# NXP USA Inc. - MC711P2CFNE3 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 | MI Bus, SCI, SPI |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 50 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | OTP |
| EEPROM Size | 640 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | A/D 8x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 84-LCC (J-Lead) |
| Supplier Device Package | 84-PLCC (29.29x29.29) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mc711p2cfne3 |

# Section 1. General Description

## 1.1 Contents

## 1.2 Introduction

The MC68HC11P2 8-bit microcomputer is a member of the M68HC11 family of HCMOS microcomputers. In addition to 32 kbytes of ROM, the MC68HC11P2 contains 1 kbyte of RAM and 640 bytes of EEPROM. With its advanced timer and communication features (including MI BUS[1]) the MC68HC11P2 is especially suitable for mobile communications and automotive applications.

The MC68HC711P2 is an EPROM version of the MC68HC11P2, with the User ROM replaced by a similar amount of EPROM. All references to the MC68HC11P2 apply equally to the MC68HC711P2, unless otherwise noted. *References specific to the MC68HC711P2 are italicised in the text.*

[1] The Motorola interconnect bus (MI BUS) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity. It allows data

outputs. Writes to PORTA do not change the pin state when the pins are configured for timer output compares.

Out of reset, port A pins [7:0] are general-purpose high-impedance inputs. When the functions associated with these pins are disabled, the bits in DDRA govern the I/O state of the associated pin. For further information, refer to **Parallel Input/Output**.

### 2.13.2 Port B

Port B is an 8-bit general-purpose I/O port with a data register (PORTB) and a data direction register (DDRB). In single chip mode, port B pins are general-purpose I/O pins (PB[7:0]). In expanded mode, port B pins act as the high-order address lines (A[15:8]) of the address bus.

PORTB can be read at any time: inputs return the pin level; outputs return the pin driver input level. If PORTB is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. For further information, refer to **Parallel Input/Output**.

Port B pins include on-chip pull-up devices which can be enabled or disabled.

### 2.13.3 Port C

Port C is an 8-bit general-purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In single chip mode, port C pins are general-purpose I/O pins (PC[7:0]). In the expanded mode, port C pins are configured as data bus pins (D[7:0]).

PORTC can be read at any time: inputs return the pin level; outputs return the pin driver input level. If PORTC is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. Port C pins are general-purpose inputs out of reset in single chip and bootstrap modes. In expanded and test modes, these pins are data bus lines out of reset.

PORTH can be read at any time: inputs return the pin level; outputs return the pin driver input level. If PORTH is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode.

Port H pins include on-chip pull-up or pull-down devices that can be enabled or disabled via the Port pull-up assignment register (PPAR). Port H [7:4] have pull-up resistors; port H [3:0] have pull-down resistors.

For further information, refer to **Parallel Input/Output**, **Serial Communications Interface (SCI)**, **Motorola Interconnect Bus (MI BUS)** and **Timing System**.

### 3.3.1 Single chip operating mode

In single chip operating mode, the MC68HC11P2 microcontroller has no external address or data bus. Ports B, C, F, and the $R/\overline{W}$ pin are available for general-purpose parallel I/O.

### 3.3.2 Expanded operating mode

In expanded operating mode, the MCU can access a 64kbyte physical address space. The address space includes the same on-chip memory addresses used for single chip mode, in addition to external memory and peripheral devices.

The expansion bus is made up of ports B, C, and F, and the $R/\overline{W}$ signal. In expanded mode, high order address bits are output on the port B pins, low order address bits on the port F pins, and the data bus on port C. The $R/\overline{W}$/PG7 pin signals the direction of data transfer on the port C bus.

### 3.3.3 Special test mode

Special test, a variation of the expanded mode, is primarily used during Motorola's internal production testing; however, it is accessible for programming the CONFIG register, programming calibration data into EEPROM, and supporting emulation and debugging during development.

After enabling the A/D power, at least 100 µs should be allowed for system stabilization.

CSEL — Clock select (refer to **Analog-to-Digital Converter**)

1 = A/D and EEPROM use internal RC clock source (about 1.5 MHz).

0 = A/D and EEPROM use system E clock (must be at least 1 MHz).

Selects alternate clock source for on-chip EEPROM and A/D charge pumps. The on-chip RC clock should be used when the E clock frequency falls below 1 MHz.

IRQE — Configure $\overline{\text{IRQ}}$ for falling edge sensitive operation

1 = Falling edge sensitive operation.

0 = Low level sensitive operation.

DLY — Enable oscillator start-up delay

1 = A delay of approximately 4064 E clock cycles is imposed as the MCU is started up from the STOP mode.

0 = The oscillator start-up delay coming out of STOP is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

CME — Clock monitor enable (refer to **Resets and Interrupts**)

1 = Clock monitor enabled.

0 = Clock monitor disabled.

In order to use both STOP and clock monitor, the CME bit should be set before executing STOP, then set again after recovering from STOP.

FCME — Force clock monitor enable (refer to **Resets and Interrupts**)

1 = Clock monitor enabled; cannot be disabled until next reset.

0 = Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits (refer to **Resets and Interrupts**)

M — Mode (select character format)
1 = Start bit, 9 data bits, 1 stop bit.
0 = Start bit, 8 data bits, 1 stop bit.

WAKE — Wakeup by address mark/idle
1 = Wakeup by address mark (most significant data bit set).
0 = Wakeup by IDLE line recognition.

ILT — Idle line type
1 = Long (SCI counts ones only after stop bit).
0 = Short (SCI counts consecutive ones after start bit).

This bit determines which of two types of idle line detection method is used by the SCI receiver. In short mode the stop bit and any bits that were ones before the stop bit will be considered as part of that string of ones, possibly resulting in erroneous or premature detection of an idle line condition. In long mode the SCI system does not begin counting ones until a stop bit is received.

PE — Parity enable
1 = Parity enabled.
0 = Parity disabled.

PT — Parity type
1 = Parity odd (an odd number of ones causes parity bit to be zero, an even number of ones causes parity bit to be one).
0 = Parity even (an even number of ones causes parity bit to be zero, an odd number of ones causes parity bit to be one).

## 5.9 Status flags and interrupts

The SCI transmitter has two status flags. These status flags can be read by software (polled) to tell when certain conditions exist. Alternatively, a local interrupt enable bit can be set to enable each of these status conditions to generate interrupt requests. Status flags are automatically set by hardware logic conditions, but must be cleared by software. This provides an interlock mechanism that enables logic to know when software has noticed the status indication. The software clearing sequence for these flags is automatic — functions that are normally performed in response to the status flags also satisfy the conditions of the clearing sequence.

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to one). The TDRE flag indicates there is room in the transmit queue to store another data character in the transmit data register. The TIE bit is the local interrupt mask for TDRE. When TIE is zero, TDRE must be polled. When TIE and TDRE are one, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is zero, TC must be polled; when TCIE is one and TC is one, an interrupt is requested.

Writing a zero to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is cleared when the transmitter is already idle, the pin reverts to its general-purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is cleared, that character is completed before the pin reverts to general-purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

# Section 6. Motorola Interconnect Bus (MI BUS)

## 6.1 Contents

## 6.2 Introduction

The Motorola Interconnect Bus (MI BUS) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity, at a typical bit rate for the data transfer of 20kHz. The MI BUS is suitable for medium speed networks requiring very low cost multiplex wiring; only one wire is required to connect to slave devices.[1]

[1] Related information on Motorola’s MI BUS is contained in the following Motorola publications:

optionally be used to indicate a multiple master bus contention. Refer to **Figure 7-2**.

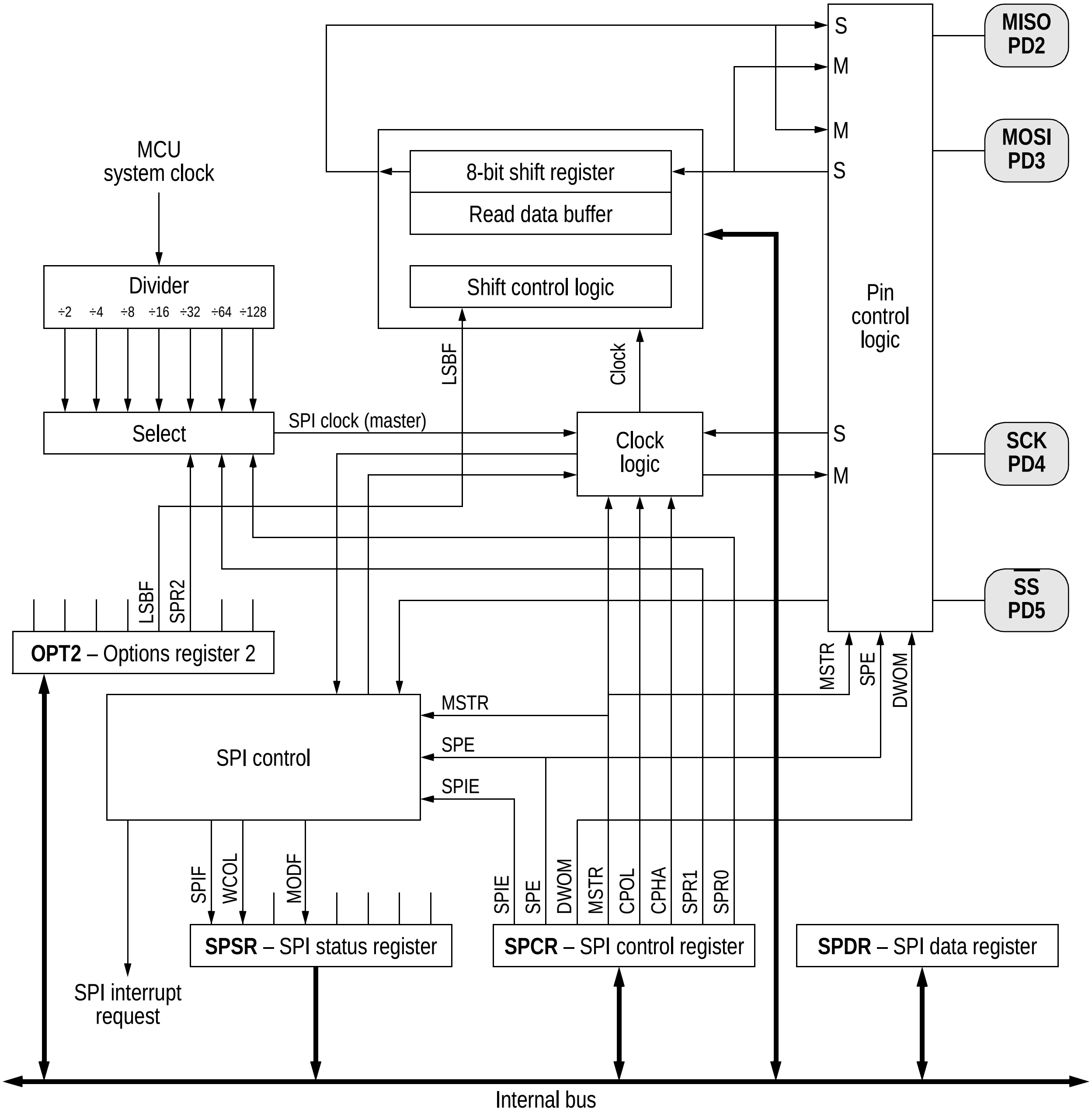
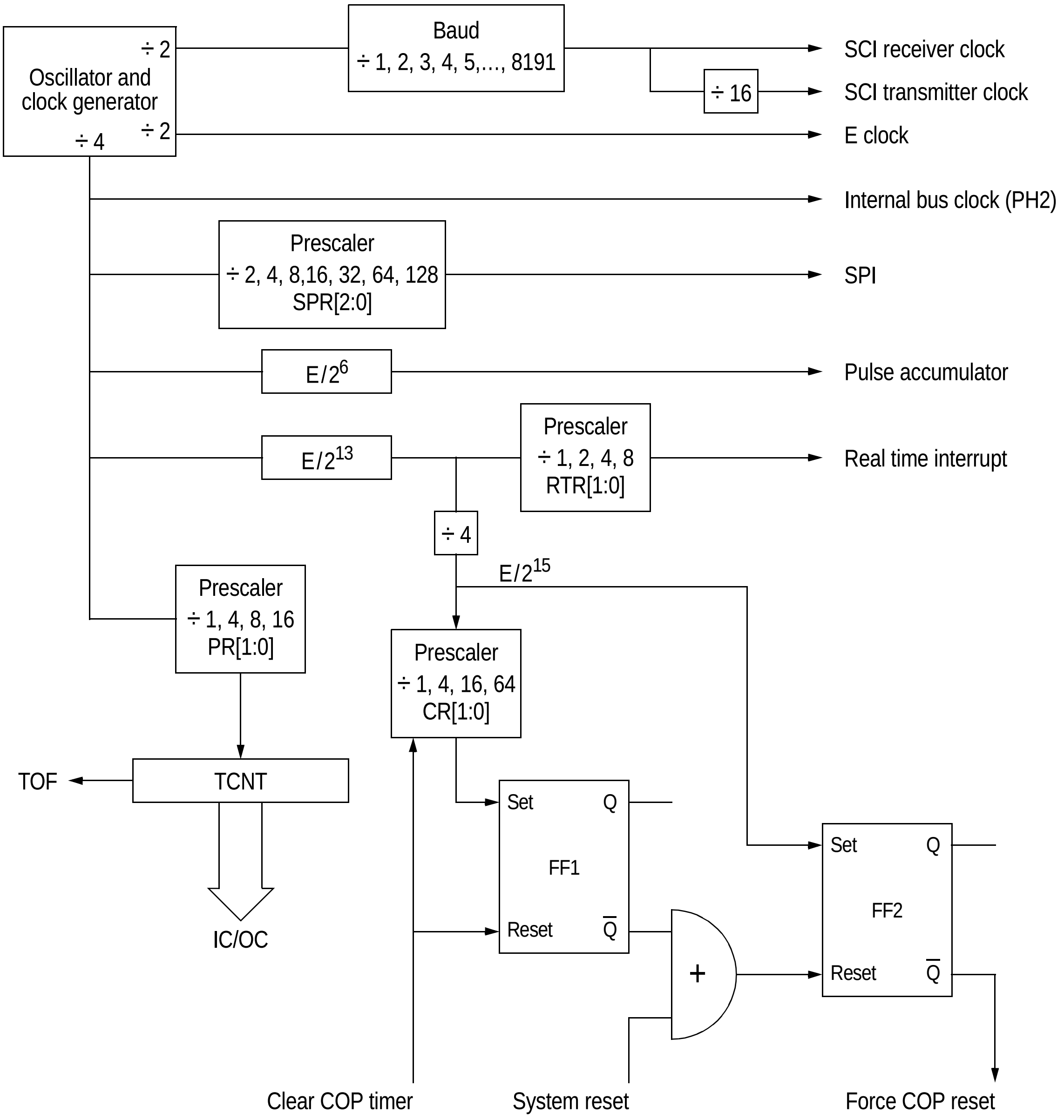


**Figure 7-1. SPI block diagram**

used for an output compare function, it cannot be written directly as if it were a general-purpose output. Each of the output compare functions (OC[5:2]) is related to one of the port A output pins. Output compare 1 (OC1) has extra control logic, allowing it optional control of any combination of the PA[7:3] pins. The PA7 pin can be used as a general-purpose I/O pin, as an input to the pulse accumulator or as an OC1 output pin.



### 8.8.1 PACTL — Pulse accumulator control register

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|---|---|---|---|---|---|---|---|---|---|---|
| Pulse accumulator control (PACTL) | $0026 | 0 | PAEN | PAMOD | PEDGE | 0 | I4/O5 | RTR1 | RTR0 | 0000 0000 |

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

**Bits [7, 3]** — Not implemented; always read zero

PAEN — Pulse accumulator system enable
1 = Pulse accumulator enabled.
0 = Pulse accumulator disabled.

PAMOD — Pulse accumulator mode
1 = Gated time accumulation mode.
0 = Event counter mode.

PEDGE — Pulse accumulator edge control

This bit has different meanings depending on the state of the PAMOD bit, as shown:

| PAMOD | PEDGE | Action of 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. |

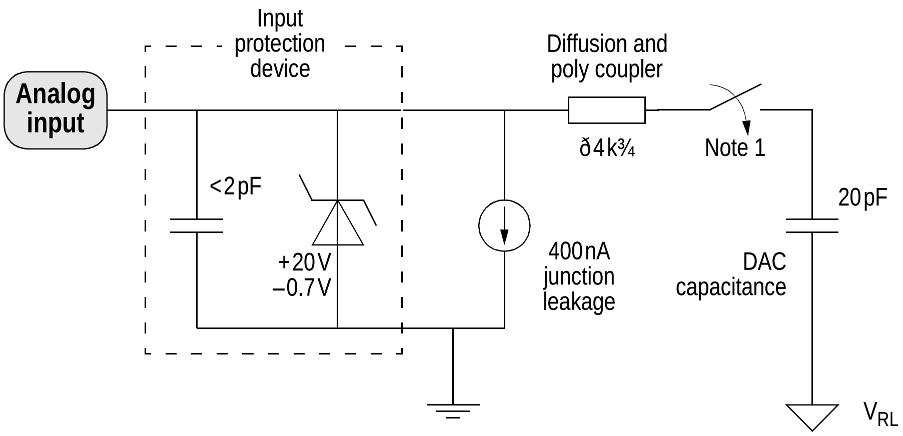
I4/O5 — Input capture 4/output compare 5
1 = Input capture 4 function is enabled (no OC5).
0 = Output compare 5 function is enabled (no IC4).

RTR[1:0] — RTI interrupt rate selects (refer to **Real-time interrupt**)

### 9.3.1 Multiplexer

The multiplexer selects one of 16 inputs for conversion. Input selection is controlled by the value of bits CD – CA in the ADCTL register. The eight port E pins are fixed-direction analog inputs to the multiplexer, and additional internal analog signal lines are routed to it.

Port E pins can also be used as digital inputs. Digital reads of port E pins are not recommended during the sample portion of an A/D conversion cycle, when the gate signal to the N-channel input gate is on. Because no P-channel devices are directly connected to either input pins or reference voltage pins, voltages above $V_{DD}$ do not cause a latchup problem, although current should be limited according to maximum ratings. Refer to **Figure 9-2**, which is a functional diagram of an input pin.



Note 1: The analog switch is closed only during the 12 cycle sample time
Note 2: All component values are approximate

**Figure 9-2. Electrical model of an A/D input pin (in sample mode)**

## 9.5 Channel assignments

The multiplexer allows the A/D converter to select one of sixteen analog signals. Eight of these channels correspond to port E input lines to the MCU, four others are internal reference points or test functions; the remaining four channels are reserved. Refer to **Table 9-1**.

**Table 9-1. A/D converter channel assignments**

| Channel number | Channel signal | Result in ADRx if MULT = 1 |
|---|---|---|
| 1 | AN0 | ADR1 |
| 2 | AN1 | ADR2 |
| 3 | AN2 | ADR3 |
| 4 | AN3 | ADR4 |
| 5 | AN4 | ADR1 |
| 6 | AN5 | ADR2 |
| 7 | AN6 | ADR3 |
| 8 | AN7 | ADR4 |
| 9–12 | reserved | — |
| 13 | $V_{RH}$(1) | ADR1 |
| 14 | $V_{RL}$* | ADR2 |
| 15 | $V_{RH}/2$* | ADR3 |
| 16 | reserved* | ADR4 |

1. Used for factory testing.

### 9.5.1 Single-channel operation

There are two types of single-channel operation. In the first type (SCAN = 0), the single selected channel is converted four consecutive times. The first result is stored in A/D result register 1 (ADR1), and the fourth result is stored in ADR4. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL register. In the second type of single-channel operation (SCAN = 1), conversions continue to be performed on the selected channel with the fifth conversion being stored in register ADR1 (overwriting the first conversion result), the sixth conversion overwriting

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence. Unused bits always read as ones.

If SMOD = 1, CONFIG bits can be written at any time. If SMOD = 0, CONFIG bits can only be written using the EEPROM programming sequence, and are neither readable nor active until latched via the next reset.

ROMAD — ROM mapping control (refer to **Operating Modes and On-Chip Memory**)
1 = ROM addressed from $8000 to $FFFF.
0 = ROM addressed from $0000 to $7FFF (expanded mode only).

Bits [6,5] — Not implemented; always read one

PAREN — Pull-up assignment register enable (refer to **Parallel Input/Output**)
1 = PPAR register enabled; pull-ups can be enabled using PPAR.
0 = PPAR register disabled; all pull-ups disabled.

NOSEC — EEPROM security disabled (refer to **Operating Modes and On-Chip Memory**)
1 = Disable security.
0 = Enable security.

NOCOP — COP system disable
1 = COP system disabled.
0 = COP system enabled (forces reset on timeout).

ROMON — ROM enable (refer to **Operating Modes and On-Chip Memory**)
1 = ROM included in the memory map.
0 = ROM excluded from the memory map.

EEON — EEPROM enable (refer to **Operating Modes and On-Chip Memory**)
1 = EEPROM included in the memory map.

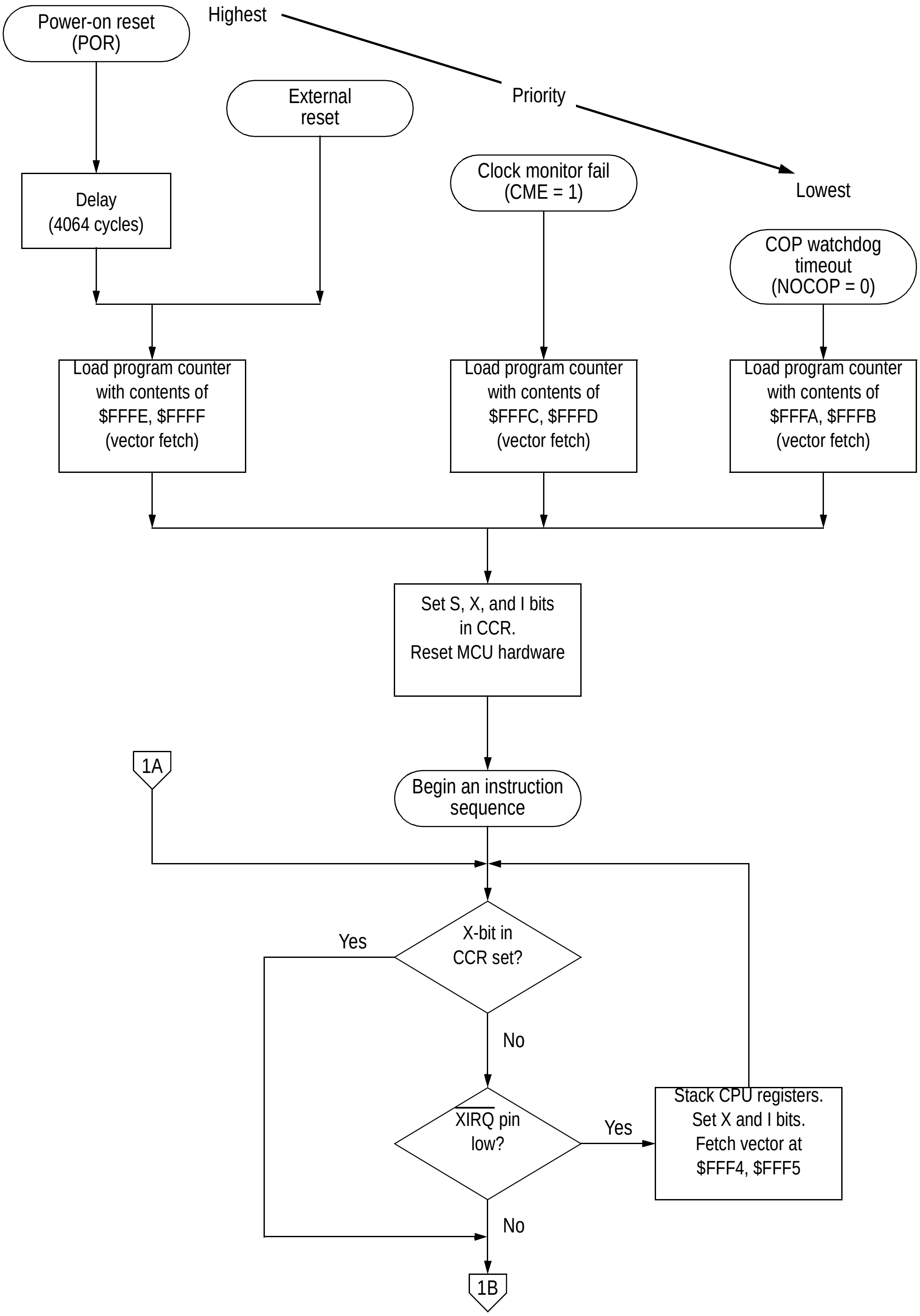
## 10.6 Interrupts

Excluding reset type interrupts, the MC68HC11P2 has 20 interrupt vectors that support 32 interrupt sources. The 17 maskable interrupts are generated by on-chip peripheral systems. These interrupts are recognized when the global interrupt mask bit (I) in the condition code register (CCR) is clear. The three nonmaskable interrupt sources are illegal opcode trap, software interrupt, and $\overline{\text{XIRQ}}$ pin. Refer to **Table 10-4**, which shows the interrupt sources and vector assignments for each source.

For some interrupt sources, such as the SCI interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions.

### 10.6.1 Interrupt recognition and register stacking

An interrupt can be recognized at any time after it is enabled by its local mask, if any, and by the global mask bit in the CCR. Once an interrupt source is recognized, the CPU responds at the completion of the instruction being executed. Interrupt latency varies according to the number of cycles required to complete the current instruction. When the CPU begins to service an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in **Table 10-5**. After the CCR value is stacked, the I-bit and the X-bit, if $\overline{\text{XIRQ}}$ is pending, are set to inhibit further interrupts. The interrupt vector for the highest priority pending source is fetched, and execution continues at the address specified by the vector. At the end of the interrupt service routine, the return from interrupt instruction is executed and the saved registers are


Power-on reset (POR)
Highest
Priority
Lowest
External reset
Delay (4064 cycles)
Clock monitor fail (CME = 1)
COP watchdog timeout (NOCOP = 0)
Load program counter with contents of $FFFE, $FFFF (vector fetch)
Load program counter with contents of $FFFC, $FFFD (vector fetch)
Load program counter with contents of $FFFA, $FFFB (vector fetch)
Set S, X, and I bits in CCR. Reset MCU hardware
1A
Begin an instruction sequence
Yes
X-bit in CCR set?
No
$\overline{\text{XIRQ}}$ pin low?
Yes
Stack CPU registers. Set X and I bits. Fetch vector at $FFF4, $FFF5
No
1B

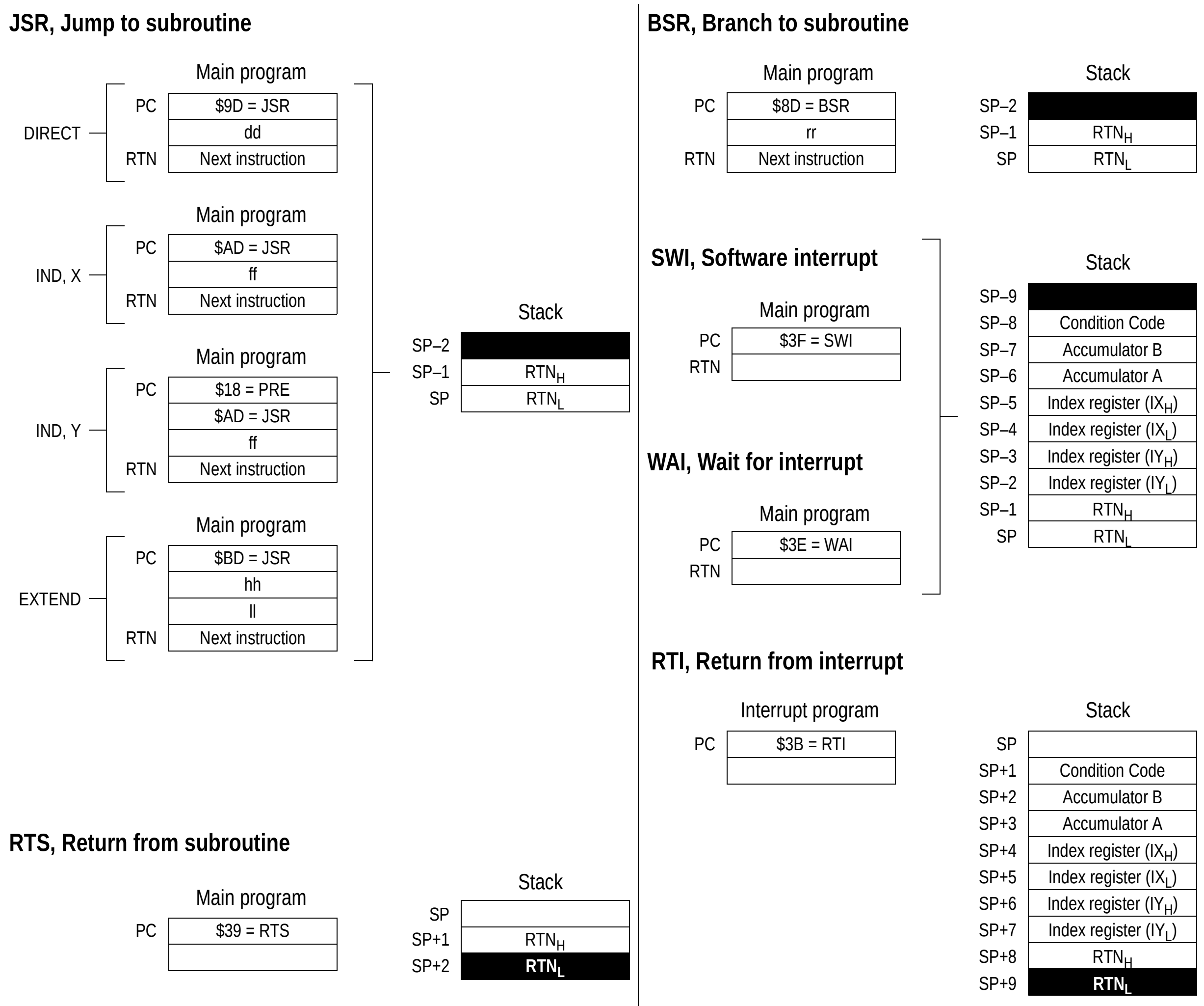
# Section 11. CPU Core and Instruction Set

## 11.1 Contents

## 11.2 Introduction

This section discusses the M68HC11 central processing unit (CPU) architecture, its addressing modes and the instruction set. For more detailed information on the instruction set, refer to the ***M68HC11 Reference Manual (M68HC11RM/AD)***.

The CPU is designed to treat all peripheral, I/O and memory locations identically, as addresses in the 64kbyte memory map. This is referred to as memory-mapped I/O. There are no special instructions for I/O that are separate from those used for memory. This architecture also allows accessing an operand from an external memory location with no execution-time penalty.



**Legend**

| | |
|---|---|
| RTN | Address of the next instruction in the main program, to be executed on return from subroutine |
| $RTN_H$ | More significant byte of return address |
| $RTN_L$ | Less significant byte of return address |
| ■ | Shaded cells show stack pointer position after the operation is complete |
| dd | 8-bit direct address ($0000–$00FF); the high byte is assumed to be $00 |
| ff | 8-bit positive offset ($00 to $FF (0 to 256)) is added to the index register contents |
| hh | High order byte of 16-bit extended address |
| ll | Low order byte of 16-bit extended address |
| rr | Signed relative offset ($80 to $7F (–128 to +127)); offset is relative to the address following the offset byte |

### 12.7.4 Nonmultiplexed expansion bus timing

($V_{DD}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = $T_L$ to $T_H$)

| Num | Characteristic [1] | Symbol | 2.0 MHz | | 3.0 MHz | | 4.0 MHz | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Max. | Min. | Max. | Min. | Max. | |
| | Frequency of operation (E clock frequency) | $f_{OP}$ | 0 | 2.0 | 0 | 3.0 | 0 | 4.0 | MHz |
| 1 | E clock period | $t_{CYC}$ | 500 | — | 333 | — | 250 | — | ns |
| 2 | Pulse width, E low [2], [3] | $PW_{EL}$ | 230 | — | 147 | — | 105 | — | ns |
| 3 | Pulse width, E high [2], [3] | $PW_{EH}$ | 225 | — | 142 | — | 100 | — | ns |
| 4A<br>4B | E clockrise time<br>fall time | $t_r$<br>$t_f$ | —<br>— | 20<br>20 | —<br>— | 20<br>18 | —<br>— | 20<br>15 | ns |
| 9 | Address hold time [3] | $t_{AH}$ | 53 | — | 32 | — | 21 | — | ns |
| 11 | Address delay time [3] | $t_{AD}$ | — | 103 | — | 82 | — | 71 | ns |
| 12 | Address valid to E rise time [3] | $t_{AV}$ | 127 | — | 65 | — | 34 | — | ns |
| 17 | Read data set-up time | $t_{DSR}$ | 30 | — | 30 | — | 20 | — | ns |
| 18 | Read data hold time | $t_{DHR}$ | 0 | — | 0 | — | 0 | — | ns |
| 19 | Write data delay time | $t_{DDW}$ | — | 40 | — | 40 | — | 40 | ns |
| 21 | Write data hold time [3] | $t_{DHW}$ | 63 | — | 42 | — | 31 | — | ns |
| 29 | MPU address access time [3] | $t_{ACCA}$ | 347 | — | 203 | — | 144 | — | ns |
| 39 | Write data set-up time [3] | $t_{DSW}$ | 185 | — | 102 | — | 60 | — | ns |
| 57 | Address valid to data tristate time | $t_{AVDZ}$ | — | 10 | — | 10 | — | 10 | ns |

[1] All timing is given with respect to 20% and 70% of $V_{DD}$, unless otherwise noted.
[2] Input clock duty cycles other than 50% will affect the bus performance.
[3] For $f_{OP}$ 2MHz the following formulae may be used to calculate parameter values:
$PW_{EL} = t_{CYC}/2 - 20$ns $PW_{EH} = t_{CYC}/2 - 25$ns
$t_{AH} = t_{CYC}/8 - 10$ns $t_{AD} = t_{CYC}/8 + 40$ns
$t_{AV} = PW_{EL} - t_{AD}$ $t_{DHW} = t_{CYC}/8$
$t_{ACCA} = t_{CYC} - t_f - t_{DSR} - t_{AD} t_{DSW} = PW_{EH} - t_{DDW}$
$t_{ECSA} = PW_{EH} - t_{ECSD} - t_{DSR} t_{ACSD} = t_{CYC}/4 + 40$ns
$t_{ACSA} = t_{CYC} - t_f - t_{DSR} - t_{ACSD}$

# Revision History