**Welcome to E-XFL.COM**

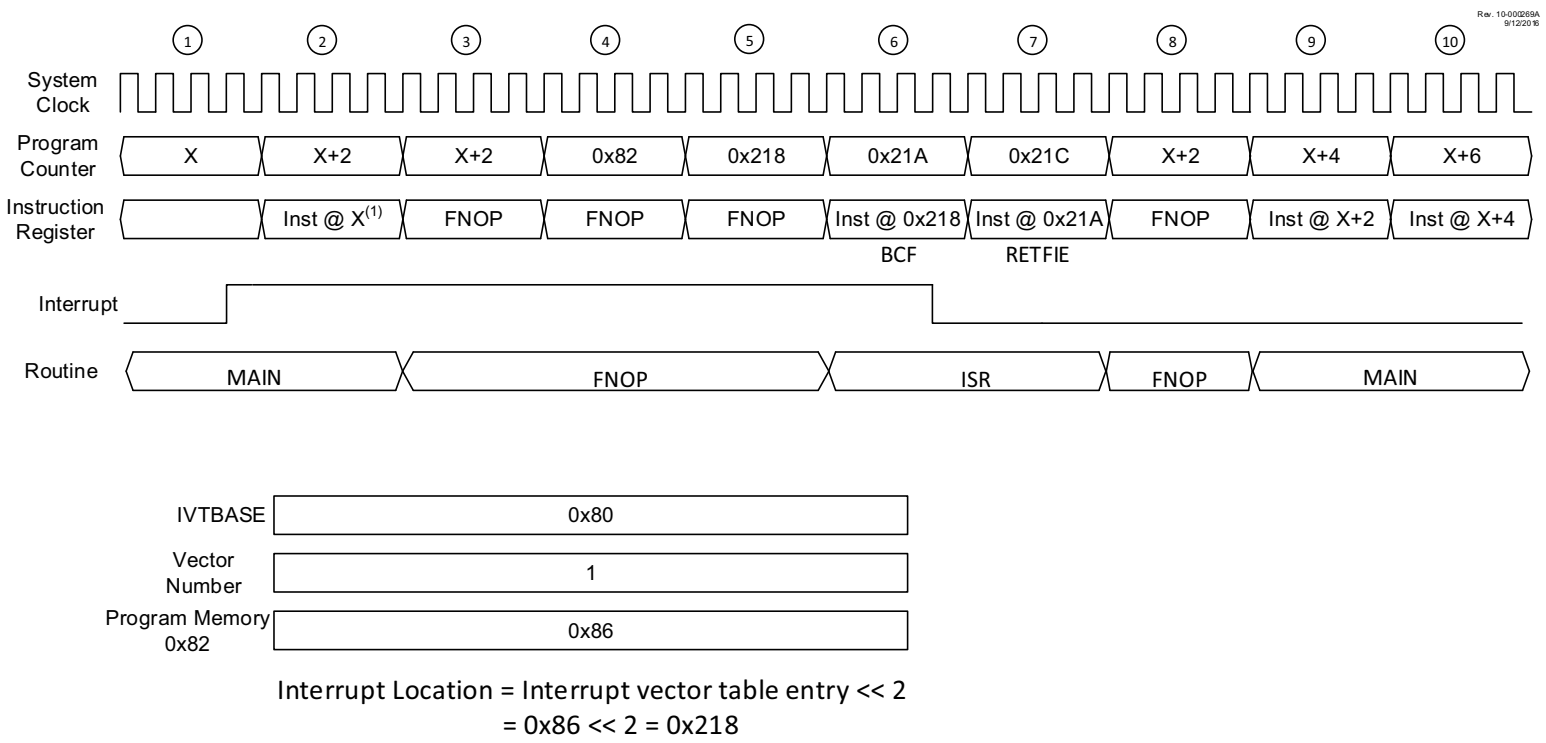
## What is "Embedded - Microcontrollers"?

"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 | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT |
| Number of I/O | 25 |
| Program Memory Size | 128KB (64K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 24x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 28-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f27k42t-i-ss |

**FIGURE 9-7: INTERRUPT TIMING DIAGRAM - ONE CYCLE INSTRUCTION**




**Note 1:** Instruction @ X is a One-cycle Instruction

### 13.1.5 ERASING PROGRAM FLASH MEMORY

The minimum erase block is 64 words (refer to Table 5-4). Only through the use of an external programmer, or through ICSP™ control, can larger blocks of program memory be bulk erased. Word erase in the program memory array is not supported.

For example, when initiating an erase sequence from a microcontroller with erase row size of 64 words, a block of 64 words (128 bytes) of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. The TBLPTR<5:0> bits are ignored.

The NVMCON1 register commands the erase operation. The REG<1:0> bits must be set to point to the Program Flash Memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

The NVM unlock sequence described in **Section 13.1.4 "NVM Unlock Sequence"** should be used to guard against accidental writes. This is sometimes referred to as a long write.

A long write is necessary for erasing program memory. Instruction execution is halted during the long write cycle. The long write is terminated by the internal programming timer.

#### 13.1.5.1 Program Flash Memory Erase Sequence

The sequence of events for erasing a block of internal program memory is:

1. REG bits of the NVMCON1 register point to PFM
2. Set the FREE and WREN bits of the NVMCON1 register
3. Perform the unlock sequence as described in **Section 13.1.4 "NVM Unlock Sequence"**

If the PFM address is write-protected, the WR bit will be cleared and the erase operation will not take place, WRERR is signaled in this scenario.

The operation erases the memory row indicated by masking the LSBs of the current TBLPTR.

While erasing PFM, CPU operation is suspended and it resumes when the operation is complete. Upon completion the WR bit is cleared in hardware, the NVMIF is set and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations and WREN will remain unchanged.

| | |
|---|---|
| **Note 1:** | If a write or erase operation is terminated by an unexpected event, WRERR bit will set which user can check to decide whether a rewrite of the location(s) is needed. |
| **2:** | WRERR is set if WR is written to '1' while TBLPTR points to a write-protected address. |
| **3:** | WRERR is set if WR is written to '1' while TBLPTR points to an invalid address location (Table 13-1). |

## 13.3 Data EEPROM Memory

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, which is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire VDD range.

Five SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- NVMCON1
- NVMCON2
- NVMDAT
- NVMADRL
- NVMADRH(1)

**Note 1:** NVMADRH register is not implemented on PIC18(L)F45/55K42.

The data EEPROM allows byte read and write. When interfacing to the data memory block, NVMDAT holds the 8-bit data for read/write and the NVMADRH:NVMADRL register pair holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an internal programming timer; it will vary with voltage and temperature as well as from chip-to-chip. Refer to the Data EEPROM Memory parameters in **Section 44.0 "Electrical Specifications"** for limits.

### 13.3.1 NVMADRL AND NVMADRH REGISTERS

The NVMADRH:NVMADRL registers are used to address the data EEPROM for read and write operations.

### 13.3.2 NVMCON1 AND NVMCON2 REGISTERS

Access to the data EEPROM is controlled by two registers: NVMCON1 and NVMCON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The NVMCON1 register (Register 13-1) is the control register for data and program memory access. Control bits REG<1:0> determine if the access will be to program, Data EEPROM Memory or the User IDs, Configuration bits, Revision ID and Device ID.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear.

The WRERR bit is set by hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is complete.

The WR control bit initiates write operations. The bit can be set but not cleared by software. It is cleared only by hardware at the completion of the write operation.

The NVMIF Interrupt Flag bit of the PIR0 register is set when the write is complete. It must be cleared by software.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (REG<1:0> = 0x10). Program memory is read using table read instructions. See **Section 13.1.1 "Table Reads and Table Writes"** regarding table reads.

## 14.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE WITH MEMORY SCANNER

The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- Configurable Polynomial
- Any seed value up to 16 bits can be used
- Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program/Data EEPROM memory user data
- Software loadable data registers for communication CRC's

### 14.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program/Data EEPROM memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using a scanner.

## 14.12 Register Definitions: CRC and Scanner Control

Long bit name prefixes for the CRC and Scanner peripherals are shown below. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

| Peripheral | Bit Name Prefix |
|---|---|
| CRC | CRC |

### REGISTER 14-1: CRCCON0: CRC CONTROL REGISTER 0

| R/W-0/0 | R/W-0/0 | R-0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R-0 |
|---|---|---|---|---|---|---|---|
| EN | GO | BUSY | ACCM | — | — | SHIFTM | FULL |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **EN:** CRC Enable bit
1 = CRC module is enabled
0 = CRC is disabled

bit 6 **GO:** CRC Go bit
1 = Start CRC serial shifter
0 = CRC serial shifter turned off

bit 5 **BUSY:** CRC Busy bit
1 = Shifting in progress or pending
0 = All valid bits in shifter have been shifted into accumulator

bit 4 **ACCM:** Accumulator Mode bit
1 = Data is concatenated with zeros
0 = Data is not concatenated with zeros

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **SHIFTM:** Shift Mode bit
1 = Shift right (LSb)
0 = Shift left (MSb)

bit 0 **FULL:** Data Path Full Indicator bit
1 = CRCDATH/L registers are full
0 = CRCDATH/L registers have shifted their data into the shifter

### REGISTER 14-2: CRCCON1: CRC CONTROL REGISTER 1

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| DLEN<3:0> | | | | PLEN<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-4 **DLEN<3:0>:** Data Length bits
Denotes the length of the data word -1 (See Example 14-1)

bit 3-0 **PLEN<3:0>:** Polynomial Length bits
Denotes the length of the polynomial -1 (See Example 14-1)

**FIGURE 15-3: DMA COUNTERS BLOCK DIAGRAM**

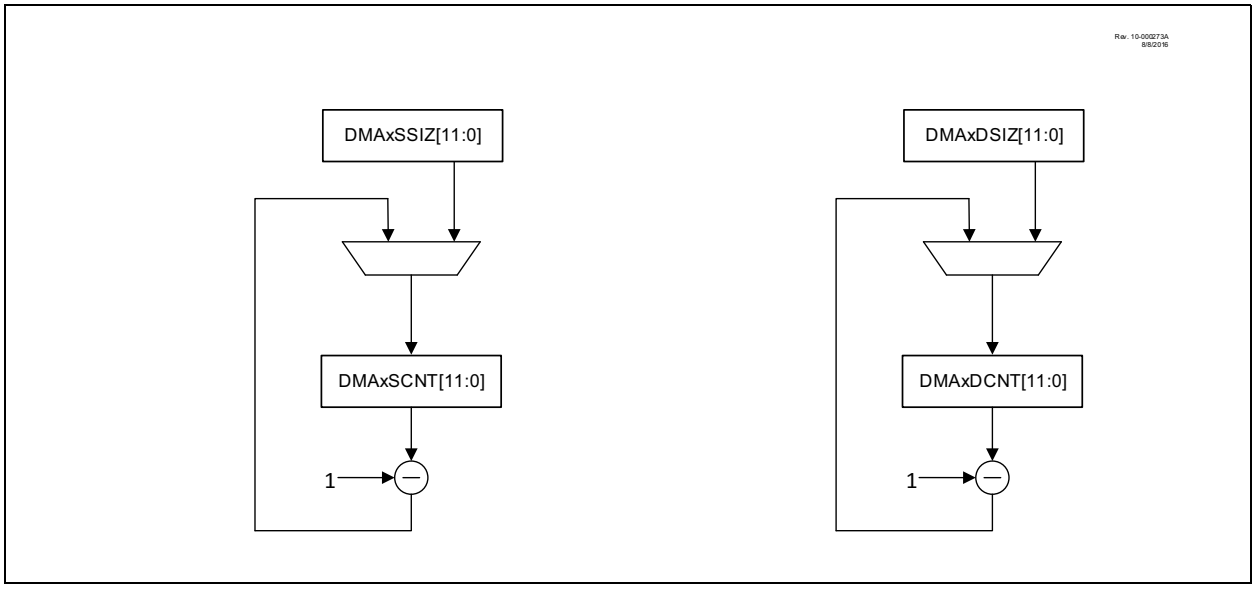



Table 15-2 has a few examples of configuring DMA Message sizes.

**TABLE 15-2: EXAMPLE MESSAGE SIZE TABLE**

| Operation | Example | SCNT | DCNT | Comments |
|---|---|---|---|---|
| Read from single SFR location to RAM | U1RXB | 1 | N | N equals the number of bytes desired in the destination buffer. N >= 1. |
| Write to single SFR location from RAM | U1TXB | N | 1 | N equals the number of bytes desired in the source buffer. N >= 1. |
| Read from multiple SFR location | ADRES[H:L] | 2 | 2*N | N equals the number of ADC results to be stored in memory. N>= 1 |
| | TMR1[H:L] | 2 | 2*N | N equals the number of TMR1 Acquisition results to be stored in memory. N>= 1 |
| | SMT1CPR[U:H:L] | 3 | 3*N | N equals the number of Capture Pulse Width measurements to be stored in memory. N>= 1 |
| Write to Multiple SFR registers | PWMDC[H:L] | 2*N | 2 | N equals the number of PWM duty cycle values to be loaded from a memory table. N>= 1 |
| | All ADC registers | N*31 | 31 | Using the DMA to transfer a complete ADC context from RAM to the ADC registers.N>= 1 |

## 16.0 I/O PORTS

The PIC18(L)F26/27/45/46/47/55/56/57K42 devices have six I/O ports, allocated as shown in Table 16-1.

**TABLE 16-1: PORT ALLOCATION TABLE FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES**

| Device | PORTA | PORTB | PORTC | PORTD | PORTE | PORTF |
|---|---|---|---|---|---|---|
| PIC18(L)F26K42 | • | • | • | | •(1) | |
| PIC18(L)F27K42 | • | • | • | | •(1) | |
| PIC18(L)F45K42 | • | • | • | • | •(2) | |
| PIC18(L)F46K42 | • | • | • | • | •(2) | |
| PIC18(L)F47K42 | • | • | • | • | •(2) | |
| PIC18(L)F55K42 | • | • | • | • | •(2) | • |
| PIC18(L)F56K42 | • | • | • | • | •(2) | • |
| PIC18(L)F57K42 | • | • | • | • | •(2) | • |

**Note 1:** Pin RE3 only.
**2:** Pins RE0, RE1, RE2 and RE3 only.

Each port has ten registers to control the operation. These registers are:

- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- TRISx registers (data direction)
- ANSELx registers (analog select)
- WPUx registers (weak pull-up)
- INLVLx (input level control)
- SLRCONx registers (slew rate control)
- ODCONx registers (open-drain control)

Most port pins share functions with device peripherals, both analog and digital. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output; however, the pin can still be read.

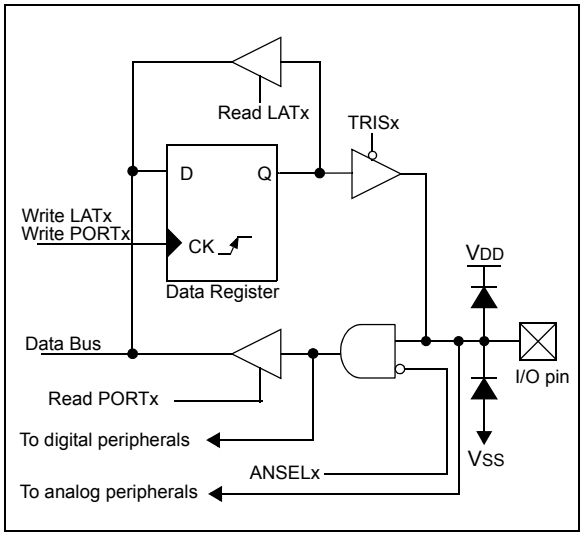
The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSELx bit is set, the digital input buffer associated with that bit is disabled.

Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 16-1.

**FIGURE 16-1: GENERIC I/O PORT OPERATION**




## 16.1 I/O Priorities

Each pin defaults to the PORT data latch after Reset. Other functions are selected with the peripheral pin select logic. See **Section 17.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx register. Digital output functions may continue to control the pin when it is in Analog mode.

Analog outputs, when enabled, take priority over digital outputs and force the digital output driver into a high-impedance state.

The pin function priorities are as follows:

1. Configuration bits
2. Analog outputs (disable the input buffers)
3. Analog inputs
4. Port inputs and outputs from PPS

## 16.2 PORTx Registers

In this section, the generic names such as PORTx, LATx, TRISx, etc. can be associated with PORTA, PORTB, and PORTC. The functionality of PORTE is different compared to other ports and is explained in a separate section.

**TABLE 17-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| PPSLOCK | — | — | — | — | — | — | — | PPSLOCKED | 283 |
| INT0PPS | — | — | — | INT0PPS<4:0> | | | | | 277 |
| INT1PPS | — | — | — | INT1PPS<4:0> | | | | | 277 |
| INT2PPS | — | — | — | INT2PPS<4:0> | | | | | 277 |
| T0CKIPPS | — | — | — | T0CKIPPS<4:0> | | | | | 277 |
| T1CKIPPS | — | — | — | T1CKIPPS<4:0> | | | | | 277 |
| T1GPPS | — | — | — | T1GPPS<4:0> | | | | | 277 |
| T3CKIPPS | — | — | — | T3CKIPPS<4:0> | | | | | 277 |
| T3GPPS | — | — | — | T3GPPS<4:0> | | | | | 277 |
| T5CKIPPS | — | — | — | T5CKIPPS<4:0> | | | | | 277 |
| T5GPPS | — | — | — | T5GPPS<4:0> | | | | | 277 |
| T2INPPS | — | — | — | T2INPPS<4:0> | | | | | 277 |
| T4INPPS | — | — | — | T4INPPS<4:0> | | | | | 277 |
| T6INPPS | — | — | — | T6INPPS<4:0> | | | | | 277 |
| CCP1PPS | — | — | — | CCP1PPS<4:0> | | | | | 277 |
| CCP2PPS | — | — | — | CCP2PPS<4:0> | | | | | 277 |
| CCP3PPS | — | — | — | CCP3PPS<4:0> | | | | | 277 |
| CCP4PPS | — | — | — | CCP4PPS<4:0> | | | | | 277 |
| SMT1WINPPS | — | — | — | SMT1WINPPS<4:0> | | | | | 277 |
| SMT1SIGPPS | — | — | — | SMT1SIGPPS<4:0> | | | | | 277 |
| CWG1PPS | — | — | — | CWG1PPS<4:0> | | | | | 277 |
| CWG2PPS | — | — | — | CWG2PPS<4:0> | | | | | 277 |
| CWG3PPS | — | — | — | CWG3PPS<4:0> | | | | | 277 |
| MD1CARLPPS | — | — | — | MDCARLPPS<4:0> | | | | | 277 |
| MD1CARHPPS | — | — | — | MDCARHPPS<4:0> | | | | | 277 |
| MD1SRCPPS | — | — | — | MDSRCPPS<4:0> | | | | | 277 |
| CLCIN0PPS | — | — | — | CLCIN0PPS<4:0> | | | | | 277 |
| CLCIN1PPS | — | — | — | CLCIN1PPS<4:0> | | | | | 277 |
| CLCIN2PPS | — | — | — | CLCIN2PPS<4:0> | | | | | 277 |
| CLCIN3PPS | — | — | — | CLCIN3PPS<4:0> | | | | | 277 |
| ADACTPPS | — | — | — | ADACTPPS<4:0> | | | | | 277 |
| SPI1SCKPPS | — | — | — | SPI1SCKPPS<4:0> | | | | | 277 |
| SPI1SDIPPS | — | — | — | SPI1SDIPPS<4:0> | | | | | 277 |
| SPI1SSPPS | — | — | — | SPI1SSPPS<4:0> | | | | | 277 |
| I2C1SCLPPS | — | — | — | I2C1SCLPPS<4:0> | | | | | 277 |
| I2C1SDAPPS | — | — | — | I2C1SDAPPS<4:0> | | | | | 277 |
| I2C2SCLPPS | — | — | — | I2C2SCLPPS<4:0> | | | | | 277 |
| I2C2SDAPPS | — | — | — | I2C2SDAPPS<4:0> | | | | | 277 |
| U1RXPPS | — | — | — | U1RXPPS<4:0> | | | | | 277 |
| U1CTSPPS | — | — | — | U1CTSPPS<4:0> | | | | | 277 |
| U2RXPPS | — | — | — | U2RXPPS<4:0> | | | | | 277 |
| U2CTSPPS | — | — | — | U2CTPPS<4:0> | | | | | 277 |
| RxyPPS | — | — | — | RxyPPS<4:0> | | | | | 280 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the PPS module.

## 19.5 Register Definitions: Peripheral Module Disable

**REGISTER 19-1: PMD0: PMD CONTROL REGISTER 0**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| SYSCMD | FVRMD | HLVDMD | CRCMD | SCANMD | NVMMD | CLKRMD | IOCMD |
| 7 | | | | | | | 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

bit 7 **SYSCMD:** Disable Peripheral System Clock Network bit(1)
See description in **Section 19.4 "System Clock Disable"**.
1 = System clock network disabled (FOSC)
0 = System clock network enabled

bit 6 **FVRMD:** Disable Fixed Voltage Reference bit
1 = FVR module disabled
0 = FVR module enabled

bit 5 **HLVDMD:** Disable High/Low-Voltage Detect bit
1 = HLVD module disabled
0 = HLVD module enabled

bit 4 **CRCMD:** Disable CRC Engine bit
1 = CRC module disabled
0 = CRC module enabled

bit 3 **SCANMD:** Disable NVM Memory Scanner bit(2)
1 = NVM Memory Scan module disabled
0 = NVM Memory Scan module enabled

bit 2 **NVMMD:** NVM Module Disable bit(3)
1 = All Memory reading and writing is disabled; NVMCON registers cannot be written
0 = NVM module enabled

bit 1 **CLKRMD:** Disable Clock Reference bit
1 = CLKR module disabled
0 = CLKR module enabled

bit 0 **IOCMD:** Disable Interrupt-on-Change bit, All Ports
1 = IOC module(s) disabled
0 = IOC module(s) enabled

**Note 1:** Clearing the SYSCMD bit disables the system clock (FOSC) to peripherals, however peripherals clocked by FOSC/4 are not affected.
**2:** Subject to SCANE bit in CONFIG4H.
**3:** When enabling NVM, a delay of up to 1 µs may be required before accessing data.

## 29.10 Register Definitions: ZCD Control

**REGISTER 29-1: ZCDCON: ZERO-CROSS DETECT CONTROL REGISTER**

| R/W-0/0 | U-0 | R-x | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| SEN | — | OUT | POL | — | — | INTP | INTN |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **SEN:** Zero-Cross Detect Software Enable bit
This bit is ignored when ZCDSEN configuration bit is set.
1= Zero-cross detect is enabled.
0= Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

bit 6 **Unimplemented:** Read as '0'

bit 5 **OUT:** Zero-Cross Detect Data Output bit
ZCDPOL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current
ZCDPOL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current

bit 4 **POL:** Zero-Cross Detect Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **INTP:** Zero-Cross Detect Positive-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCD_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCD_output transition

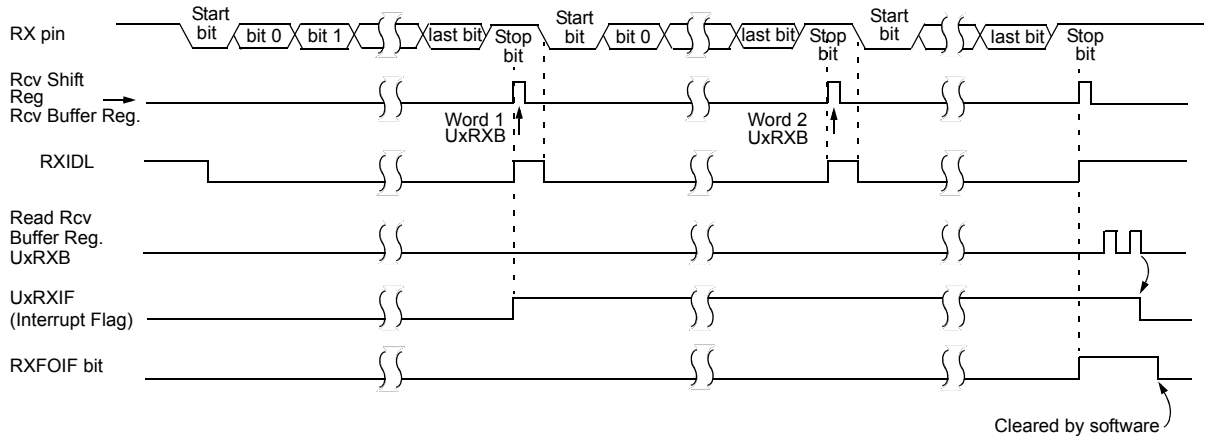
bit 0 **INTN:** Zero-Cross Detect Negative-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCD_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCD_output transition

**TABLE 29-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| ZCDCON | SEN | — | OUT | POL | — | — | INTP | INTN | 462 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

**FIGURE 31-5: ASYNCHRONOUS RECEPTION**




**Note:** This timing diagram shows three words appearing on the RX input. The UxRXB (receive buffer) is not read before the third word is received, causing the RXFOIF (FIFO overrun) bit to be set. STPMD = 0, STP<1:0> = 00.

## 32.4 Transfer Counter

In all master modes, the transfer counter can be used to determine how many data transfers the SPI will send/receive. The transfer counter is comprised of the SPIxTCTH/L set of registers, and is also partially controlled by the SPIxTWIDTH register. The Transfer Counter has two primary modes, determined by the BMODE bit of the SPIxCON0 register. Each mode uses the SPIxTCTH/L and SPIxTWIDTH registers to determine the number and size of the transfers. In both modes, when the transfer counter reaches zero, the TCZIF interrupt flag is set.

| | |
|---|---|
| **Note:** | When BMODE=1 in all master modes (and at all times in slave modes), the Transfer Counter will still decrement as transfers occur and can be used to count the number of messages sent/received, as well as to control SS(out) and to trigger TCZIF. Also when BMODE = 1, the SPIxTWIDTH register can be used in Master and Slave modes to determine the size of messages sent and received by the SPI, even if the Transfer Counter is not being actively used to control the number of messages being sent/received by the SPI module. |

### 32.4.1 TOTAL BIT COUNT MODE (BMODE = 0)

In this mode, SPIxTCTH/L and SPIxTWIDTH are concatenated to determine the total number of bits to be transferred. These bits will be loaded from/into the transmit/receive FIFOs in 8-bit increments and the transfer counter will be decremented by eight until the total number of remaining bits is less than eight. If there are any remaining bits (SPIxTWIDTH ≠ 0), the transmit FIFO will send out one final message with any extra bits greater than the remainder ignored. The SPIxTWIDTH is the remaining bit count but the value does not change as it does for the SPIxTCT value. Similarly, the receiver will load a final byte into the receiver FIFO, and pad the extra bits with zeros. The LSBF bit of SPIxCON0 determines whether the Most Significant or Least Significant bits of this final byte are ignored/padded. For example, when LSBF = 0 and the final transfer contains only two bits then if the last byte sent was 5Fh then the RXB of the receiver will contain 40h which are the two MSbits of the final byte padded with zeros in the LSbits.

In this mode, the SPI master will only transmit messages when the SPIxTCT value is greater than zero, regardless of TXR and RXR settings. In Master Transmit mode, the transfer starts with the data write to the SPIxTXB register or the count value written to the SPIxTCTL register, which ever occurs last. In Master Receive-only mode, the transfer clocks start when the SPIxTCTL value is written. Transfer clocks are suspended when the receive FIFO is full and resume as the FIFO is read.

### 32.4.2 VARIABLE TRANSFER SIZE MODE (BMODE = 1)

In this mode, SPIxTWIDTH specifies the width of every individual piece of the data transfer in bits. SPIxTCTH/SPIxTCTL specifies the number of transfers of this bit length. If SPIxTWIDTH = 0, each piece is a full byte of data. If SPIxTWIDTH ≠ 0, then only the specified number of bits from the transmit FIFO are shifted out, with the unused bits ignored. Received data is padded with zeros in the unused bit areas when transfered into the receive FIFO. The LSBF bit of SPIxCON0 determines whether the Most Significant or Least Significant bits of the transfers are ignored/padded. In this mode, the transfer counter being zero only stops messages from being sent/received when in “Receive only” mode.

| | |
|---|---|
| **Note:** | With BMODE = 1, it is possible for the transfer counter (SPIxTCTH/L) to decrement below zero, although when in “Receive only” Master mode, transfer clocks will cease when the transfer counter reaches zero. |

**FIGURE 33-8: I²C SLAVE, 7-BIT ADDRESS, RECEPTION NO I2CxCNT (ACKTIE = 0, ADRIE = 1, WRIE = 1)**

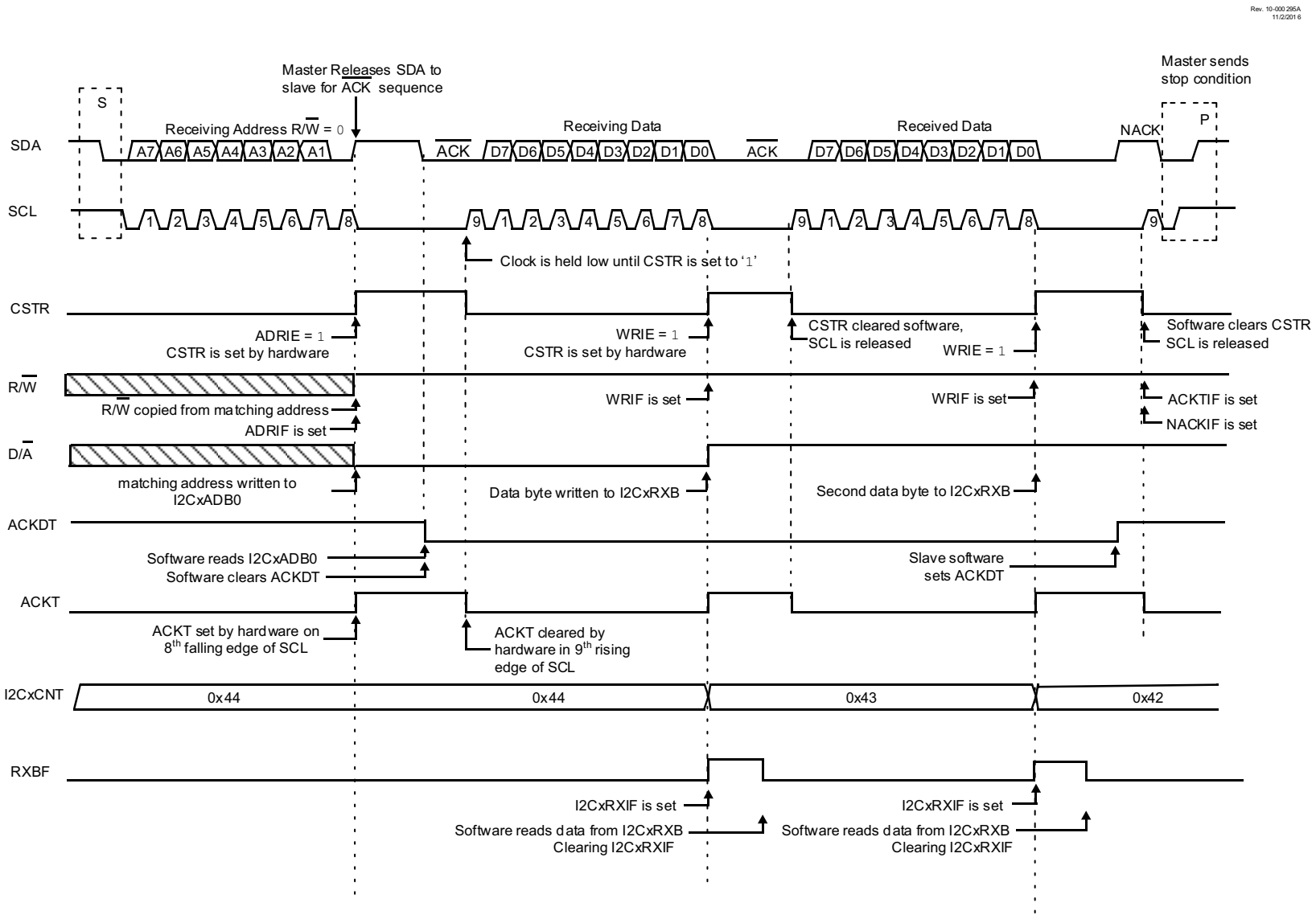


**FIGURE 33-11: $I^2C$ SLAVE, 10-BIT ADDRESS, RECEPTION WITH STOP (ADB = 1)**

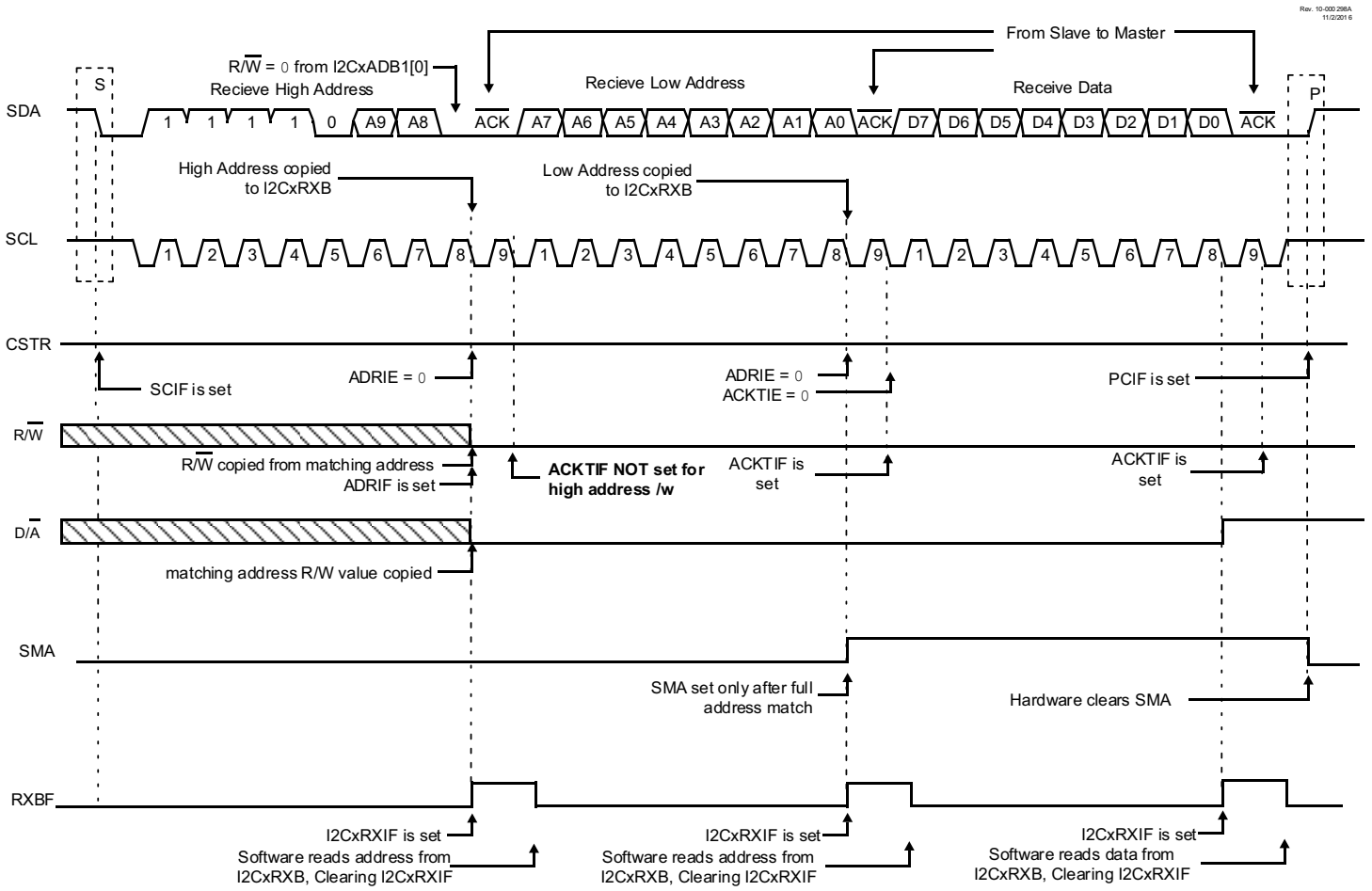
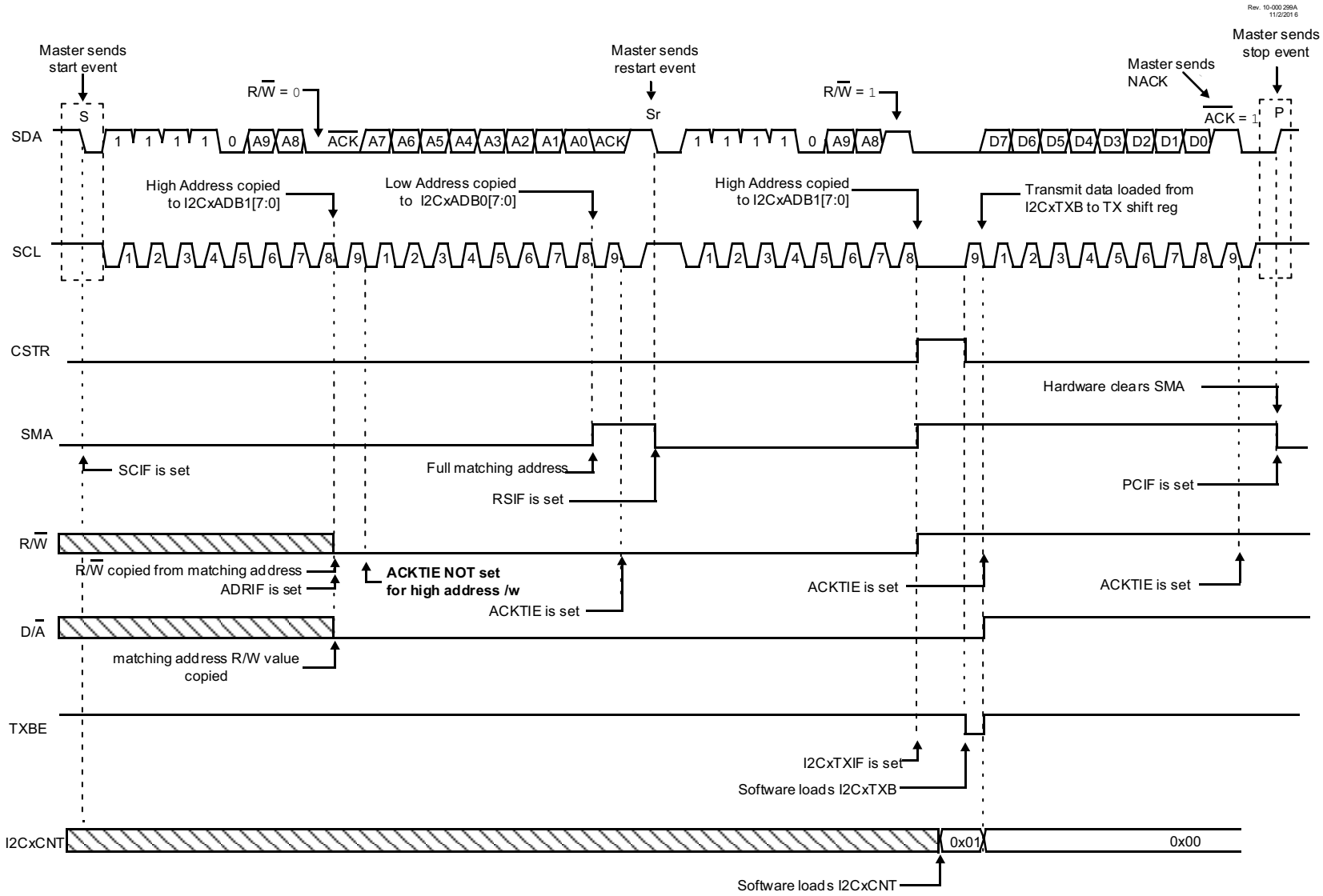


**FIGURE 33-12: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION**



## 38.9 CWG1 Auto-Shutdown Source

The output of the comparator module can be used as an auto-shutdown source for the CWG1 module. When the output of the comparator is active and the corresponding WGASxE is enabled, the CWG operation will be suspended immediately (see **Section 26.10.1.2 "External Input Source"**).

## 38.10 ADC Auto-Trigger Source

The output of the comparator module can be used to trigger an ADC conversion. When the ADACT register is set to trigger on a comparator output, an ADC conversion will trigger when the Comparator output goes high.

## 38.11 TMR2/4/6 Reset

The output of the comparator module can be used to reset Timer2. When the TxRST register is appropriately set, the timer will reset when the Comparator output goes high.

## 38.12 Operation in Sleep Mode

The comparator module can operate during Sleep. The comparator clock source is based on the Timer1 clock source. If the Timer1 clock source is either the system clock (FOSC) or the instruction clock (FOSC/4), Timer1 will not operate during Sleep, and synchronized comparator outputs will not operate.

A comparator interrupt will wake the device from Sleep. The CxIE bits of the respective PIE register must be set to enable comparator interrupts.

**ADDWFC** **ADD W and CARRY bit to f**

Syntax: ADDWFC f {,d {,a}}

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (W) + (f) + (C) → dest

Status Affected: N,OV, C, DC, Z

Encoding:

| 0010 | 00da | ffff | ffff |
|---|---|---|---|

Description: Add W, the CARRY flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write to destination |

Example: ADDWFC REG, 0, 1

Before Instruction
CARRY bit = 1
REG = 02h
W = 4Dh
After Instruction
CARRY bit = 0
REG = 02h
W = 50h

**ANDLW** **AND literal with W**

Syntax: ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. k → W

Status Affected: N, Z

Encoding:

| 0000 | 1011 | kkkk | kkkk |
|---|---|---|---|

Description: The contents of W are AND'ed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: ANDLW 05Fh

Before Instruction
W = A3h
After Instruction
W = 03h

**CALLW** **Subroutine Call Using WREG**

Syntax: CALLW

Operands: None

Operation: (PC + 2) → TOS,
(W) → PCL,
(PCLATH) → PCH,
(PCLATU) → PCU

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0001 | 0100 |

Description: First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched. Unlike CALL, there is no option to update W, Status or BSR.

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read WREG | PUSH PC to stack | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE    CALLW
```
Before Instruction
PC = address (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h
After Instruction
PC = 001006h
TOS = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h

**CLRF** **Clear f**

Syntax: CLRF f {,a}

Operands: 0 ≤ f ≤ 255
a ∈ [0,1]

Operation: 000h → f
1 → Z

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 0110 | 101a | ffff | ffff |

Description: Clears the contents of the specified register.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write register 'f' |

Example:
```
CLRF    FLAG_REG, 1
```
Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

**RETURN** **Return from Subroutine**

Syntax: RETURN {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
if s = 1
(WS) → W,
(STATUSS) → Status,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0001 | 001s |

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers, WS, STATUSS and BSRS, are loaded into their corresponding registers, W, Status and BSR. If 's' = 0, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No operation | Process Data | POP PC from stack |
| No operation | No operation | No operation | No operation |

Example: RETURN

After Instruction:
PC = TOS

**RLCF** **Rotate Left f through Carry**

Syntax: RLCF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

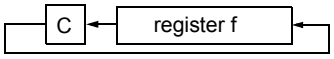
Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

Encoding:

| | | | |
|---|---|---|---|
| 0011 | 01da | ffff | ffff |

Description: The contents of register 'f' are rotated one bit to the left through the CARRY flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.




Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write to destination |

Example: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

**TABLE 44-15: ANALOG-TO-DIGITAL CONVERTER (ADC) ACCURACY SPECIFICATIONS(1,2)**

| **Operating Conditions (unless otherwise stated)**<br>VDD = 3.0V, TA = 25°C, TAD = 1µs | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD01 | NR | Resolution | — | — | 12 | bit | |
| AD02 | EIL | Integral Error | — | ±0.1 | ±2.0 | LSb | ADCREF+ = 3.0V, ADCREF-= 0V |
| AD03 | EDL | Differential Error | — | ±0.1 | ±1.0 | LSb | ADCREF+ = 3.0V, ADCREF-= 0V |
| AD04 | EOFF | Offset Error | — | 0.5 | 6.0 | LSb | ADCREF+ = 3.0V, ADCREF-= 0V |
| AD05 | EGN | Gain Error | — | ±0.2 | ±6.0 | LSb | ADCREF+ = 3.0V, ADCREF-= 0V |
| AD06 | VADREF | ADC Reference Voltage (ADREF+ - ADREF-) | 1.8 | — | VDD | V | |
| AD07 | VAIN | Full-Scale Range | ADREF- | — | ADREF+ | V | |
| AD08 | ZAIN | Recommended Impedance of Analog Voltage Source | — | 1 | — | kΩ | |
| AD09 | RVREF | ADC Voltage Reference Ladder Impedance | — | 50 | — | kΩ | **Note 3** |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Total Absolute Error is the sum of the offset, gain and integral non-linearity (INL) errors.

**2:** The ADC conversion result never decreases with an increase in the input and has no missing codes.

**3:** This is the impedance seen by the VREF pads when the external reference pads are selected.