**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

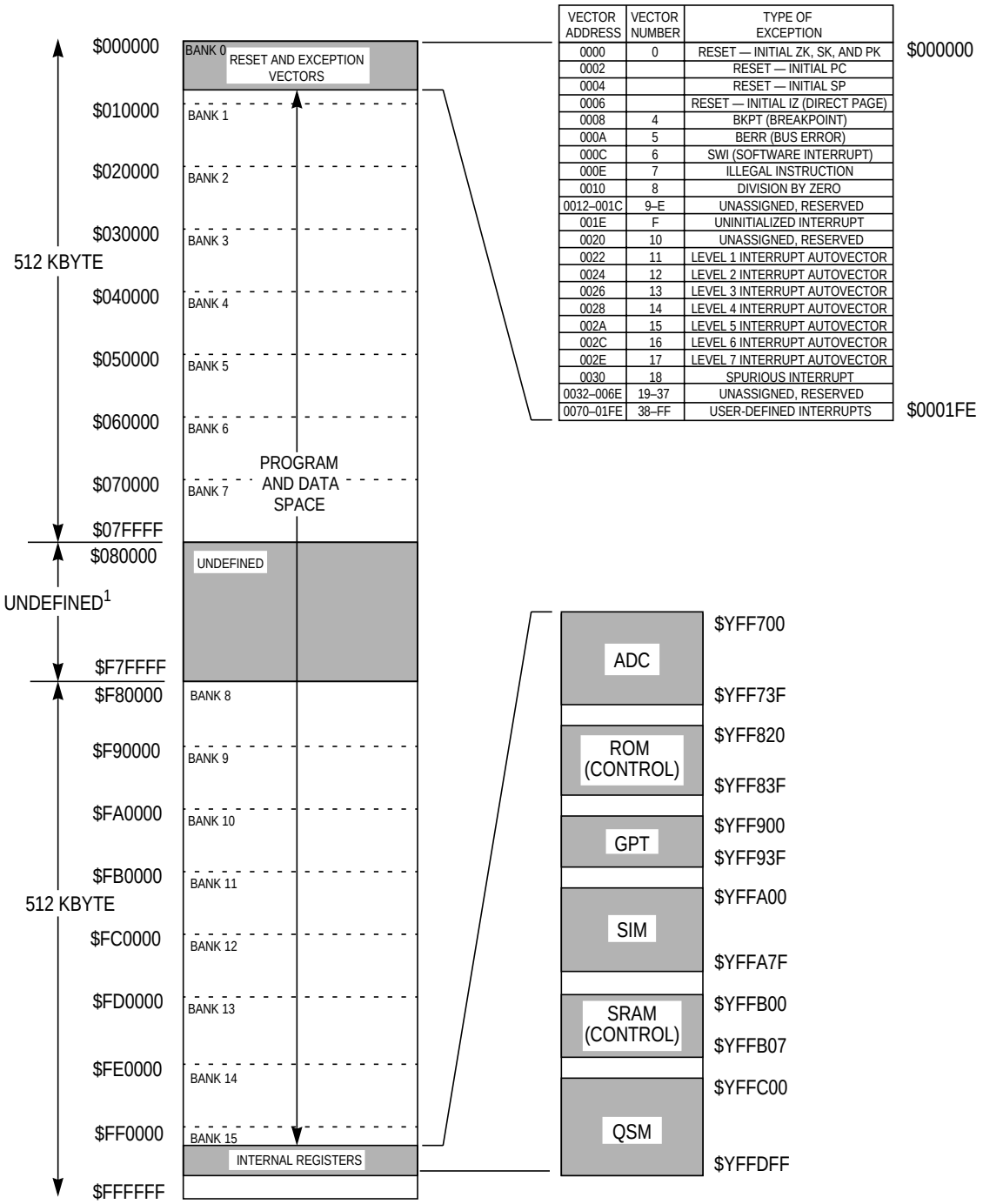
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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 | Not For New Designs |
| Core Processor | CPU16 |
| Core Size | 16-Bit |
| Speed | 20MHz |
| Connectivity | EBI/EMI, SCI, SPI |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 16 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 144-LQFP |
| Supplier Device Package | 144-LQFP (20x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc16z1cag20 |

# TABLE OF CONTENTS
(Continued)

Paragraph Title Page




NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

HC16 Z2/Z3 MEM MAP (C)

**Figure 3-12 MC68HC16Z2/Z3 Combined Program and Data Space Map**

**Table 4-1 Addressing Modes**

| Mode | Mnemonic | Description |
|---|---|---|
| Accumulator Offset | E,X | Index register X with accumulator E offset |
| | E,Y | Index register Y with accumulator E offset |
| | E,Z | Index register Z with accumulator E offset |
| Extended | EXT | Extended |
| | EXT20 | 20-bit extended |
| Immediate | IMM8 | 8-bit immediate |
| | IMM16 | 16-bit immediate |
| Indexed 8-Bit | IND8, X | Index register X with unsigned 8-bit offset |
| | IND8, Y | Index register Y with unsigned 8-bit offset |
| | IND8, Z | Index register Z with unsigned 8-bit offset |
| Indexed 16-Bit | IND16, X | Index register X with signed 16-bit offset |
| | IND16, Y | Index register Y with signed 16-bit offset |
| | IND16, Z | Index register Z with signed 16-bit offset |
| Indexed 20-Bit | IND20, X | Index register X with signed 20-bit offset |
| | IND20, Y | Index register Y with signed 20-bit offset |
| | IND20, Z | Index register Z with signed 20-bit offset |
| Inherent | INH | Inherent |
| Post-Modified Index | IXP | Signed 8-bit offset added to index register X after effective address is used |
| Relative | REL8 | 8-bit relative |
| | REL16 | 16-bit relative |

All modes generate ADDR[15:0]. This address is combined with ADDR[19:16] from an operand or an extension field to form a 20-bit effective address.

**NOTE**

Access across 64-Kbyte address boundaries is transparent. ADDR[19:16] of the effective address are changed to make an access across a bank boundary. Extension field values will not change as a result of effective address computation.

### 4.6.1 Immediate Addressing Modes

In the immediate modes, an argument is contained in a byte or word immediately following the instruction. For IMM8 and IMM16 modes, the effective address is the address of the argument.

There are three specialized forms of IMM8 addressing.

- The AIS, AIX, AIY, AIZ, ADDD, and ADDE instructions decrease execution time by sign-extending the 8-bit immediate operand to 16 bits, then adding it to an appropriate register.
- The MAC and RMAC instructions use an 8-bit immediate operand to specify two signed 4-bit index register offsets.
- The PSHM and PULM instructions use an 8-bit immediate mask operand to indicate which registers must be pushed to or pulled from the stack.

# SECTION 6
# STANDBY RAM MODULE

The standby RAM (SRAM) module consists of a fixed-location control register block and an array of fast (two clock) static RAM that may be mapped to a user specified location in the system memory map. Array size depends on the M68HC16, M68CK16, and M68CM16 Z-series version. Refer to **Table 6-1** for appropriate SRAM array size. The SRAM is especially useful for system stacks and variable storage.

**Table 6-1 SRAM Configuration**

| Z-Series Device | Array Size |
|---|---|
| MC68HC16Z1<br>MC68CK16Z1<br>MC68CM16Z1<br>MC68HC16Z4<br>MC68CK16Z4 | 1 Kbyte |
| MC68HC16Z2 | 2 Kbytes |
| MC68HC16Z3 | 4 Kbytes |

The SRAM can be mapped to any address that is a multiple of the array size so long as SRAM boundaries do not overlap the module control registers (overlap makes the registers inaccessible). Data can be read/written in bytes, words or long words. SRAM is powered by $V_{DD}$ in normal operation. During power-down, SRAM contents can be maintained by power from the $V_{STBY}$ input. Power switching between sources is automatic.

## 6.1 SRAM Register Block

There are four SRAM control registers: the RAM module configuration register (RAMMCR), the RAM test register (RAMTST), and the RAM array base address registers (RAMBAH/RAMBAL).

The module mapping bit (MM) in the SIM configuration register (SIMCR) defines the most significant bit (ADDR23) of the IMB address for each M68HC16, M68CK16, and M68CM16 Z-series module. Because ADDR[23:20] are driven to the same value as ADDR19, MM must be set to one. If MM is cleared, IMB modules are inaccessible. For more information about how the state of MM affects the system, refer to **5.2.1 Module Mapping**.

The SRAM control register consists of eight bytes, but not all locations are implemented. Unimplemented register addresses are read as zeros, and writes have no effect. Refer to **D.3 Standby RAM Module** for the register block address map and register bit/field definitions.

$I_{SB}$ (SRAM standby current) values may vary while $V_{DD}$ transitions occur. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for standby switching and power consumption specifications.

## 6.6 Reset

Reset places the SRAM in low-power stop mode, enables program space access, and clears the base address registers and the register lock bit. These actions make it possible to write a new base address into the ROMBAH and ROMBAL registers.

When a synchronous reset occurs while a byte or word SRAM access is in progress, the access is completed. If reset occurs during the first word access of a long-word operation, only the first word access is completed. If reset occurs during the second word access of a long-word operation, the entire access is completed. Data being read from or written to the RAM may be corrupted by an asynchronous reset. For more information, refer to **5.7 Reset**.

In Figure 8-10, $R_F$ and $C_F$ comprise the user's external filter circuit. $C_S$ is the internal sample capacitor. Each channel has its own capacitor. $C_S$ is never precharged; it retains the value of the last sample. $V_l$ is an internal voltage source used to precharge the DAC capacitor array ($C_{DAC}$) before each sample. The value of this supply is $V_{DDA}$/2, or 2.5 volts for 5-volt operation.

The following paragraphs provide a simplified description of the interaction between the ADC and the user's external circuitry. This circuitry is assumed to be a simple RC low-pass filter passing a signal from a source to the ADC input pin. The following simplifying assumptions are made:

- The source impedance is included with the series resistor of the RC filter.
- The external capacitor is perfect (no leakage, no significant dielectric absorption characteristics, etc.)
- All parasitic capacitance associated with the input pin is included in the value of the external capacitor.
- Inductance is ignored.
- The “on” resistance of the internal switches is zero ohms and the “off” resistance is infinite.

### 8.8.6.1 Settling Time for the External Circuit

The values for $R_F$ and $C_F$ in the user's external circuitry determine the length of time required to charge $C_F$ to the source voltage level ($V_{SRC}$).

At time t = 0, S1 in Figure 8-10 closes. S2 is open, disconnecting the internal circuitry from the external circuitry. Assume that the initial voltage across CF is zero. As CF charges, the voltage across it is determined by the following equation, where t is the total charge time:

$$V_{CF} = V_{SRC}(1 - e^{-t/R_F C_F})$$

When t = 0, the voltage across $C_F$ = 0. As t approaches infinity, $V_{CF}$ will equal $V_{SRC}$. (This assumes no internal leakage.) With 10-bit resolution, 1/2 of a count is equal to 1/2048 full-scale value. Assuming worst case ($V_{SRC}$ = full scale), Table 8-10 shows the required time for $C_F$ to charge to within 1/2 of a count of the actual source voltage during 10-bit conversions. Table 8-10 is based on the RC network in Figure 8-10.

**NOTE**

The following times are completely independent of the A/D converter architecture (assuming the ADC is not affecting the charging).

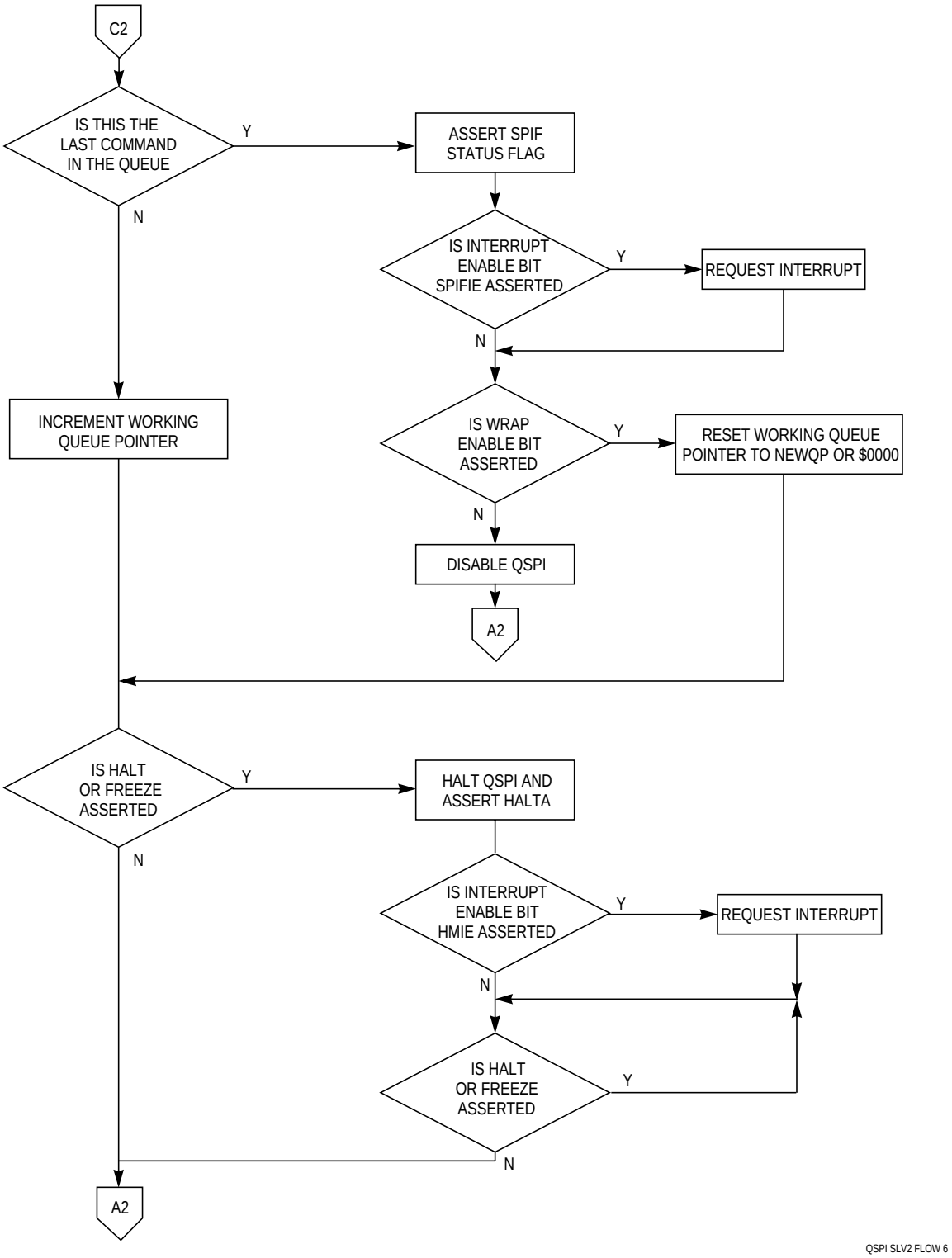



**Figure 9-9 Flowchart of QSPI Slave Operation (Part 2)**

Data transfer is synchronized with the internally-generated serial clock SCK. Control bits, CPHA and CPOL, in SPCR0, control clock phase and polarity. Combinations of CPHA and CPOL determine upon which SCK edge to drive outgoing data from the MOSI pin and to latch incoming data from the MISO pin.

Baud rate is selected by writing a value from two to 255 into SPBR[7:0] in SPCR0. The QSPI uses a modulus counter to derive the SCK baud rate from the MCU system clock.

The following expressions apply to the SCK baud rate:

$$\text{SCK Baud Rate} = \frac{f_{sys}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{f_{sys}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator and SCK assumes its inactive state.

The DSCK bit in each command RAM byte inserts either a standard (DSCK = 0) or user-specified (DSCK = 1) delay from chip-select assertion until the leading edge of the serial clock. The DSCKL field in SPCR1 determines the length of the user-defined delay before the assertion of SCK. The following expression determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = \frac{\text{DSCKL[6:0]}}{f_{sys}}$$

where DSCKL[6:0] equals {1, 2, 3,..., 127}.

When DSCK equals zero, DSCKL[6:0] is not used. Instead, the PCS valid-to-SCK transition is one-half the SCK period.

There are two transfer length options. The user can choose a default value of eight bits, or a programmed value from eight to sixteen bits, inclusive. The programmed value must be written into BITS[3:0] in SPCR0. The BITSE bit in each command RAM byte determines whether the default value (BITSE = 0) or the BITS[3:0] value (BITSE = 1) is used. **Table 9-3** shows BITS[3:0] encoding.

**Table 9-5 Effect of Parity Checking on Data Size**

| M | PE | Result |
|---|---|---|
| 0 | 0 | 8 data bits |
| 0 | 1 | 7 data bits, 1 parity bit |
| 1 | 0 | 9 data bits |
| 1 | 1 | 8 data bits, 1 parity bit |

#### 9.4.3.5 Transmitter Operation

The transmitter consists of a serial shifter and a parallel data register (TDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU16. The transmitter is double-buffered, which means that data can be loaded into TDR while other data is shifted out. The TE bit in SCCR1 enables (TE = 1) and disables (TE = 0) the transmitter.

The shifter output is connected to the TXD pin while the transmitter is operating (TE = 1, or TE = 0 and transmission in progress). Wired-OR operation should be specified when more than one transmitter is used on the same SCI bus. The WOMS bit in SCCR1 determines whether TXD is an open-drain (wired-OR) output or a normal CMOS output. An external pull-up resistor on TXD is necessary for wired-OR operation. WOMS controls TXD function whether the pin is used by the SCI or as a general-purpose I/O pin.

Data to be transmitted is written to SCDR, then transferred to the serial shifter. The transmit data register empty (TDRE) flag in SCSR shows the status of TDR. When TDRE = 0, TDR contains data that has not been transferred to the shifter. Writing to SCDR again overwrites the data. TDRE is set when the data in TDR is transferred to the shifter. Before new data can be written to SCDR, however, the processor must clear TDRE by writing to SCSR. If new data is written to SCDR without first clearing TDRE, the data will not be transmitted.

The transmission complete (TC) flag in SCSR shows transmitter shifter state. When TC = 0, the shifter is busy. TC is set when all shifting operations are completed. TC is not automatically cleared. The processor must clear it by first reading SCSR while TC is set, then writing new data to SCDR.

The state of the serial shifter is checked when the TE bit is set. If TC = 1, an idle frame is transmitted as a preamble to the following data frame. If TC = 0, the current operation continues until the final bit in the frame is sent, then the preamble is transmitted. The TC bit is set at the end of preamble transmission.

The SBK bit in SCCR1 is used to insert break frames in a transmission. A non-zero integer number of break frames is transmitted while SBK is set. Break transmission begins when SBK is set, and ends with the transmission in progress at the time either SBK or TE is cleared. If SBK is set while a transmission is in progress, that transmission finishes normally before the break begins. To assure the minimum break time, toggle SBK quickly to one and back to zero. The TC bit is set at the end of break transmission. After break transmission, at least one bit-time of logic level one (mark idle) is transmitted to ensure that a subsequent start bit can be detected.

Select a value for INTV so that each MCCI interrupt vector corresponds to one of the user-defined vectors ($40–$FF). Refer to the *CPU16 Reference Manual* (CPU16RM/AD) for additional information on interrupt vectors.

### 10.2.2 Pin Control and General-Purpose I/O

The eight pins used by the SPI and SCI subsystems have alternate functions as general-purpose I/O pins. Configuring the MCCI submodule includes programming each pin for either general-purpose I/O or its serial interface function. In either function, each pin must also be programmed as input or output.

The MCCI data direction register (MDDR) assigns each MCCI pin as either input or output. The MCCI pin assignment register (MPAR) assigns the MOSI, MISO, and $\overline{SS}$ pins as either SPI pins or general-purpose I/O. (The fourth pin, SCK, is automatically assigned to the SPI whenever the SPI is enabled, for example, when the SPE bit in the SPI control register is set.) The receiver enable (RE) and transmitter enable (TE) bits in the SCI control registers (SCCR0A, SCCR0B) automatically assign the associated pin as an SCI pin when set or general-purpose I/O when cleared. **Table 10-2** summarizes how pin function and direction are assigned.

**Table 10-2 Pin Assignments**

| Pin | Function Assigned By | Direction Assigned By |
|---|---|---|
| TXDA/PMC7 | TE bit in SCCR0A | MMDR7 |
| RXDA/PMC6 | RE bit in SCCR0A | MMDR6 |
| TXDB/PMC5 | TE bit in SCCR0B | MMDR5 |
| RXDB/PMC4 | RE bit in SCCR0B | MMDR4 |
| $\overline{SS}$/PMC3 | $\overline{SS}$ bit in MPAR | MMDR3 |
| SCK/PMC2 | SPE bit in SPCR | MMDR2 |
| MOSI/PMC1 | MOSI bit in MPAR | MMDR1 |
| MISO/PMC0 | MISO bit in MPAR | MMDR0 |

## 10.3 Serial Peripheral Interface (SPI)

The SPI submodule communicates with external peripherals and other MCUs via a synchronous serial bus. The SPI is fully compatible with the serial peripheral interface systems found on othe Freescale devices, such as the M68HC11 and M68HC05 families. The SPI can perform full duplex three-wire or half duplex two-wire transfers. Serial transfer of eight or sixteen bits can begin with the MSB or LSB. The system can be configured as a master or slave device.

**Figure 10-2** shows a block diagram of the SPI.

### 10.5 MCCI Initialization

After reset, the MCCI remains in an idle state. Several registers must be initialized before serial operations begin. A general sequence guide for initialization follows.

A. Global
   1. Configure MMCR
      a. Write an interrupt arbitration number greater than zero into the IARB field.
      b. Clear the STOP bit if it is not already cleared.
   2. Interrupt vector and interrupt level registers (MIVR, ILSPI, and ILSCI)
      a. Write the SPI/SCI interrupt vector into MIVR.
      b. Write the SPI interrupt request level into the ILSPI and the interrupt request levels for the two SCI interfaces into the ILSCI.
   3. Port data register
      a. Write a data word to PORTMC.
      b. Read a port pin state from PORTMCP.
   4. Pin control registers
      a. Establish the direction of MCCI pins by writing to the MDDR.
      b. Assign pin functions by writing to the MPAR.

B. Serial Peripheral Interface
   1. Configure SPCR
      a. Write a transfer rate value into the BAUD field.
      b. Determine clock phase (CPHA) and clock polarity (CPOL).
      c. Specify an 8- or 16-bit transfer (SIZE) and MSB- or LSB-first transfer mode (LSBF).
      d. Select master or slave operating mode (MSTR).
      e. Enable or disable wired-OR operation (WOMP).
      f. Enable or disable SPI interrupts (SPIE).
      g. Enable the SPI by setting the SPE bit.

C. Serial Communication Interface (SCIA/SCIB)
   1. To transmit, read the SCSR, and then write transmit data to the SCDR. This clears the TDRE and TC indicators in the SCSR.
      a. SCI control register 0 (SCCR0)
      b. Write a baud rate value into the BR field.
   2. Configure SCCR1
      a. Select 8- or 9-bit frame format (M).
      b. Determine use (PE) and type (PT) of parity generation or detection.
      c. To receive, set the RE and RIE bits in SCCR1. Select use (RWU) and type (WAKE) of receiver wakeup. Select idle-line detection type (ILT) and enable or disable idle-line interrupt (ILIE).
      d. To transmit, set TE and TIE bits in SCCR1, and enable or disable WOMC and TCIE bits. Disable break transmission (SBK) for normal operation.

### 11.11.1 PWM Counter

The 16-bit counter in the PWM unit is similar to the timer counter in the capture/compare unit. During reset, the GPT is configured to use the system clock divided by two to drive the counter. Initialization software can reconfigure the counter to use one of seven prescaler outputs or an external clock input from the PCLK pin.

The PWM count register (PWMCNT) can be read at any time without affecting its value. A read must be a word access to ensure coherence, but byte accesses can be made if coherence is not needed. The counter is cleared to $0000 during reset and is a read-only register except in freeze or test mode.

Fifteen of the sixteen counter bits are output to multiplexers A and B. The multiplexers provide the fast and slow modes of the PWM unit. Mode for PWMA is selected by the SFA bit in the PWM control register C (PWMC). Mode for PWMB is selected by the SFB bit in the same register.

PWMA, PWMB, and PPR[2:0] bits in PWMC control PWM output frequency. In fast mode, bits [7:0] of PWMCNT are used to clock the PWM logic; in slow mode, bits [14:7] are used. The period of a PWM output in slow mode is 128 times longer than the fast mode period. **Table 11-3** shows a range of PWM output frequencies using 16.78 MHz, 20.97 MHz, and 25.17 MHz system clocks.

**Table 11-3 PWM Frequency Ranges**

| PPR [2:0] | Prescaler Tap | | | SFA/B = 0 | | | SFA/B = 1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 16.78 MHz | 20.97 MHz | 25.17 MHz | 16.78 MHz | 20.97 MHz | 25.17 MHz | 16.78 MHz | 20.97 MHz | 25.17 MHz |
| 000 | Div 2 = 8.39 MHz | Div 2 = 10.5 MHz | Div 2 = 12.6 MHz | 32.8 kHz | 41 kHz | 49.2 kHz | 256 Hz | 320 Hz | 384 Hz |
| 001 | Div 4 = 4.19 MHz | Div 4 = 5.25 MHz | Div 4 = 6.29 MHz | 16.4 kHz | 20.5 kHz | 24.6 kHz | 128 Hz | 160 Hz | 192 Hz |
| 010 | Div 8 = 2.10 MHz | Div 8 = 2.62 MHz | Div 8 = 3.15 MHz | 8.19 kHz | 10.2 kHz | 12.3 kHz | 64.0 Hz | 80.0 Hz | 96 Hz |
| 011 | Div 16 = 1.05 MHz | Div 16 = 1.31 MHz | Div 16 = 1.57 MHz | 4.09 kHz | 5.15 kHz | 6.13 kHz | 32.0 Hz | 40.0 Hz | 48 Hz |
| 100 | Div 32 = 524 kHz | Div 32 = 655 kHz | Div 32 = 787 kHz | 2.05 kHz | 2.56 kHz | 3.07 kHz | 16.0 Hz | 20.0 Hz | 24 Hz |
| 101 | Div 64 = 262 kHz | Div 64 = 328 kHz | Div 64 = 393 kHz | 1.02 kHz | 1.28 kHz | 1.54 kHz | 8.0 Hz | 10.0 Hz | 12 Hz |
| 110 | Div 128 = 131 kHz | Div 128 = 164 kHz | Div 128 = 197 kHz | 512 Hz | 641 Hz | 770 Hz | 4.0 Hz | 5.0 Hz | 6 Hz |
| 111 | PCLK | PCLK | PCLK | PCLK/256 | PCLK/256 | PCLK/256 | PCLK/ 32768 | PCLK/ 32768 | PCLK/ 32768 |

### 11.11.2 PWM Function

The pulse width values of the PWM outputs are determined by control registers PWMA and PWMB. PWMA and PWMB are 8-bit registers implemented as two bytes of a 16-bit register. PWMA and PWMB can be accessed as separate bytes or as one 16-bit register. A value of $00 loaded into either register causes the corresponding output pin to output a continuous logic level zero signal. A value of $80 causes the corresponding output signal to have a 50% duty cycle, and so on, to the maximum value of $FF, which corresponds to an output which is at logic level one for 255/256 of the cycle.

Setting the F1A (for PWMA) or F1B (for PWMB) bits in the CFORC register causes the corresponding pin to output a continuous logic level one signal. The logic level of the associated pin does not change until the end of the current cycle. F1A and F1B are the lower two bits of CFORC, but can be accessed at the same word address as PWMC.

13. The base configuration of the MC68HC16Z1, MC68CK16Z1, MC68HC16Z4, and the MC68CK16Z4 requires a 32.768 kHz crystal reference. The base configuration of the MC68CM16Z1, MC68HC16Z2, and the MC68HC16Z3 requires a 4.194 MHz crystal reference.
14. The RAM module will not switch into standby mode as long as $V_{SB}$ does not exceed $V_{DD}$ by more than 0.5 volts. The RAM array cannot be accessed while the module is in standby mode.
15. When $V_{SB}$ is more than 0.3 V greater than $V_{DD}$, current flows between the $V_{STBY}$ and $V_{DD}$ pins, which causes standby current to increase toward the maximum transient condition specification. System noise on the $V_{DD}$ and $V_{STBY}$ pin can contribute to this condition.
16. Power dissipation is measured with the appropriate system clock frequency, all modules active. Power dissipation can be calculated using the following expression:

$$P_D = \text{Maximum } V_{DD} (I_{DD} + I_{DDSYN} + I_{SB}) + \text{Maximum } V_{DDA} (I_{DDA})$$

$I_{DD}$ includes supply currents for all device modules powered by $V_{DD}$ pins.

## Table A-16 16.78-MHz AC Timing

($V_{DD}$ and $V_{DDSYN}$ = 5.0 Vdc ± 10 %, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$)[1]

| Num | Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| F1 | Frequency of Operation | f | — | 16.78 | MHz |
| 1 | Clock Period | $t_{cyc}$ | 59.6 | — | ns |
| 1A | ECLK Period | $t_{Ecyc}$ | 476 | — | ns |
| 1B | External Clock Input Period[2] | $t_{Xcyc}$ | 59.6 | — | ns |
| 2, 3 | Clock Pulse Width[3] | $t_{CW}$ | 24 | — | ns |
| 2A, 3A | ECLK Pulse Width | $t_{ECW}$ | 236 | — | ns |
| 2B, 3B | External Clock Input High/Low Time[2] | $t_{XCHL}$ | 29.8 | — | ns |
| 4, 5 | CLKOUT Rise and Fall Time | $t_{Crf}$ | — | 5 | ns |
| 4A, 5A | Rise and Fall Time (All Outputs except CLKOUT) | $t_{rf}$ | — | 8 | ns |
| 4B, 5B | External Clock Input Rise and Fall Time[3] | $t_{XCrf}$ | — | 5 | ns |
| 6 | Clock High to ADDR, FC, SIZE Valid[4] | $t_{CHAV}$ | 0 | 29 | ns |
| 7 | Clock High to ADDR, Data, FC, SIZE, High Impedance | $t_{CHAZx}$ | 0 | 59 | ns |
| 8 | Clock High to ADDR, FC, SIZE, Invalid | $t_{CHAZn}$ | 0 | — | ns |
| 9 | Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4] | $t_{CLSA}$ | 2 | 24 | ns |
| 9A | $\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5] | $t_{STSA}$ | -15 | 15 | ns |
| 11 | ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted | $t_{AVSA}$ | 15 | — | ns |
| 12 | Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated | $t_{CLSN}$ | 2 | 29 | ns |
| 13 | $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC SIZE Invalid (Address Hold) | $t_{SNAI}$ | 15 | — | ns |
| 14 | $\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted | $t_{SWA}$ | 100 | — | ns |
| 14A | $\overline{DS}$, $\overline{CS}$ Width Asserted (Write) | $t_{SWAW}$ | 45 | — | ns |
| 14B | $\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle) | $t_{SWDW}$ | 40 | — | ns |
| 15 | $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6] | $t_{SN}$ | 40 | — | ns |
| 16 | Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance | $t_{CHSZ}$ | — | 59 | ns |
| 17 | $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High | $t_{SNRN}$ | 15 | — | ns |
| 18 | Clock High to R/$\overline{W}$ High | $t_{CHRH}$ | 0 | 29 | ns |
| 20 | Clock High to R/$\overline{W}$ Low | $t_{CHRL}$ | 0 | 29 | ns |
| 21 | R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted | $t_{RAAA}$ | 15 | — | ns |
| 22 | R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write) | $t_{RASA}$ | 70 | — | ns |
| 23 | Clock High to Data Out Valid | $t_{CHDO}$ | — | 29 | ns |
| 24 | Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle) | $t_{DVASN}$ | 15 | — | ns |
| 25 | $\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold) | $t_{SNDOI}$ | 15 | — | ns |
| 26 | Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write) | $t_{DVSA}$ | 15 | — | ns |
| 27 | Data In Valid to Clock Low (Data Setup)[4] | $t_{DICL}$ | 5 | — | ns |
| 27A | Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time) | $t_{BELCL}$ | 20 | — | ns |
| 28 | $\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated | $t_{SNDN}$ | 0 | 80 | ns |
| 29 | $\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7] | $t_{SNDI}$ | 0 | — | ns |
| 29A | $\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8] | $t_{SHDI}$ | — | 55 | ns |

## Table A-19 Low Voltage 16.78-MHz Background Debug Mode Timing

($V_{DD}$ and $V_{DDSYN}$ = 2.7 to 3.6Vdc, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$)[1]

| Num | Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| B0 | DSI Input Setup Time | $t_{DSISU}$ | 15 | — | ns |
| B1 | DSI Input Hold Time | $t_{DSIH}$ | 15 | — | ns |
| B2 | DSCLK Setup Time | $t_{DSCSU}$ | 15 | — | ns |
| B3 | DSCLK Hold Time | $t_{DSCH}$ | 15 | — | ns |
| B4 | DSO Delay Time | $t_{DSOD}$ | — | 35 | ns |
| B5 | DSCLK Cycle Time | $t_{DSCCYC}$ | 2 | — | $t_{cyc}$ |
| B6 | CLKOUT High to FREEZE Asserted/Negated | $t_{FRZAN}$ | — | 50 | ns |
| B7 | CLKOUT High to IPIPE1 High Impedance | $t_{IFZ}$ | — | 50 | ns |
| B8 | CLKOUT High to IPIPE1 Valid | $t_{IF}$ | — | 50 | ns |
| B9 | DSCLK Low Time | $t_{DSCLO}$ | 1 | — | $t_{cyc}$ |
| B10 | IPIPE1 High Impedance to FREEZE Asserted | $t_{IPFA}$ | TBD | — | $t_{cyc}$ |
| B11 | FREEZE Negated to IPIPE[0:1] Active | $t_{FRIP}$ | TBD | — | $t_{cyc}$ |

NOTES:
1. All AC timing is shown with respect to $V_{IH}/V_{IL}$ levels unless otherwise noted.

## Table A-20 16.78-MHz Background Debug Mode Timing

($V_{DD}$ and $V_{DDSYN}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$)[1]

| Num | Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| B0 | DSI Input Setup Time | $t_{DSISU}$ | 15 | — | ns |
| B1 | DSI Input Hold Time | $t_{DSIH}$ | 10 | — | ns |
| B2 | DSCLK Setup Time | $t_{DSCSU}$ | 15 | — | ns |
| B3 | DSCLK Hold Time | $t_{DSCH}$ | 10 | — | ns |
| B4 | DSO Delay Time | $t_{DSOD}$ | — | 25 | ns |
| B5 | DSCLK Cycle Time | $t_{DSCCYC}$ | 2 | — | $t_{cyc}$ |
| B6 | CLKOUT High to FREEZE Asserted/Negated | $t_{FRZAN}$ | — | 50 | ns |
| B7 | CLKOUT High to IPIPE1 High Impedance | $t_{IFZ}$ | — | TBD | ns |
| B8 | CLKOUT High to IPIPE1 Valid | $t_{IF}$ | — | TBD | ns |
| B9 | DSCLK Low Time | $t_{DSCLO}$ | 1 | — | $t_{cyc}$ |
| B10 | IPIPE1 High Impedance to FREEZE Asserted | $t_{IPFA}$ | TBD | — | $t_{cyc}$ |
| B11 | FREEZE Negated to IPIPE[0:1] Active | $t_{FRIP}$ | TBD | — | $t_{cyc}$ |

NOTES:
1. All AC timing is shown with respect to $V_{IH}/V_{IL}$ levels unless otherwise noted.

## Table A-21 20.97-MHz Background Debug Mode Timing

($V_{DD}$ and $V_{DDSYN}$ = 5.0 Vdc ± 5%, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$)[1]

| Num | Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| B0 | DSI Input Setup Time | $t_{DSISU}$ | 15 | — | ns |
| B1 | DSI Input Hold Time | $t_{DSIH}$ | 10 | — | ns |
| B2 | DSCLK Setup Time | $t_{DSCSU}$ | 15 | — | ns |
| B3 | DSCLK Hold Time | $t_{DSCH}$ | 10 | — | ns |
| B4 | DSO Delay Time | $t_{DSOD}$ | — | 25 | ns |
| B5 | DSCLK Cycle Time | $t_{DSCCYC}$ | 2 | — | $t_{cyc}$ |
| B6 | CLKOUT High to FREEZE Asserted/Negated | $t_{FRZAN}$ | — | 50 | ns |
| B7 | CLKOUT High to IPIPE1 High Impedance | $t_{IFZ}$ | — | 50 | ns |
| B8 | CLKOUT High to IPIPE1 Valid | $t_{IF}$ | — | 50 | ns |
| B9 | DSCLK Low Time | $t_{DSCLO}$ | 1 | — | $t_{cyc}$ |
| B10 | IPIPE1 High Impedance to FREEZE Asserted | $t_{IPFA}$ | TBD | — | $t_{cyc}$ |
| B11 | FREEZE Negated to IPIPE[0:1] Active | $t_{FRIP}$ | TBD | — | $t_{cyc}$ |

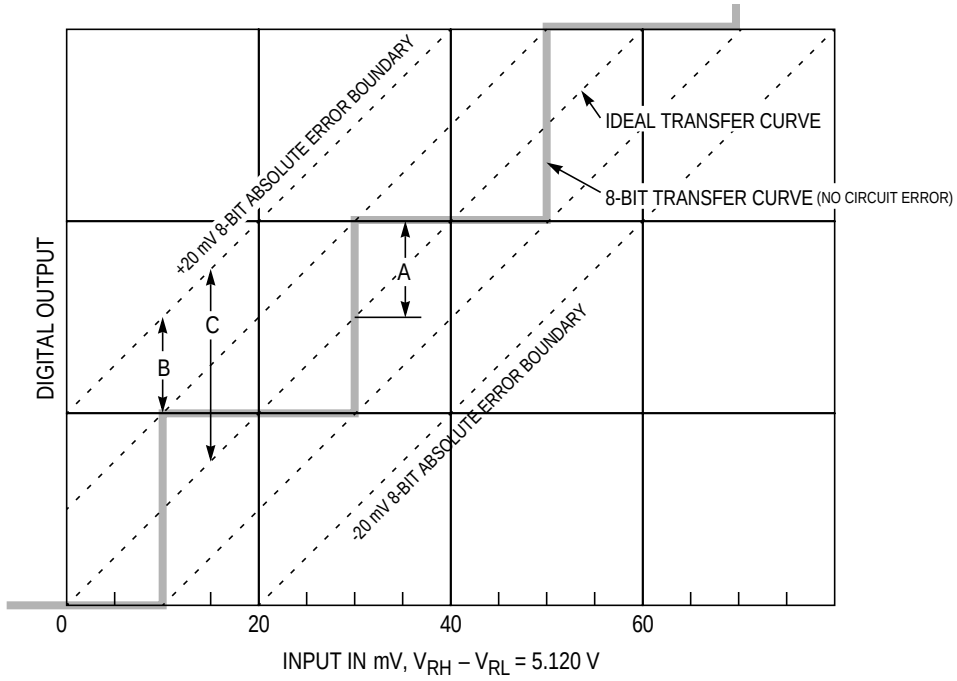
NOTES:
1. All AC timing is shown with respect to $V_{IH}/V_{IL}$ levels unless otherwise noted.

## Table A-22 25.17-MHz Background Debug Mode Timing

($V_{DD}$ and $V_{DDSYN}$ = 5.0 Vdc ± 5%, $V_{SS}$ = 0 Vdc, $T_A = T_L$ to $T_H$)[1]

| Num | Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|---|
| B0 | DSI Input Setup Time | $t_{DSISU}$ | 10 | — | ns |
| B1 | DSI Input Hold Time | $t_{DSIH}$ | 5 | — | ns |
| B2 | DSCLK Setup Time | $t_{DSCSU}$ | 10 | — | ns |
| B3 | DSCLK Hold Time | $t_{DSCH}$ | 5 | — | ns |
| B4 | DSO Delay Time | $t_{DSOD}$ | — | 20 | ns |
| B5 | DSCLK Cycle Time | $t_{DSCCYC}$ | 2 | — | $t_{cyc}$ |
| B6 | CLKOUT High to FREEZE Asserted/Negated | $t_{FRZAN}$ | — | 20 | ns |
| B7 | CLKOUT High to IPIPE1 High Impedance | $t_{IFZ}$ | — | 20 | ns |
| B8 | CLKOUT High to IPIPE1 Valid | $t_{IF}$ | — | 20 | ns |
| B9 | DSCLK Low Time | $t_{DSCLO}$ | 1 | — | $t_{cyc}$ |
| B10 | IPIPE1 High Impedance to FREEZE Asserted | $t_{IPFA}$ | TBD | — | $t_{cyc}$ |
| B11 | FREEZE Negated to IPIPE[0:1] Active | $t_{FRIP}$ | TBD | — | $t_{cyc}$ |

NOTES:
1. All AC timing is shown with respect to $V_{IH}/V_{IL}$ levels unless otherwise noted.




A – +1/2 COUNT (10 mV) INHERENT QUANTIZATION ERROR
B – CIRCUIT-CONTRIBUTED +10 mV ERROR
C – + 20 mV ABSOLUTE ERROR (ONE 8-BIT COUNT)

ADC 8-BIT ACCURACY

**Figure A-35 8-Bit ADC Conversion Accuracy**

ILT — Idle-Line Detect Type
0 = Short idle-line detect (start count on first one).
1 = Long idle-line detect (start count on first one after stop bit(s)).

PT — Parity Type
0 = Even parity
1 = Odd parity

PE — Parity Enable
0 = SCI parity disabled.
1 = SCI parity enabled.

M — Mode Select
0 = 10-bit SCI frame (1 start bit, 8 data bits, 1 stop bit)
1 = 11-bit SCI frame (1 start bit, 9 data bits, 1 stop bit)

WAKE — Wake-Up by Address Mark
0 = SCI receiver awakened by idle-line detection.
1 = SCI receiver awakened by address mark (last data bit set).

TIE — Transmit Interrupt Enable
0 = SCI TDRE interrupts disabled.
1 = SCI TDRE interrupts enabled.

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

RIE — Receiver Interrupt Enable
0 = SCI RDRF and OR interrupts disabled.
1 = SCI RDRF and OR interrupts enabled.

ILIE — Idle-Line Interrupt Enable
0 = SCI IDLE interrupts disabled.
1 = SCI IDLE interrupts enabled.

TE — Transmitter Enable
0 = SCI transmitter disabled (TXD pin can be used as I/O).
1 = SCI transmitter enabled (TXD pin dedicated to SCI transmitter).

RE — Receiver Enable
0 = SCI receiver disabled.
1 = SCI receiver enabled.

RWU — Receiver Wake-Up
0 = Normal receiver operation (received data recognized).
1 = Wake-up mode enabled (received data ignored until receiver is awakened).

SBK — Send Break
0 = Normal operation
1 = Break frame(s) transmitted after completion of the current frame.

# INDEX

## –A–