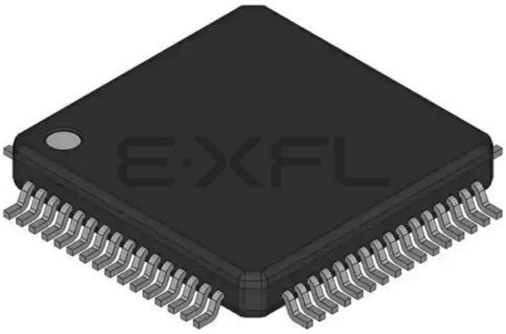


Welcome to E-XFL.COM

## 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 | HC08 |
| Core Size | 8-Bit |
| Speed | 6MHz |
| Connectivity | I²C, USB |
| Peripherals | OSD, POR, PWM |
| Number of I/O | 39 |
| Program Memory Size | 60KB (60K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 6x8b |
| Oscillator Type | Internal |
| Operating Temperature | 0°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-QFP |
| Supplier Device Package | 64-QFP (14x14) |
| Purchase URL | https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908ld64ifue |

## Section 13. Analog-to-Digital Converter (ADC)



## Section 14. Universal Serial Bus Module (USB)

## Section 19. Input/Output (I/O) Ports



## Section 20. External Interrupt (IRQ)

| Figure | Title | Page |
|---|---|---|

| Table | Title | Page |
|---|---|---|

## 4.4 FLASH Control Registers

The two FLASH control registers control FLASH program and erase operations.

This register controls the 47,616-byte array:

Address: $FE07

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | 0 | 0 | 0 | 0 | HVEN | MASS | ERASE | PGM |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(shaded) = Unimplemented

**Figure 4-2. 47,616-byte FLASH Control Register (FLCR)**

This register controls the 13K-byte array:

Address: $FE0A

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | 0 | 0 | 0 | 0 | HVEN1 | MASS1 | ERASE1 | PGM1 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(shaded) = Unimplemented

**Figure 4-3. 13K-byte FLASH Control Register (FLCR1)**

FLCR1 is used with the OSD FLASH even high byte write buffer (OSDEHBUF) in programming operations. See **4.4.1 OSD FLASH Even High Byte Write Buffer (OSDEHBUF)**.

The following are bit definitions for FLCR and FLCR1.

HVEN — High-Voltage Enable Bit

This read/write bit enables the charge pump to drive high voltages for program and erase operations in the array. HVEN can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

1 = High voltage enabled to array and charge pump on
0 = High voltage disabled to array and charge pump off

## 4.7 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 64 consecutive bytes starting from addresses $XX00, $XX40, $XX80, and $XXC0. Use this step-by-step procedure to program a row of FLASH memory (**Figure 4-5** is a flowchart representation):

***NOTE:*** *In order to avoid program disturbs, the row must be erased before any byte on that row is programmed.*

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any FLASH address within the row address range desired.
3. Wait for a time, $t_{nvs}$ (min. 5μs).
4. Set the HVEN bit.
5. Wait for a time, $t_{pgs}$ (min. 10μs).
6. For 47,616-byte array: Write data to the FLASH address to be programmed.
   For 13K-byte array: Write even address data to OSDEHBUF then write odd address data to the odd FLASH address to be programmed.
7. Wait for time, $t_{PROG}$ (min. 20μs).
8. Repeat step 6 and 7 until all the bytes within the row are programmed.
9. Clear the PGM bit.
10. Wait for time, $t_{nvh}$ (min. 5μs).
11. Clear the HVEN bit.
12. After time, $t_{rcv}$ (min 1μs), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

***NOTE:*** *Programming and erasing of FLASH locations cannot be performed by code being executed from the same FLASH array that is being programmed or erased. While these operations must be performed in the order shown, other unrelated operations may occur between the steps. Do not exceed $t_{PROG}$ maximum. See* ***24.14 FLASH Memory Characteristics****.*

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt may take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared). (See **Figure 9-10. Interrupt Processing**.)

### 9.6.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register), and if the corresponding interrupt enable bit is set, the SIM proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. **Figure 9-11** demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

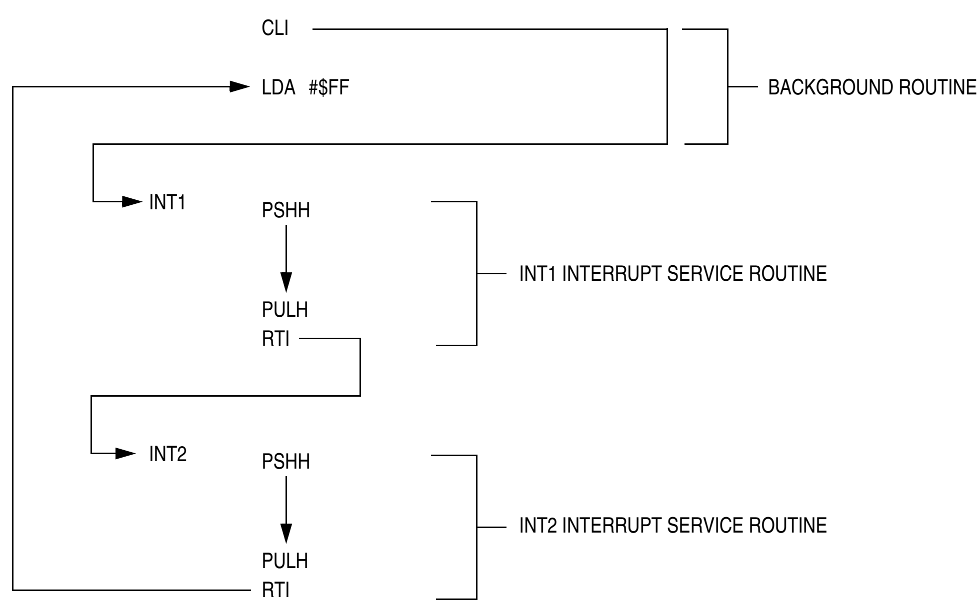



**Figure 9-11. Interrupt Recognition Example**

### 10.4.1 Entering Monitor Mode

Table 10-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

1. If monitor entry is by high voltage on $\overline{IRQ}$ ($\overline{IRQ}$ = $V_{TST}$)
   - The external clock is 4.9152 MHz with PTC3 low or 9.8304 MHz with PTC3 high
2. If monitor entry is by blank reset vector ($FFFE and $FFFF both contain $FF; erased state):
   - The external clock is 9.8304 MHz

**NOTE:** *Holding the PTC3 pin low when entering monitor mode by a high voltage causes a bypass of a divide-by-two stage at the oscillator. The OSCOUT frequency is equal to the OSCXCLK frequency, and the OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50% duty cycle at maximum bus frequency.*

**NOTE:** *If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial POR reset. Once the part has been programmed, the traditional method of applying a high voltage, $V_{TST}$, to $\overline{IRQ}$ must be used to enter monitor mode.*

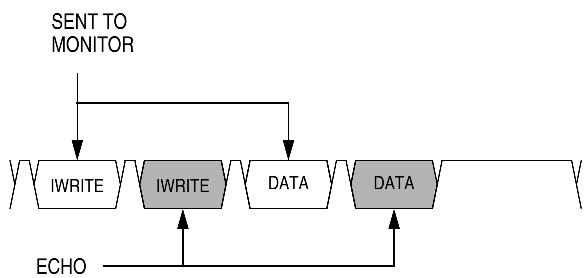
Enter monitor mode with the pin configuration shown in Table 10-1 after a reset. The rising edge of reset latches monitor mode. Once monitor mode is latched, the values on the specified pins can change.

Once out of reset, the MCU monitor mode firmware then sends a break signal (10 consecutive logic zeros) to the host computer, indicating that it is ready to receive a command. The break signal also provides a timing reference to allow the host to determine the necessary baud rate.

**Table 10-6. IWRITE (Indexed Write) Command**

| | |
|---|---|
| **Description** | Write to last address accessed + 1 |
| **Operand** | Specifies single data byte |
| **Data Returned** | None |
| **Opcode** | $19 |

**Command Sequence**

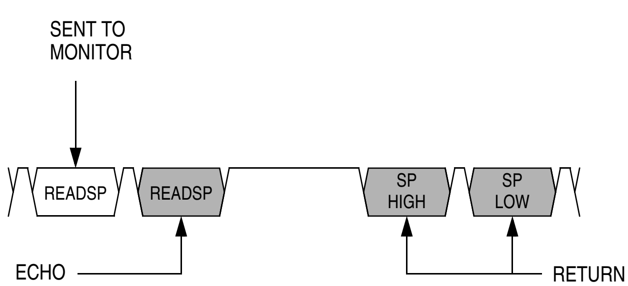



A sequence of IREAD or IWRITE commands can sequentially access a block of memory over the full 64K-byte memory map.

**Table 10-7. READSP (Read Stack Pointer) Command**

| | |
|---|---|
| **Description** | Reads stack pointer |
| **Operand** | None |
| **Data Returned** | Returns stack pointer in high byte:low byte order |
| **Opcode** | $0C |

**Command Sequence**



## 11.5 Functional Description

Figure 11-1 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

The two TIM channels are programmable independently as input capture or output compare channels.

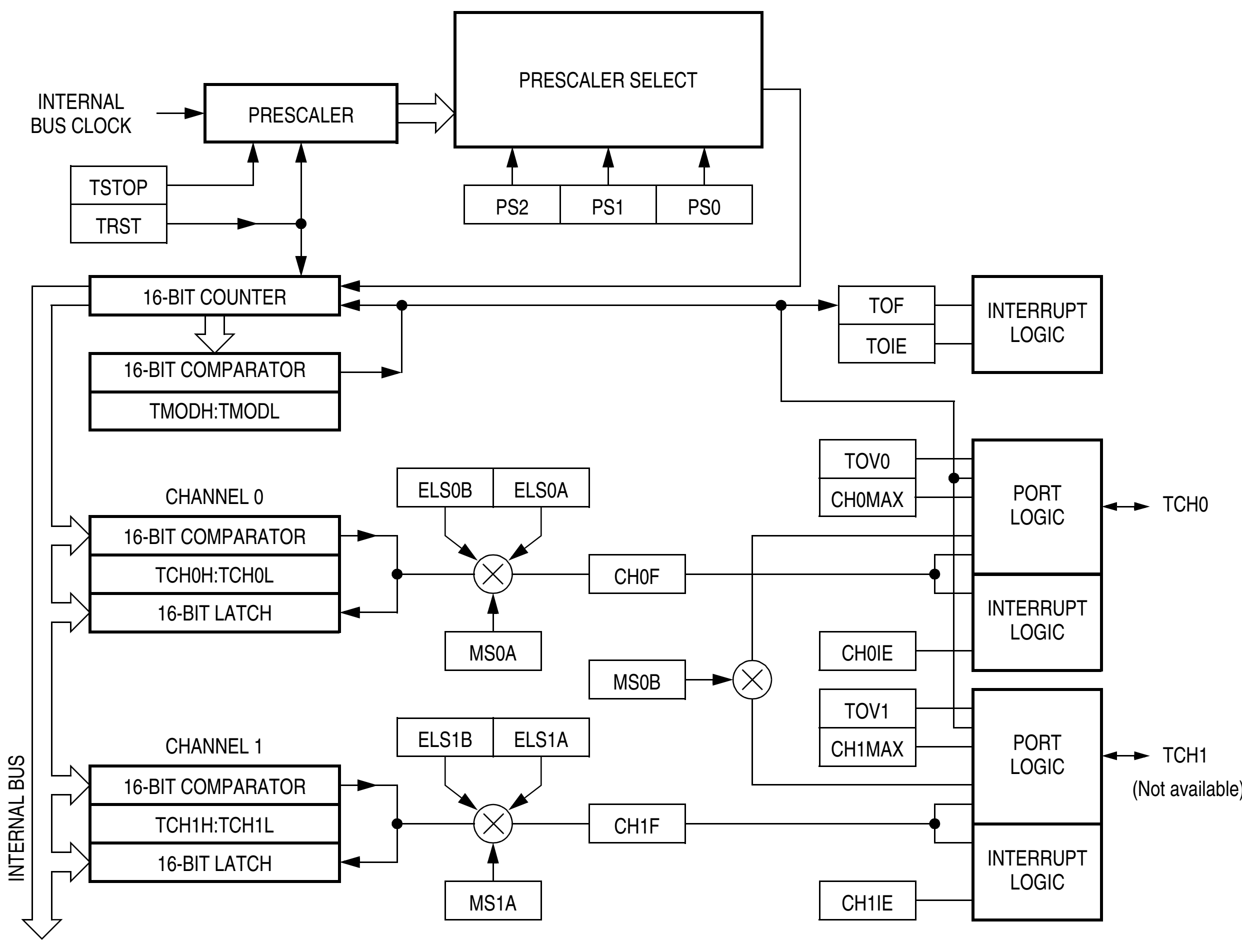



**Figure 11-1. TIM Block Diagram**

**NOTE:** *In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.*

### 11.5.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use the following initialization procedure:

1. In the TIM status and control register (TSC):
   a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
   b. Reset the TIM counter and prescaler by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH:TMODL), write the value for the required PWM period.
3. In the TIM channel x registers (TCHxH:TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
   a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. (See **Table 11-3**.)
   b. Write 1 to the toggle-on-overflow bit, TOVx.
   c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. (See **Table 11-3**.)

**NOTE:** *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output.TIM channel 0 status and control register (TSC0) controls and monitors the PWM signal from the linked channels.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. See **11.10.4 TIM Channel Status and Control Registers (TSC0:TSC1)**.

## 11.6 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow CPU interrupt requests. TOF and TOIE are in the TIM status and control register.
- TIM channel flags (CH1F:CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE=1. CHxF and CHxIE are in the TIM channel x status and control register.

## 11.7 Low-Power Modes

The WAIT and STOP instructions puts the MCU in low-power-consumption standby modes.

### 11.7.1 Wait Mode

The TIM remains active after the execution of a WAIT instruction. In wait mode the TIMA registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

Address: $0011 TCH0H

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| Reset: | Indeterminate after reset | | | | | | | |

Address: $0012 TCH0L

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Reset: | Indeterminate after reset | | | | | | | |

Address: $0014 TCH1H

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| Reset: | Indeterminate after reset | | | | | | | |

Address: $0015 TCH1L

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Reset: | Indeterminate after reset | | | | | | | |

**Figure 11-8. TIM Channel Registers (TCH0H/L:TCH1H/L)**

### 13.4.1 ADC Port I/O Pins

PTC5/ADC5–PTC0/ADC0 are general-purpose I/O pins that are shared with the ADC channels. The channel select bits, ADCH[4:0], in the ADC status and control register define which ADC channel/port pin will be used as the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O. Writes to the port register or DDR will not have any affect on the port pin that is selected by the ADC. Read of a port pin which is in use by the ADC will return a logic 0 if the corresponding DDR bit is at logic 0. If the DDR bit is at logic 1, the value in the port data latch is read.

### 13.4.2 Voltage Conversion

When the input voltage to the ADC equals to VRH, the ADC converts the signal to $FF (full scale). If the input voltage equals to VRL, the ADC converts it to $00. Input voltages between VRH and VRL is a straight-line linear conversion. All other input voltages will result in $FF if greater than VRH and $00 if less than VRL.

***NOTE:*** *Input voltage should not exceed the analog supply voltages.*

### 13.4.3 Conversion Time

Sixteen ADC internal clocks are required to perform one conversion. The ADC starts a conversion on the first rising edge of the ADC internal clock immediately following a write to the ADSCR. If the ADC internal clock is selected to run at 1 MHz, then one conversion will take 16 μs to complete. With a 1 MHz ADC internal clock the maximum sample rate is 62.5 kHz.

$$\text{Conversion time} = \frac{\text{16 to17 ADC cycles}}{\text{ADC frequency}}$$

$$\text{Number of bus cycles} = \text{conversion time} \times \text{bus frequency}$$

## 14.6 Hub Function I/O Registers

The USB hub function provides a set of control/status registers and sixteen data registers that provide storage for the buffering of data between the USB hub function and the CPU.

### 14.6.1 USB Hub Root Port Control Register (HRPCR)

Address: $005E

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | 0 | 0 | 0 | RESUM0 | SUSPND | 0 | D0+ | D0– |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | X | X |

X = Indeterminate ▒ = Unimplemented

**Figure 14-3. USB Hub Root Port Control Register (HRPCR)**

RESUM0 — Force Resume to the Root Port

This read/write bit forces a resume signal (K state) onto the USB root port data lines to initiate a remote wakeup. Software should control the timing of the forced resume to be between 10ms and 15ms. Reset clears this bit.

1 = Force root port data lines to K state
0 = Default

SUSPND — USB Suspend Control Bit

To save power, this read/write bit should be set by the software if a constant idle state for more than 3ms is detected on the USB bus. Setting this bit puts the transceiver into a power savings mode.

This bit also determines the latch scheme for the data lines of the root port and the downstream port. When this bit is 1, the current state shown on the data lines will be reflected to the data register (D+/D–) directly. When the bit is 0, the data registers are the latched state sampled at the last EOF2 sample point. The hub repeater's function is affected by this bit too. The upstream and downstream traffic will be blocked if this bit is set to 1. When the global resume or the downstream remote wakeup signal is found by the suspended hub, In slave mode, the data in MMDRR is:

- the calling address from the master when the address match flag is set (MMATCH = 1); or
- the last data received when MMATCH = 0.

In master mode, the data in the MMDRR is:

- the last data received.

When the MMDRR is read by the CPU, the receive buffer full flag is cleared (MMRXBF = 0), and the next received data is loaded to the MMDRR. Each time when new data is loaded to the MMDRR, the MMRXIF interrupt flag is set, indicating that new data is available in MMDRR.

The sequence of events for slave receive and master receive are illustrated in **Figure 15-8**.

## 15.6 Programming Considerations

When the MMIIC module detects an arbitration loss in master mode, it will release both SDA and SCL lines immediately. But if there are no further STOP conditions detected, the module will hang up. Therefore, it is recommended to have time-out software to recover from such ill condition. The software can start the time-out counter by looking at the MMBB (Bus Busy) flag in the MIMCR and reset the counter on the completion of one byte transmission. If a time-out occur, software can clear the MMEN bit (disable MMIIC module) to release the bus, and hence clearing the MMBB flag. This is the only way to clear the MMBB flag by software if the module hangs up due to a no STOP condition received. The MMIIC can resume operation again by setting the MMEN bit.

## 18.2 Introduction

This section describes the on-screen display (OSD) module. This module includes a 15 row × 30 column display window and video pattern generator.

## 18.3 Features

Features of the on-screen display module include:

- Up to 384 fonts: 12 × 16 or 16 × 16
- Resolution: up to 2048 dots/line
- Scan lines per frame: up to 2048 lines
- Fully programmable display character array of 15 rows by 30 columns
- Eight selections of color for menu windows and fonts
- Row to row spacing control
- Four programmable background windows
- Window shadowing with programmable width, height, and color
- Programmable vertical and horizontal positioning for display center
- Full screen pattern output of free-running VGA, SVGA, XGA, SXGA timing from Sync Processor module
- Double character height and double character width

### 24.12.4 USB Signaling Levels

**Table 24-14. USB Signaling Levels**

| Bus State | Signaling Levels | |
|---|---|---|
| | **From Originating Driver** | **At Receiver** |
| Differential “1” | (D+) – (D–) > 200 mV and D+ or D– > $V_{SE}$ (min.) | |
| Differential “0” | (D+) – (D–) < –200 mV and D+ or D– > $V_{SE}$ (min.) | |
| Data J State:<br>Low Speed<br>Full Speed | <br>Differential “0”<br>Differential “1” | |
| Data K State:<br>Low Speed<br>Full Speed | <br>Differential “1”<br>Differential “0” | |
| Idle State:<br>Low Speed<br>Full Speed | <br>Differential “0” and D– > $V_{SE}$ (max.) and D+ < $V_{SE}$ (min.)<br>Differential “1” and D+ > $V_{SE}$ (max.) and D– < $V_{SE}$ (min.) | |
| Resume State:<br>Low Speed<br>Full Speed | <br>Differential “1” and D+ > $V_{SE}$ (max.) and D– < $V_{SE}$ (min.)<br>Differential “0” and D– > $V_{SE}$ (max.) and D+ < $V_{SE}$ (min.) | |
| Start of Packet (SOP) | Data lines switch from Idle to K State | |
| End of Packet (EOP) | D+ and D– < $V_{SE}$ (min) for 2 bit times[1] followed by an Idle for 1 bit time | D+ and D– < $V_{SE}$(min) for ≥ 1 bit time[2] followed by a J State |
| Disconnect (Upstream only) | — | D+ and D– < $V_{SE}$(max) for ≥ 2.5 µs |
| Connect (Upstream only) | — | D+ or D– > $V_{SE}$(max) for ≥ 2.5 µs |
| Reset (Downstream only) | D+ and D– < $V_{SE}$ for ≥10 ms | D+ and D– < $V_{SE}$ (min) for ≥ 2.5 µs (must be recognized within 5.5 µs)[3] |

**Notes**:
1. The width of EOP is defined in bit times relative to the speed of transmission.
2. The width of EOP is defined in bit times relative to the device type receiving the EOP.
3. These times apply to an active device that is not in the suspend state.

# Section 25. Mechanical Specifications

## 25.1 Contents



## 25.2 Introduction

This section gives the dimensions for:

- 64-pin plastic quad flat pack (case #840B)