction register bits, so that the data written for output handshake can be read even if the pins are in a three-state condition.

STAF — Strobe A Interrupt Status Flag

This bit is set when a selected edge occurs on strobe A. Clearing it depends on the state of HNDS and OIN bits. In simple strobed mode or in full input handshake mode, STAF is cleared by reading the PIOC register with STAF set followed by reading the PORTCL register. In output handshake, STAF is cleared by reading the PIOC register with STAF set followed by writing to the PORTCL register.

STAI — Strobe A Interrupt Enable Mask

When the I bit in the condition code register is clear and STAI is set, STAF (when set) will request an interrupt.

CWOM — Port C Wire-OR Mode

CWOM affects all eight port C pins together

0 = Port C outputs are normal CMOS outputs
1 = Port C outputs act as open-drain outputs

HNDS — Handshake Mode

When clear, strobe A acts as a simple input strobe to latch data into PORTCL, and strobe B acts as a simple output strobe which pulses after a write to port B. When set, a handshake protocol involving port C, STRA, and STRB is selected (see the definition for the OIN bit).

0 = Simple strobe mode
1 = Full input or output handshake mode

OIN — Output or Input Handshaking

This bit has no meaning when HNDS = 0.

0 = Input handshake
1 = Output handshake

PLS — Pulse/Interlocked Handshake Operation

This bit has no meaning if HNDS = 0. When interlocked handshake operation is selected, strobe B, once activated, stays active until the selected edge of strobe A is detected. When pulsed handshake operation is selected, strobe B is pulsed for two E cycles.

0 = Interlocked handshake select
1 = Pulsed handshake selected

EGA — Active Edge for Strobe A

0 = Falling edge of STRA is selected. When output handshake is selected, port C lines obey the data direction register while STRA is low, but port C is forced to output when STRA is high.
1 = Rising edge of STRA is selected. When output handshake is selected, port C lines obey the data direction register while STRA is high, but port C is forced to output when STRA is low.

INVB — Invert Strobe B

0 = Active level is logic zero
1 = Active level is logic one

TCIE — Transmit Complete Interrupt Enable
0 = TC interrupts disabled
1 = SCI Interrupt if TC = 1

RIE — Receive Interrupt Enable
0 = RDRF and OR interrupts disabled
1 = SCI interrupt if RDRF or OR = 1

ILIE — Idle Line Interrupt Enable
0 = IDLE interrupts disabled
1 = SCI interrupt if IDLE = 1

TE — Transmit Enable

When the transmit enable bit is set, the transmit shift register output is applied to the TxD line. Depending on the state of control bit M (SCCR1), a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state. After loading the last byte in the serial communications data register and receiving the TDRE flag, the user can clear TE. Transmission of the last byte will then be completed before the transmitter gives up control of the TxD pin. While the transmitter is active, the data direction register control for port D bit 1 is overridden and the line is forced to be an output.

RE — Receive Enable

When the receive enable bit is set, the receiver is enabled. When RE is clear, the receiver is disabled and all of the status bits associated with the receiver (RDRF, IDLE, OR, NF, and FE) are inhibited. While the receiver is enabled, the data direction register control for port D bit 0 is overridden and the line is forced to be an input.

RWU — Receiver Wake Up

When the receiver wake-up bit is set by the user’s software, it puts the receiver to sleep and enables the “wake up” function. If the WAKE bit is cleared, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If the WAKE bit is set, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If the send break bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or sending data. If SBK remains set, the transmitter will continually send whole blocks of zeros (sets of 10 or 11) until cleared. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. If the transmitter is currently empty and idle, setting and clearing SBK is likely to queue two character times of break because the first break transfers almost immediately to the shift register and the second is then queued into the parallel transmit buffer.

6 SERIAL PERIPHERAL INTERFACE

This section contains a description on the serial peripheral interface (SPI).

6.1 Overview and Features

The serial peripheral interface (SPI) is a synchronous interface which allows several SPI microcontrollers or SPI-type peripherals to be interconnected. In a serial peripheral interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. The MC68HC11A8 SPI system may be configured either as a master or as a slave. The SPI contains the following features:

- Full Duplex, Three-Wire Synchronous Transfers
- Master or Slave Operation
- 1.5 MHz (Maximum) Master Bit Frequency
- 3 MHz (Maximum) Slave Bit Frequency
- Four Programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End-of-Transmission Interrupt Flag
- Write Collision Flag Protection
- Master-Master Mode Fault Protection
- Easily Interfaces to Simple Expansion Parts (PLLs, D/As, Latches, Display Drivers, etc.)

6.2 SPI Signal Descriptions

The four basic SPI signals (MISO, MOSI, SCK, and $\overline{\text{SS}}$) are discussed in the following paragraphs. Each signal is described for both the master and slave modes.

Any SPI output line has to have its corresponding data direction register bit set. If this bit is clear, the line is disconnected from the SPI logic and becomes a general-purpose input line. Any SPI input line is forced to act as an input regardless of what is in the corresponding data direction register bit.

6.2.1 Master In Slave Out (MISO)

The MISO line is configured as an input in a master device and as an output in a slave device. It is one of the two lines that transfer serial data in one direction, with the most significant bit sent first. The MISO line of a slave device is placed in the high-impedance state if the slave is not selected.

6.2.2 Master Out Slave In (MOSI)

The MOSI line is configured as an output in a master device and as an input in a slave device. It is one of the two lines that transfer serial data in one direction with the most significant bit sent first.

PAMOD — Pulse Accumulator Mode

0 = External event counting
1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control

This bit has different meanings depending on the state of the PAMOD bit.

PAMOD	PEDGE	Action on Clock
0	0	PAI Falling Edge Increments the Counter
0	1	PAI Rising Edge Increments the Counter
1	0	A zero on PAI Inhibits Counting
1	1	A one on PAI Inhibits Counting

Bits 3-2 — Not Implemented

These bits always read zero.

RTR1 and RTR0 — RTI Interrupt Rate Selects

These two bits select one of four rates for the real time periodic interrupt circuit (see **Table 8-1**). Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

Table 8-1 Real Time Interrupt Rate versus RTR1 and RTR0

RTR1	RTR0	Rate	XTAL = 12.0 MHz	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	$2^{13} \div E$	8.192 ms	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	$2^{14} \div E$	16.384 ms	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	$2^{15} \div E$	32.768 ms	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	$2^{16} \div E$	65.536 ms	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	**3.0 MHz**	**2.1 MHz**	**2.0 MHz**	**1.2288 MHz**	**1.0 MHz**	**921.6 kHz**

9.1.4 Clock Monitor Reset

The clock monitor function is enabled by the CME control bit in the OPTION register. When CME is clear, the monitor function is disabled. When the CME bit is set, the clock monitor function detects the absence of an E clock for more than a certain period of time. The timeout period is dependent on processing parameters and will be between 5 and 100 microseconds. This means that an E-clock rate of 200 kHz or more will never cause a clock monitor failure and an E-clock rate of 10 kHz or less will definitely cause a clock monitor failure. This implies that systems operating near or below an E-clock rate of 200 kHz should not use the clock monitor function.

Upon detection of a slow or absent clock, the clock monitor circuit will cause a system reset. This reset is issued to the external system via the bidirectional $\overline{\text{RESET}}$ pin. The clock monitor system has a separate reset vector.

Special considerations are needed when using a STOP function and clock monitor in the same system. Since the STOP function causes the clocks to be halted, the clock monitor function will generate a reset sequence if it is enabled at the time the STOP mode is entered.

The clock monitor is useful as a backup for the COP watchdog timer. Since the watchdog timer requires a clock to function, it will not indicate any failure if the system clocks fail. The clock monitor would detect such a failure and force the MCU to its reset state. Note that clocks are not required for the MCU to reach its reset configuration, although clocks are required to sequence through reset back to the run condition.

9.1.5 Configuration Options Register (OPTION)

This is a special purpose 8-bit register that is used (optionally) during initialization to configure internal system configuration options. With the exception of bits 7, 6, and 3 (ADPU, CSEL, and CME) which may be read or written at any time, this register may be written to only once after a reset and thereafter is a read-only register. If no write is performed to this location within 64 E-clock cycles after reset, then bits 5, 4, 1, and 0 (IRQE, DLY, CR1, and CR0) will become read-only to minimize the possibility of any accidental changes to the system configuration (writes will be ignored). While in special test modes, the protection mechanism on this register is preempted and all bits in the OPTION register may be written repeatedly.

	7	6	5	4	3	2	1	0	
$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
RESET	0	0	0	1	0	0	0	0	

ADPU — A/D Power-up

This bit controls operations of the on-chip analog-to-digital converter. When ADPU is clear, the A/D system is powered down and conversion requests will not return meaningful information. To use the A/D system, this bit should be set. A 100 microsecond delay is required after ADPU is turned on to allow the A/D system to stabilize.

into two basic categories, maskable and non-maskable. In the MC68HC11A8 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 local control bits.

Table 9-2 IRQ Vector Interrupts

Interrupt Cause	Local Mask
External Pin	None
Parallel I/O Handshake	STAI

The software interrupt (SWI instruction) is a non-maskable instruction rather than a maskable interrupt source. The illegal opcode interrupt is a non-maskable interrupt. The last interrupt source, external input to the $\overline{\text{XIRQ}}$ pin, is considered 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 $\overline{\text{XIRQ}}$ pin. **Table 9-2**, **Table 9-3**, and **Table 9-4** provide a list of each interrupt, its vector location in memory, and the actual condition code and control bits that mask it. A discussion of the various interrupts is provided below. **Figure 9-3** shows the interrupt stacking order.

Table 9-3 Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
•	•		
•	•		
FFD4, D5	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	See Table 9-3
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Output Compare 5	I Bit	OC5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real Time Interrupt	I Bit	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (External Pin or Parallel I/O)	I Bit	See Table 9-4
FFF4, F5	$\overline{\text{XIRQ}}$ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

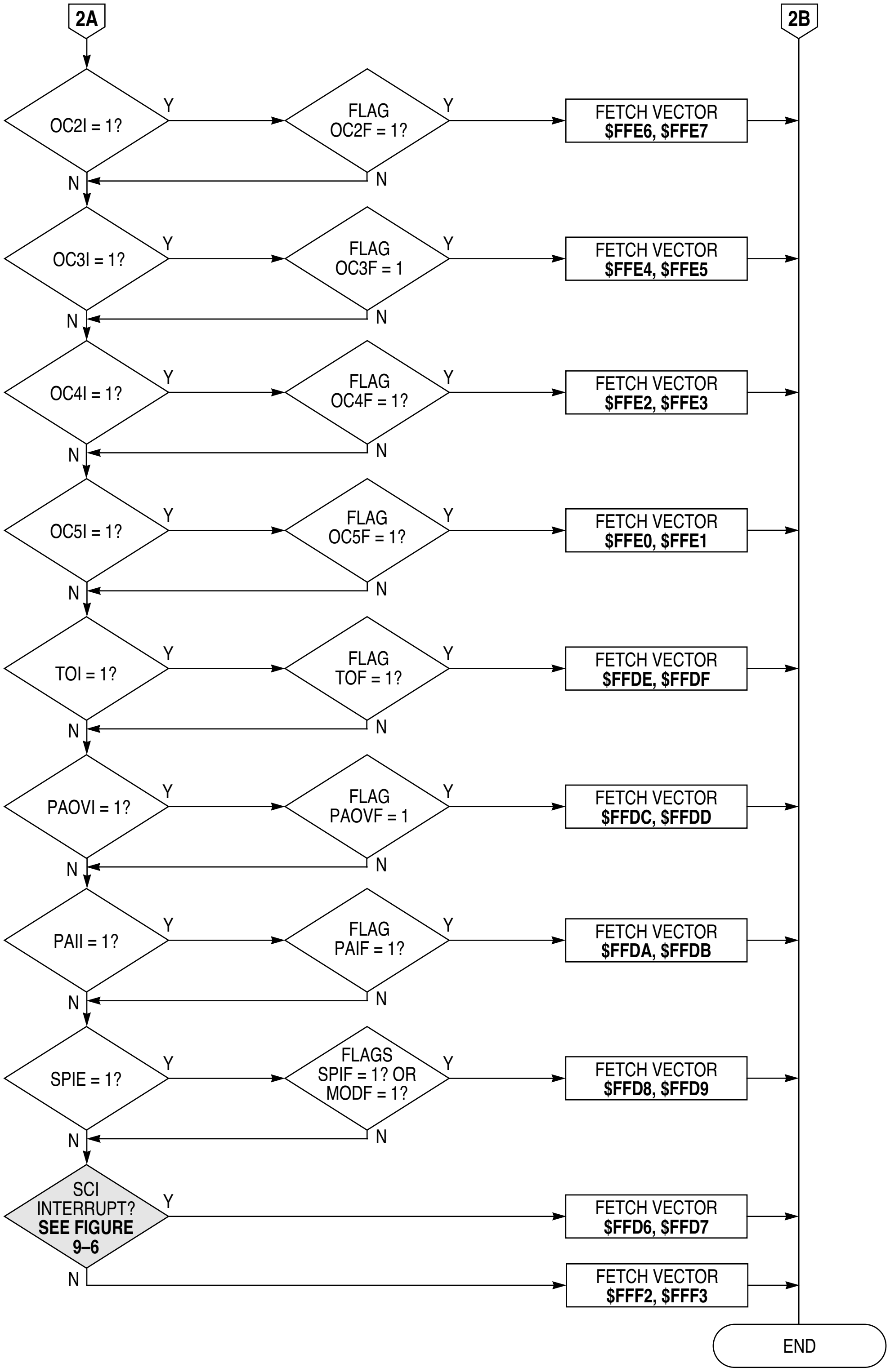

2A
2B
OC2I = 1?
FLAG OC2F = 1?
FETCH VECTOR $FFE6, $FFE7
OC3I = 1?
FLAG OC3F = 1
FETCH VECTOR $FFE4, $FFE5
OC4I = 1?
FLAG OC4F = 1?
FETCH VECTOR $FFE2, $FFE3
OC5I = 1?
FLAG OC5F = 1?
FETCH VECTOR $FFE0, $FFE1
TOI = 1?
FLAG TOF = 1?
FETCH VECTOR $FFDE, $FFDF
PAOVI = 1?
FLAG PAOVF = 1
FETCH VECTOR $FFDC, $FFDD
PAII = 1?
FLAG PAIF = 1?
FETCH VECTOR $FFDA, $FFDB
SPIE = 1?
FLAGS SPIF = 1? OR MODF = 1?
FETCH VECTOR $FFD8, $FFD9
SCI INTERRUPT? SEE FIGURE 9–6
FETCH VECTOR $FFD6, $FFD7
FETCH VECTOR $FFF2, $FFF3
Y
N
END

Figure 9-5 Interrupt Priority Resolution (Sheet 2 of 2)

10.2.7 Prebyte

In order to expand the number of instructions used in the MC68HC11A8, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. The instruction opcodes 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.

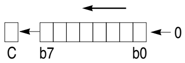
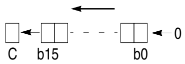
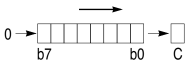
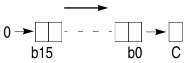
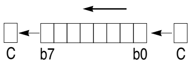
10.3 Instruction Set

The central processing unit (CPU) in the MC68HC11A8 is basically a proper extension of the MC6801 CPU. In addition to its ability to execute all M6800 and M6801 instructions, the MC68HC11A8 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, STOP and WAIT instructions, and bit manipulation instructions.

Table 10-1 shows all MC68HC11A8 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 10-1** which explain the letters in the Operand and Execution Time columns for some instructions. Definitions of “Special Ops” found in the Boolean Expression column are found in **Figure 10-2**.

Table 10-2 through **Table 10-8** provide a detailed description of the information present on the address bus, data bus, and the read/write (R/$\overline{\text{W}}$) line during each cycle of each instruction. The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same address mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Table 10-1 MC68HC11A8 Instructions, Addressing Modes, and Execution Times (Sheet 4 of 6)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycle	Cycle by Cycle*	Condition Codes
				Opcode	Operand(s)				S X H I N Z V C
LDS (opr)	Load Stack Pointer	M:M + 1 → SP	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 2 3 2 3	3 4 5 5 6	3-2 4-3 5-4 6-6 7-6	- - - - ↕ ↕ 0 -
LDX (opr)	Load Index Register X	M:M + 1 → IX	IMM DIR EXT IND,X IND,Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 2 3 2 3	3 4 5 5 6	3-2 4-3 5-4 6-6 7-6	- - - - ↕ ↕ 0 -
LDY (opr)	Load Index Register Y	M:M + 1 → IY	IMM DIR EXT IND,X IND,Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 3 4 3 3	4 5 6 6 6	3-4 4-5 5-6 6-7 7-6	- - - - ↕ ↕ 0 -
LSL (opr) LSLA LSLB	Logical Shift Left		EXT IND,X IND,Y A INH B INH	78 68 18 68 48 58	hh ll ff ff	3 2 3 1 1	6 6 7 2 2	5-8 6-3 3-7 2-1 2-1	- - - - ↕ ↕ ↕ ↕
LSLD	Logical Shift Left Double		INH	05		1	3	2-2	- - - - ↕ ↕ ↕ ↕
LSR (opr) LSRA LSRB	Logical Shift Right		EXT IND,X IND,Y A INH B INH	74 64 18 64 44 54	hh ll ff ff	3 2 3 1 1	6 6 7 2 2	5-8 6-3 7-3 2-1 2-1	- - - - ↕ ↕ ↕ ↕
LSRD	Logical Shift Right Double		INH	04		1	3	2-2	- - - - 0 ↕ ↕ ↕
MUL	Multiply 8 by 8	AxB → D	INH	3D		1	10	2-13	- - - - - - - ↕
NEG (opr)	2's Complement Memory Byte	0 – M → M	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	3 2 3	6 6 7	5-8 6-3 7-3	- - - - ↕ ↕ ↕ ↕
NEGA	2's Complement A	0 – A → A	A INH	40		1	2	2-1	- - - - ↕ ↕ ↕ ↕
NEGB	2's Complement B	0 – B → B	B INH	50		1	2	2-1	- - - - ↕ ↕ ↕ ↕
NOP	No Operation	No Operation	INH	01		1	2	2-1	- - - - - - - -
ORAA (opr)	OR Accumulator A (Inclusive)	A + M → A	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A BA AA 18 AA	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	- - - - ↕ ↕ 0 -
ORAB (opr)	OR Accumulator B (Inclusive)	B + M → B	B IMM B DIR B EXT B IND,X B IND,Y	CA DA FA EA 18 EA	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	3-1 4-1 5-2 6-2 7-2	- - - - ↕ ↕ 0 -
PSHA	Push A onto Stack	A → Stk, SP = SP–1	A INH	36		1	3	2-6	- - - - - - - -
PSHB	Push B onto Stack	B → Stk, SP = SP–1	B INH	37		1	3	2-6	- - - - - - - -
PSHX	Push X onto Stack (Lo First)	IX → Stk, SP = SP–2	INH	3C		1	4	2-7	- - - - - - - -
PSHY	Push Y onto Stack (Lo First)	IY → Stk, SP = SP–2	INH	18 3C		2	5	2-8	- - - - - - - -
PULA	Pull A from Stack	SP = SP + 1, A←Stk	A INH	32		1	4	2-9	- - - - - - - -
PULB	Pull B from Stack	SP = SP + 1, B←Stk	B INH	33		1	4	2-9	- - - - - - - -
PULX	Pull X from Stack (Hi First)	SP = SP + 2, IX←Stk	INH	38		1	5	2-10	- - - - - - - -
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY←Stk	INH	18 38		2	6	2-11	- - - - - - - -
ROL (opr) ROLA ROLB	Rotate Left		EXT IND,X IND,Y A INH B INH	79 69 18 69 49 59	hh ll ff ff	3 2 3 1 1	6 6 7 2 2	5-8 6-3 7-3 2-1 2-1	- - - - ↕ ↕ ↕ ↕

*Cycle-by-cycle number provides a reference to Tables 10-2 through 10-8 which detail cycle-by-cycle operation. Example: Table 10-1 Cycle-by-Cycle column reference number 2-4 equals Table 10-2 line item 2-4.

Table 10-2 Cycle-by-Cycle Operation — Inherent Mode (Sheet 2 of 4)

Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
2-11	PULY	6	1	Opcode Address	1	Opcode (Page Select Byte) ($18)
			2	Opcode Address + 1	1	Opcode (Second Byte) ($38)
			3	Opcode Address + 2	1	Irrelevant Data
			4	Stack Pointer	1	Irrelevant Data
			5	Stack Pointer + 1	1	IYH (High Byte) from Stack
			6	Stack Pointer + 2	1	IYH (Low Byte) from Stack
2-12	RTS	5	1	Opcode Address	1	Opcode ($39)
			2	Opcode Address + 1	1	Irrelevant Data
			3	Stack Pointer	1	Irrelevant Data
			4	Stack Pointer +1	1	Address of Next Instruction (High Byte)
			5	Stack Pointer + 2	1	Address of Next Instruction (Low Byte)
2-13	MUL	10	1	Opcode Address	1	Opcode ($3D)
			2	Opcode Address + 1	1	Irrelevant Data
			3	$FFFF	1	Irrelevant Data
			4	$FFFF	1	Irrelevant Data
			5	$FFFF	1	Irrelevant Data
			6	$FFFF	1	Irrelevant Data
			7	$FFFF	1	Irrelevant Data
			8	$FFFF	1	Irrelevant Data
			9	$FFFF	1	Irrelevant Data
			10	$FFFF	1	Irrelevant Data
2-14	RTI	12	1	Opcode Address	1	Opcode ($3B)
			2	Opcode Address + 1	1	Irrelevant Data
			3	Stack Pointer	1	Irrelevant Data
			4	Stack Pointer + 1	1	Condition Code Register from Stack
			5	Stack Pointer + 2	1	B Accumulator from Stack
			6	Stack Pointer + 3	1	A Accumulator from Stack
			7	Stack Pointer + 4	1	IXH (High Byte) from Stack
			8	Stack Pointer + 5	1	IXL (Low Byte) from Stack
			9	Stack Pointer + 6	1	IYH (High Byte) from Stack
			10	Stack Pointer + 7	1	IYL (Low Byte) from Stack
			11	Stack Pointer + 8	1	Address of Next Instruction (High Byte)
			12	Stack Pointer + 9	1	Address of Next Instruction (Low Byte)
2-15	SWI	14	1	Opcode Address	1	Opcode ($3F)
			2	Opcode Address + 1	1	Irrelevant Data
			3	Stack Pointer	0	Return Address (Low Byte)
			4	Stack Pointer – 1	0	Return Address (High Byte)
			5	Stack Pointer – 2	0	IYL (Low Byte) to Stack
			6	Stack Pointer – 3	0	IYH (High Byte) to Stack
			7	Stack Pointer – 4	0	IXL (Low Byte) to Stack
			8	Stack Pointer – 5	0	IXH (High Byte) to Stack
			9	Stack Pointer – 6	0	A Accumulator to Stack
			10	Stack Pointer – 7	0	B Accumulator to Stack
			11	Stack Pointer – 8	0	Condition Code Register to Stack
			12	Stack Pointer – 8	1	Irrelevant Data
			13	Address of SWI Vector (First Location)	1	SWI Service Routine Address (High Byte)
			14	Address of Vector + 1 (Second Location)	1	SWI Service Routine Address (Low Byte)

* The reference number is given to provide a cross-reference to Table 10-1.

Table 10-5 Cycle-by-Cycle Operation — Extended Mode (Sheet 2 of 2)

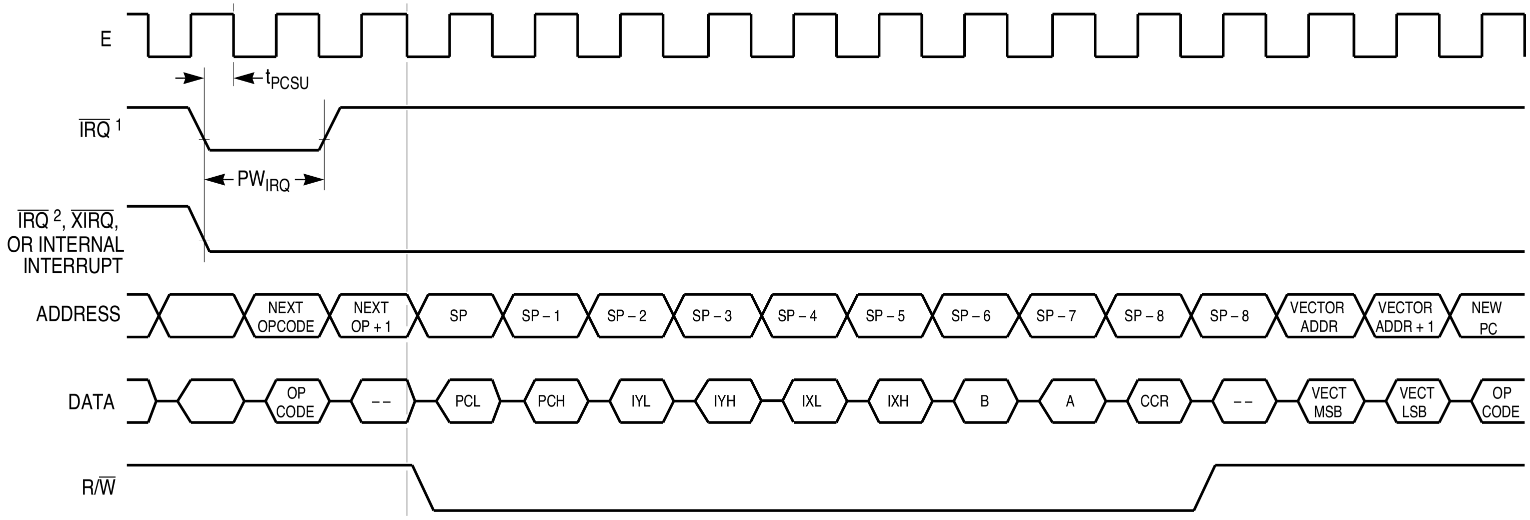
Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{\text{W}}$ Line	Data Bus
5-11	CPD, CPY	7	1	Opcode Address	1	Opcode (Page Select Byte)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Operand Address (High Byte)
			4	Opcode Address + 3	1	Operand Address (Low Byte)
			5	Operand Address	1	Operand Data (High Byte)
			6	Operand Address + 1	1	Operand Data (Low Byte)
			7	$FFFF	1	Irrelevant Data
5-12	JSR	6	1	Opcode Address	1	Opcode ($BD)
			2	Opcode Address + 1	1	Subroutine Address (High Byte)
			3	Opcode Address + 2	1	Subroutine Address (Low Byte)
			4	Subroutine Address	1	First Opcode in Subroutine
			5	Stack Pointer	0	Return Address (Low Byte)
			6	Stack Pointer – 1	0	Return Address (High Byte)

*The reference number is given to provide a cross-reference to Table 10-1.

Table 10-6 Cycle-by-Cycle Operation — Indexed X Mode (Sheet 1 of 2)

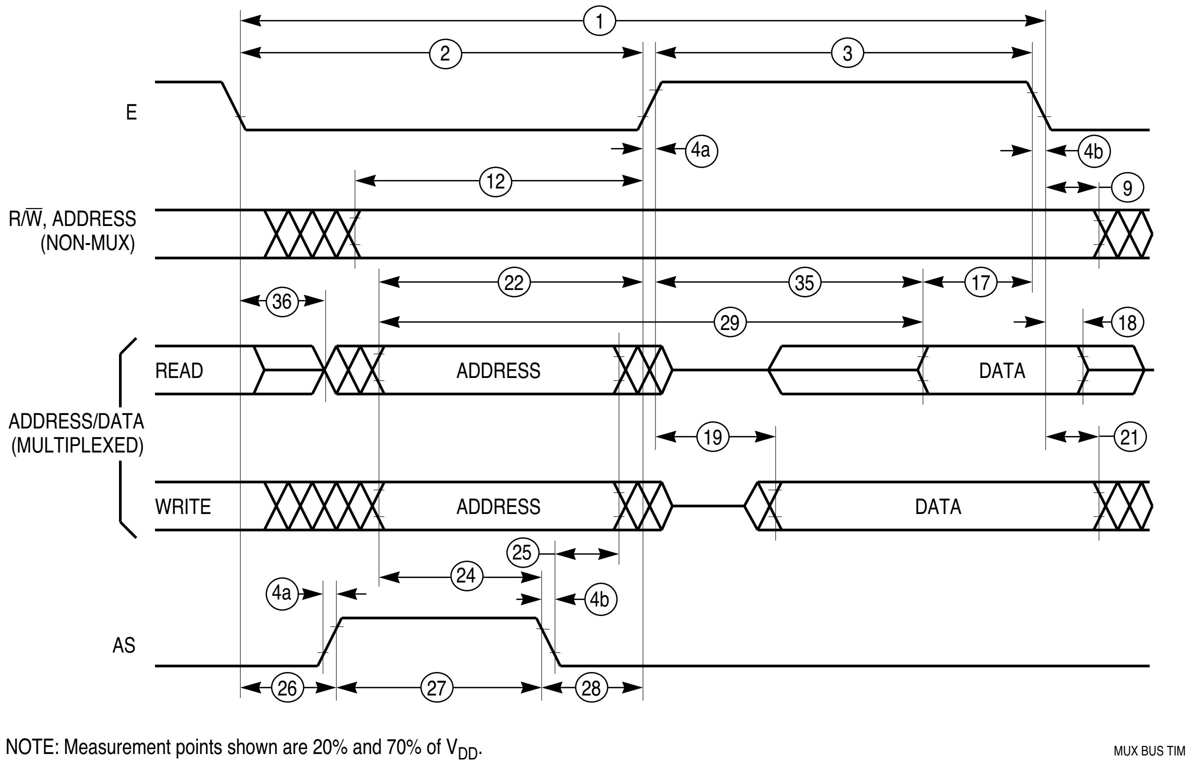
Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{\text{W}}$ Line	Data Bus
6-1	JMP	3	1	Opcode Address	1	Opcode ($6E)
			2	Opcode Address + 1	1	Index Offset
			3	$FFFF	1	Irrelevant Data
6-2	ADCA, ADCB, ADDA, ADDB, ANDA, ANDB, BITA, BITB, CMPA, CMPB, EORA, EORB, LDAA, LDAB, ORAA, ORAB, SBCA, SBCB, SUBA, SUBB	4	1	Opcode Address	1	Opcode
			2	Opcode Address + 1	1	Index Offset
			3	$FFFF	1	Irrelevant Data
			4	(IX) + Offset	1	Operand Data
6-3	ASL, ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, ROR	6	1	Opcode Address	1	Opcode
			2	Opcode Address + 1	1	Index Offset
			3	$FFFF	1	Irrelevant Data
			4	(IX) + Offset	1	Original Operand Data
			5	$FFFF	1	Irrelevant Data
			6	(IX) + Offset	0	Result Operand Data
6-4	TST	6	1	Opcode Address	1	Opcode ($6D)
			2	Opcode Address + 1	1	Index Offset
			3	$FFFF	1	Irrelevant Data
			4	(IX) + Offset	1	Original Operand Data
			5	$FFFF	1	Irrelevant Data
			6	$FFFF	1	Irrelevant Data
6-5	STAA, STAB	4	1	Opcode Address	1	Opcode
			2	Opcode Address + 1	1	Index Offset
			3	$FFFF	1	Irrelevant Data
			4	(IX) + Offset	0	Accumulator Data
6-6	LDD, LDS, LDX	5	1	Opcode Address	1	Opcode
			2	Opcode Address + 1	1	Index Offset
			3	$FFFF	1	Irrelevant Data
			4	(IX) + Offset	1	Operand Data (High Byte)
			5	(IX) + Offset + 1	1	Operand Data (Low Byte)

*The reference number is given to provide a cross-reference to Table 10-1.



NOTES:
1. Edge sensitive $\overline{IRQ}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{IRQ}$ pin (IRQE bit = 0)

Figure A-6 Interrupt Timing Diagram



NOTE: Measurement points shown are 20% and 70% of V_{DD}.

Figure A-14 Multiplexed Expansion Bus Timing Diagram

Table B-1 Ordering Information

Package	Temperature	CONFIG	Description	MC Order Number
52-Pin PLCC	– 40° to + 85°C	$0C	No ROM, No EEPROM	MC68HC11A0CFN2
	– 40° to + 85°C	$0C	No ROM, No EEPROM, 3 MHz	MC68HC11A0CFN3
	– 40° to + 85°C	$0D	No ROM	MC68HC11A1CFN2
	– 40° to + 85°C	$0D	No ROM, 3 MHz	MC68HC11A1CFN3
	– 40° to + 105°C	$0D	No ROM	MC68HC11A1VFN2
	– 40° to + 125°C	$0D	No ROM	MC68HC11A1MFN2
	– 40° to + 85°C	$09	No ROM, COP On	MC68HCP11A1CFN2
	– 40° to + 85°C	$09	No ROM, COP On, 3 MHz	MC68HCP11A1CFN3
	– 40° to + 105°C	$09	No ROM, COP On	MC68HCP11A1VFN2
	– 40° to + 125°C	$09	No ROM, COP On	MC68HCP11A1MFN2
	– 40° to + 85°C	$0F	BUFFALO ROM	MC68HC11A8BCFN2
	– 40° to + 105°C	$0F	BUFFALO ROM	MC68HC11A8VCFN2
	– 40° to + 125°C	$0F	BUFFALO ROM	MC68HC11A8MCFN2
48-Pin DIP	– 40° to + 85°C	$0C	No ROM, No EEPROM	MC68HC11A0CP2
	– 40° to + 85°C	$0C	No ROM, No EEPROM, 3 MHz	MC68HC11A0CP3
	– 40° to + 85°C	$0D	No ROM	MC68HC11A1CP2
	– 40° to + 85°C	$0D	No ROM, 3 MHz	MC68HC11A1CP3
	– 40° to + 105°C	$0D	No ROM	MC68HC11A1VP2
	– 40° to + 125°C	$0D	No ROM	MC68HC11A1MP2
	– 40° to + 85°C	$09	No ROM, COP On	MC68HCP11A1CP2
	– 40° to + 85°C	$09	No ROM, COP On, 3 MHz	MC68HCP11A1CP3
	– 40° to + 105°C	$09	No ROM, COP On	MC68HCP11A1VP2
	– 40° to + 125°C	$09	No ROM, COP On	MC68HCP11A1MP2
	– 40° to + 85°C	$0F	BUFFALO ROM	MC68HC11A8BCP2
	– 40° to + 105°C	$0F	BUFFALO ROM	MC68HC11A8BVP2
	– 40° to + 125°C	$0F	BUFFALO ROM	MC68HC11A8BMP2
64-Pin QFP	– 40° to + 85°C	$0C	No ROM, No EEPROM	MC68HC11A0CFU2
	– 40° to + 85°C	$0C	No ROM, No EEPROM, 3 MHz	MC68HC11A0CFU3
	– 40° to + 85°C	$0D	No ROM	MC68HC11A1CFU2
	– 40° to + 85°C	$0D	No ROM, 3 MHz	MC68HC11A1CFU3
	– 40° to + 105°C	$0D	No ROM	MC68HC11 A1VFU2
	– 40° to + 125°C	$0D	No ROM	MC68HC11A1MFU2
	– 40° to + 85°C	$0F	BUFFALO ROM	MC68HC11A8BCFU2
	– 40° to + 105°C	$0F	BUFFALO ROM	MC68HC11A8VCFU2