**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, IrDA, LINbus, SPI, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT |
| Number of I/O | 68 |
| Program Memory Size | 128KB (64K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 3.6V |
| Data Converters | A/D 24x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 80-TQFP |
| Supplier Device Package | 80-TQFP (12x12) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f87j94t-i-pt |

### 4.2.3 IDLE MODE

When the device enters Idle mode, the following events occur:

- The CPU will stop executing instructions.
- The WDT is automatically cleared.
- The system clock source will remain active and the peripheral modules, by default, will continue to operate normally from the system clock source. Peripherals can optionally be shut down in Idle mode using their 'Stop in Idle' control bit. (See peripheral descriptions for further details.)
- If the WDT or FSCM is enabled, the LPRC will also remain active.

The processor will wake-up from Idle mode on the following events:

- On any interrupt that is individually enabled.
- On any source of device Reset.
- On a WDT time-out.

Upon wake-up from Idle mode, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the SLEEP instruction, or the first instruction in the Interrupt Service Routine (ISR).

#### 4.2.3.1 Time Delays on Wake-up from Idle Mode

Unlike a wake-up from Sleep mode, there are no additional time delays associated with wake-up from Idle mode. The system clock is running during Idle mode, therefore, no start-up times are required at wake-up.

#### 4.2.3.2 Wake-up from Idle on Interrupt

Any source of interrupt that is individually enabled using the corresponding control bit in the PIEx register, will be able to wake-up the processor from Idle mode. When the device wakes from Idle mode, one of two options may occur:

- If the GIE bit is set, the processor will wake and the Program Counter will begin execution at the interrupt vector.
- If the GIE bit is not set, the processor will wake and the Program Counter will continue execution following the SLEEP instruction.

The $\overline{PD}$ Status bit (RCON<2>) is set upon wake-up.

#### 4.2.3.3 Wake-up from Idle on Reset

Any Reset, other than a Power-on Reset (POR), will wake-up the CPU from Idle mode on any device Reset, except a POR.

#### 4.2.3.4 Wake-up from Idle on WDT Time-out

If the WDT is enabled, then the processor will wake-up from Idle mode on a WDT time-out and continue code execution with the instruction following the SLEEP instruction that initiated Idle mode. Note that the WDT time-out does not reset the device in this case. The $\overline{TO}$ bit (RCON<3>) will be set.

### 4.2.4 SLEEP MODES

Most 08KA101 family devices that incorporate power-saving features and VBAT, offer two distinct Sleep modes: Sleep mode and Retention Sleep mode. The characteristics of both Sleep modes are:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be optimum, provided no I/O pin is sourcing the current.
- The Fail-Safe Clock Monitor (FSCM) does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT is enabled.
- If Brown-out Reset (BOR) is enabled, the Brown-out Reset (BOR) circuit remains operational during Sleep mode.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some peripherals may continue to operate in Sleep mode. These peripherals include I/O pins that detect a change in the input signal or peripherals that use an external clock input. Any peripheral that operates from the system clock source will be disabled in Sleep mode.

The processor will exit, or 'wake-up' from Sleep on one of the following events:

- On any interrupt source that is individually enabled
- On any form of device Reset
- On a WDT time-out

**TABLE 6-2: REGISTER FILE SUMMARY (CONTINUED)**

| File Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| F69h | SSP1CON3 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN |
| F68h | SSP1MSK | MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 |
| F67h | BAUDCON1 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | IREN | WUE | ABDEN |
| F66h | OSCCON2 | CLKLOCK | IOLOCK | LOCK | — | CF | POSCEN | SOSCGO | — |
| F65h | OSCCON3 | — | — | — | — | — | IRCF2 | IRCF1 | IRCF0 |
| F64h | OSCCON4 | CPDIV1 | CPDIV0 | PLLEN | — | — | — | — | — |
| F63h | ACTCON | ACTEN | — | ACTSIDL | ACTSRC | ACTLOCK | ACTLOCKPOL | ACTORS | ACTORSPOL |
| F62h | WPUB | WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 |
| F61h | PIE6 | RC4IE | TX4IE | RC3IE | TX3IE | — | CMP3IE | CMP2IE | CMP1IE |
| F60h | DMACON1 | SSCON1 | SSCON0 | TXINC | RXINC | DUPLEX1 | DUPLEX0 | DLYINTEN | DMAEN |
| F5Fh | RTCCON1 | RTCEN | — | RTCWREN | RTCSYNC | HALFSEC | RTCOE | RTCPTR1 | RTCPTR0 |
| F5Eh | RTCCAL | CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 |
| F5Dh | RTCVALH | RTCC Value High Register Window Based on RTCPTR<1:0> | | | | | | | |
| F5Ch | RTCVALL | RTCC Value Low Register Window Based on RTCPTR<1:0> | | | | | | | |
| F5Bh | ALRMCFG | ALRMEN | CHIME | AMASK3 | AMASK2 | AMASK1 | AMASK0 | ALRMPTR1 | ALRMPTR0 |
| F5Ah | ALRMRPT | ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 |
| F59h | ALRMVALH | Alarm Value High Register Window Based on APTR<1:0> | | | | | | | |
| F58h | ALRMVALL | Alarm Value Low Register Window Based on APTR<1:0> | | | | | | | |
| F57h | RTCCON2 | PWCEN | PWCPOL | PWCCPRE | PWCSPRE | RTCCLKSEL1 | RTCCLKSEL0 | RTCSECSEL1 | RTCSECSEL0 |
| F56h | IOCP | IOCP7 | IOCP6 | IOCP5 | IOCP4 | IOCP3 | IOCP2 | IOCP1 | IOCP0 |
| F55h | IOCN | IOCN7 | IOCN6 | IOCN5 | IOCN4 | IOCN3 | IOCN2 | IOCN1 | IOCN0 |
| F54h | PADCFG1 | RDPU | REPU | RFPU | RGPU | RHPU | RJPU | RKPU | RLPU |
| F53h | CM1CON | CON | COE | CPOL | EVPOL1 | EVPOL0 | CREF | CCH1 | CCH0 |
| F52h | ECCP2AS | ECCP2ASE | ECCP2AS2 | ECCP2AS1 | ECCP2AS0 | PSS2AC1 | PSS2AC0 | PSS2BD1 | PSS2BD0 |
| F51h | ECCP2DEL | P2RSEN | P2DC6 | P2DC5 | P2DC4 | P2DC3 | P2DC2 | P2DC1 | P2DC0 |
| F50h | CCPR2H | Capture/Compare/PWM Register 1 High Byte | | | | | | | |
| F4Fh | CCPR2L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | |
| F4Eh | CCP2CON | P2M1 | P2M0 | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 |
| F4Dh | ECCP3AS | ECCP3ASE | ECCP3AS2 | ECCP3AS1 | ECCP3AS0 | PSS3AC1 | PSS3AC0 | PSS3BD1 | PSS3BD0 |
| F4Ch | ECCP3DEL | P3RSEN | P3DC6 | P3DC5 | P3DC4 | P3DC3 | P3DC2 | P3DC1 | P3DC0 |
| F4Bh | CCPR3H | Capture/Compare/PWM Register 1 High Byte | | | | | | | |
| F4Ah | CCPR3L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | |
| F49H | CCP3CON | P3M1 | P3M0 | CCP3X | CCP3Y | CCP3M3 | CCP3M2 | CCP3M1 | CCP3M0 |
| F48h | CCPR8H | Capture/Compare/PWM Register 8 High Byte | | | | | | | |
| F47h | CCPR8L | Capture/Compare/PWM Register 8 Low Byte | | | | | | | |
| F46h | CCP8CON | — | — | CCP8X | CCP8Y | CCP8M3 | CCP8M2 | CCP8M1 | CCP8M0 |
| F45h | CCPR9H | Capture/Compare/PWM Register 9 High Byte | | | | | | | |
| F44h | CCPR9L | Capture/Compare/PWM Register 9 Low Byte | | | | | | | |
| F43h | CCP9CON | — | — | CCP9X | CCP9Y | CCP9M3 | CCP9M2 | CCP9M1 | CCP9M0 |
| F42h | CCPR10H | Capture/Compare/PWM Register 10 High Byte | | | | | | | |
| F41h | CCPR10L | Capture/Compare/PWM Register 10 Low Byte | | | | | | | |
| F40h | CCP10CON | — | — | CCP10X | CCP10Y | CCP10M3 | CCP10M2 | CCP10M1 | CCP10M0 |
| F3Fh | TMR6 | Timer6 Register | | | | | | | |
| F3Eh | PR6 | Timer6 Period Register | | | | | | | |
| F3Dh | T6CON | — | T6OUTPS3 | T6OUTPS2 | T6OUTPS1 | T6OUTPS0 | TMR6ON | T6CKPS1 | T6CKPS0 |
| F3Ch | TMR8 | Timer8 Register | | | | | | | |
| F3Bh | PR8 | Timer8 Period Register | | | | | | | |
| F3Ah | T8CON | — | T8OUTPS3 | T8OUTPS2 | T8OUTPS1 | T8OUTPS0 | TMR8ON | T8CKPS1 | T8CKPS0 |
| F39H | SSP2CON3 | ACKTIM | PCIE | SCIE | BOEN | SDAHT | SBCDE | AHEN | DHEN |
| F38h | CM2CON | CON | COE | CPOL | EVPOL1 | EVPOL0 | CREF | CCH1 | CCH0 |
| F37h | CM3CON | CON | COE | CPOL | EVPOL1 | EVPOL0 | CREF | CCH1 | CCH0 |

**Legend:** — = unimplemented, read as '0'.

## 8.0 EXTERNAL MEMORY BUS

**Note:** The External Memory Bus is not implemented on 64-pin devices.

The External Memory Bus (EMB) allows the device to access external memory devices (such as Flash, EPROM or SRAM) as program or data memory. It supports both 8 and 16-Bit Data Width modes, and three address widths of up to 20 bits.

The bus is implemented with 28 pins, multiplexed across four I/O ports. Three ports (PORTD, PORTE and PORTH) are multiplexed with the address/data bus for a total of 20 available lines, while PORTJ is multiplexed with the bus control signals.

A list of the pins and their functions is provided in Table 8-1.

**TABLE 8-1: PIC18F97J94 FAMILY EXTERNAL BUS – I/O PORT FUNCTIONS**

| Name | Port | Bit | External Memory Bus Function |
|---|---|---|---|
| RD0/AD0 | PORTD | 0 | Address Bit 0 or Data Bit 0 |
| RD1/AD1 | PORTD | 1 | Address Bit 1 or Data Bit 1 |
| RD2/AD2 | PORTD | 2 | Address Bit 2 or Data Bit 2 |
| RD3/AD3 | PORTD | 3 | Address Bit 3 or Data Bit 3 |
| RD4/AD4 | PORTD | 4 | Address Bit 4 or Data Bit 4 |
| RD5/AD5 | PORTD | 5 | Address Bit 5 or Data Bit 5 |
| RD6/AD6 | PORTD | 6 | Address Bit 6 or Data Bit 6 |
| RD7/AD7 | PORTD | 7 | Address Bit 7 or Data Bit 7 |
| RE0/AD8 | PORTE | 0 | Address Bit 8 or Data Bit 8 |
| RE1/AD9 | PORTE | 1 | Address Bit 9 or Data Bit 9 |
| RE2/AD10 | PORTE | 2 | Address Bit 10 or Data Bit 10 |
| RE3/AD11 | PORTE | 3 | Address Bit 11 or Data Bit 11 |
| RE4/AD12 | PORTE | 4 | Address Bit 12 or Data Bit 12 |
| RE5/AD13 | PORTE | 5 | Address Bit 13 or Data Bit 13 |
| RE6/AD14 | PORTE | 6 | Address Bit 14 or Data Bit 14 |
| RE7/AD15 | PORTE | 7 | Address Bit 15 or Data Bit 15 |
| RH0/A16 | PORTH | 0 | Address Bit 16 |
| RH1/A17 | PORTH | 1 | Address Bit 17 |
| RH2/A18 | PORTH | 2 | Address Bit 18 |
| RH3/A19 | PORTH | 3 | Address Bit 19 |
| RJ0/ALE | PORTJ | 0 | Address Latch Enable (ALE) Control Pin |
| RJ1/$\overline{OE}$ | PORTJ | 1 | Output Enable ($\overline{OE}$) Control Pin |
| RJ2/$\overline{WRL}$ | PORTJ | 2 | Write Low ($\overline{WRL}$) Control Pin |
| RJ3/$\overline{WRH}$ | PORTJ | 3 | Write High ($\overline{WRH}$) Control Pin |
| RJ4/BA0 | PORTJ | 4 | Byte Address Bit 0 (BA0) |
| RJ5/$\overline{CE}$ | PORTJ | 5 | Chip Enable ($\overline{CE}$) Control Pin |
| RJ6/$\overline{LB}$ | PORTJ | 6 | Lower Byte Enable ($\overline{LB}$) Control Pin |
| RJ7/$\overline{UB}$ | PORTJ | 7 | Upper Byte Enable ($\overline{UB}$) Control Pin |

**Note:** For the sake of clarity, only I/O port and external bus assignments are shown here. One or more additional multiplexed features may be available on some pins.

**REGISTER 10-19: IPR4: PERIPHERAL INTERRUPT PRIORITY REGISTER 4**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| CCP10IP | CCP9IP | CCP8IP | CCP7IP | CCP6IP | CCP5IP | CCP4IP | ECCP3IP |
| 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 **CCP10IP:** CCP10 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **CCP9IP:** CCP9 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **CCP8IP:** CCP8 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **CCP7IP:** CCP7 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CCP6IP:** CCP6 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP5IP:** CCP5 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **CCP4IP:** CCP4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **ECCP3IP:** ECCP3 Interrupt Priority bits
1 = High priority
0 = Low priority

## 11.4 PORTC, LATC and TRISC Registers

PORTC is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISC and LATC. Only PORTC pins, RC2 through RC7, are digital only pins. The pins have Schmitt Trigger input buffers.

When enabling peripheral functions, use care in defining TRIS bits for each PORTC pin. Some peripherals can override the TRIS bit to make a pin an output or input. Consult the corresponding peripheral section for the correct TRIS bit settings.

**Note:** These pins are configured as digital inputs on any device Reset.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

**EXAMPLE 11-3: INITIALIZING PORTC**

```
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

**TABLE 11-3: PORTC FUNCTIONS**

| Pin Name | Function | TRIS Setting | I/O | I/O Type | Description |
|---|---|---|---|---|---|
| RC0/ PWRLCLK/ SCLKI/SOSCO | RC0 | 1 | I | ST | PORTC<0> data input. |
| | PWRLCLK | 1 | I | ST | Optional RTCC input from power line clock (50 or 60 Hz). |
| | SCLKI | x | I | ST | Digital SOSC input. |
| | SOSCO | x | O | ANA | Secondary Oscillator (SOSC) feedback output connection. |
| RC1/SOSCI | RC1 | 1 | I | ST | PORTC<1> data input. |
| | SOSCI | x | I | ANA | Secondary Oscillator (SOSC) input connection. |
| RC2/CTED7/ RP11/AN9/ SEG13 | RC2 | 0 | O | DIG | LATC<2> data output; not affected by analog input. |
| | | 1 | I | ST | PORTC<2> data input; disabled when analog input is enabled. |
| | CTED7 | 1 | I | ST | CTMU Edge 7 input. |
| | RP11 | x | x | DIG | Reconfigurable Pin 11 for PPS-Lite; TRIS must be set to match input/output of module. |
| | AN9 | 1 | I | ANA | A/D Input Channel 9. Default input configuration on POR; does not affect digital output. |
| | SEG13 | 0 | O | ANA | LCD Segment 13 output; disables all other pin functions. |
| RC3/CTED8/ RP15/SCL1/ SEG17 | RC3 | 0 | O | DIG | LATC<3> data output. |
| | | 1 | I | ST | PORTC<3> data input. |
| | CTED8 | 1 | I | ST | CTMU Edge 8 input. |
| | RP15 | x | x | DIG | Reconfigurable Pin 15 for PPS-Lite; TRIS must be set to match input/output of module. |
| | SCL1 | x | I/O | $I^2C$ | Synchronous serial clock input/output for $I^2C$ mode. |
| | SEG17 | 0 | O | ANA | LCD Segment 17 output; disables all other pin functions |
| RC4/CTED9/ RP17/SDA1/ SEG16 | RC4 | 0 | O | DIG | LATC<4> data output. |
| | | 1 | I | ST | PORTC<4> data input. |
| | CTED9 | 1 | I | ST | CTMU Edge 9 input. |
| | RP17 | x | x | DIG | Reconfigurable Pin 17 for PPS-Lite; TRIS must be set to match input/output of module. |
| | SDA1 | x | I/O | $I^2C$ | $I^2C$ mode data I/O |
| | SEG16 | 0 | O | ANA | LCD Segment 16 output; disables all other pin functions. |

**Legend:** O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, $I^2C$ = $I^2C$/SMBus, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

**TABLE 11-6: PORTF FUNCTIONS (CONTINUED)**

| Pin Name | Function | TRIS Setting | I/O | I/O Type | Description |
|---|---|---|---|---|---|
| RF6/RP40/C1INA/ AN11/SEG24 | RF6 | 0 | O | DIG | LATF<6> data output. |
| | | 1 | I | ST | PORTF<6> data input. |
| | RP40 | x | x | DIG | Reconfigurable Pin 40 for PPS-Lite; TRIS must be set to match input/ output of module. |
| | C1INA | 1 | I | ANA | Comparator 1 Input A. |
| | AN11 | 1 | I | ANA | A/D Input Channel 11. Default input configuration on POR; does not affect digital output. |
| | SEG24 | 0 | O | ANA | LCD Segment 24 output; disables all other pin functions. |
| RF7/RP38/AN5/ SEG25 | RF7 | 0 | O | DIG | LATF<7> data output. |
| | | 1 | I | ST | PORTF<7> data input. |
| | RP38 | x | x | DIG | Reconfigurable Pin 38 for PPS-Lite; TRIS must be set to match input/ output of module. |
| | AN5 | 1 | I | ANA | A/D Input Channel 5. Default input configuration on POR; does not affect digital output. |
| | SEG25 | 0 | O | ANA | LCD Segment 25 output; disables all other pin functions. |

**Legend:** O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, XCVR = USB Transceiver, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

**TABLE 11-8: PORTH FUNCTIONS (CONTINUED)**

| Pin Name | Function | TRIS Setting | I/O | I/O Type | Description |
|---|---|---|---|---|---|
| RH5/C2IND/ AN13/SEG41 | RH5 | 0 | O | DIG | LATH<5> data output; not affected by analog input. |
| | | 1 | I | ST | PORTH<5> data input; disabled when analog input is enabled. |
| | C2IND | 1 | I | ANA | Comparator 2 Input D. |
| | AN13 | 1 | I | ANA | A/D Input Channel 13. Default input configuration on POR; does not affect digital output. |
| | SEG41 | 0 | O | ANA | LCD Segment 41 output; disables all other pin functions. |
| RH6/C1INC/ AN14/SEG42 | RH6 | 0 | O | DIG | LATH<6> data output; not affected by analog input. |
| | | 1 | I | ST | PORTH<6> data input; disabled when analog input is enabled. |
| | C1INC | 1 | I | ANA | Comparator 1 Input C. |
| | AN14 | 1 | I | ANA | A/D Input Channel 14. Default input configuration on POR; does not affect digital output. |
| | SEG42 | 0 | O | ANA | LCD Segment 42 output; disables all other pin functions. |
| RH7/AN15/ SEG43 | RH7 | 0 | O | DIG | LATH<7> data output; not affected by analog input. |
| | | 1 | I | ST | PORTH<7> data input; disabled when analog input is enabled. |
| | AN15 | 1 | I | ANA | A/D Input Channel 15. Default input configuration on POR; does not affect digital output. |
| | SEG43 | 0 | O | ANA | LCD Segment 43 output; disables all other pin functions. |

**Legend:** O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

## 12.0 DATA SIGNAL MODULATOR

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module, either internally from the output of a peripheral, or externally through an input pin.

The carrier signal is comprised of two distinct and separate signals: a Carrier High (CARH) signal and a Carrier Low (CARL) signal. During the time in which the Modulator (MOD) signal is in a logic high state, the DSM mixes the Carrier High signal with the Modulator signal. When the Modulator signal is in a logic low state, the DSM mixes the Carrier Low signal with the Modulator signal.

Using this method, the DSM can generate the following types of key modulation schemes:

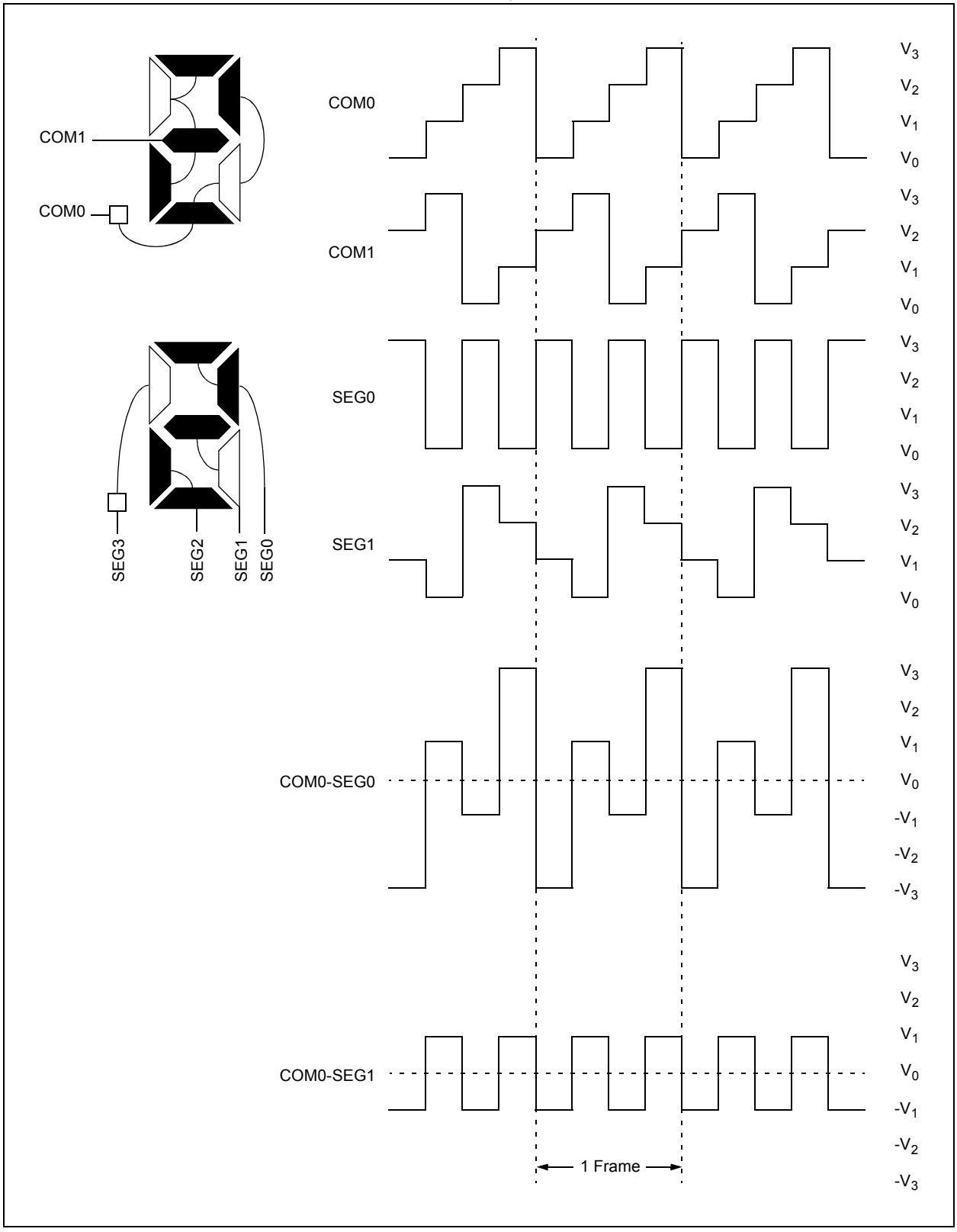
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Carrier Source Pin Disable
- Programmable Modulator Data
- Modulator Source Pin Disable
- Modulator Output Polarity Select
- Slew Rate Control

Figure 12-1 shows a simplified block diagram of the Data Signal Modulator peripheral.

**FIGURE 13-12: TYPE-A WAVEFORMS IN 1/2 MUX, 1/3 BIAS DRIVE**



**REGISTER 19-2: CCPTMRS1: CCP TIMER SELECT REGISTER 1**

| R/W-0 | R/W-0 | U-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| C7TSEL1 | C7TSEL0 | — | C6TSEL0 | — | C5TSEL0 | C4TSEL1 | C4TSEL0 |
| 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 7-6 **C7TSEL<1:0>:** CCP7 Timer Selection bits
00 =CCP7 is based off of TMR1/TMR2
01 =CCP7 is based off of TMR5/TMR4
10 =CCP7 is based off of TMR5/TMR6
11 =CCP7 is based off of TMR5/TMR8

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **C6TSEL0:** CCP6 Timer Selection bit
0 = CCP6 is based off of TMR1/TMR2
1 = CCP6 is based off of TMR5/TMR2

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **C5TSEL0:** CCP5 Timer Selection bit
0 = CCP5 is based off of TMR1/TMR2
1 = CCP5 is based off of TMR5/TMR4

bit 1-0 **C4TSEL<1:0>:** CCP4 Timer Selection bits
00 =CCP4 is based off of TMR1/TMR2
01 =CCP4 is based off of TMR3/TMR4
10 =CCP4 is based off of TMR3/TMR6
11 =Reserved; do not use

### 20.3.9 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full-power mode. In the case of Sleep mode, all clocks are halted.

In Idle modes, a clock is provided to the peripherals. That clock can be from the primary clock source, the secondary clock (SOSC Oscillator) or the INTOSC source.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSPx interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSPx interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSPx interrupt flag bit will be set, and if enabled, will wake the device.

### 20.3.10 EFFECTS OF A RESET

A Reset disables the MSSPx module and terminates the current transfer.

### 20.3.11 BUS MODE COMPATIBILITY

Table 20-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 20-1: SPI BUS MODES**

| Standard SPI Mode Terminology | Control Bits State | |
|---|---|---|
| | CKP | CKE |
| 0, 0 | 0 | 1 |
| 0, 1 | 0 | 0 |
| 1, 0 | 1 | 1 |
| 1, 1 | 1 | 0 |

There is also an SMP bit which controls when the data is sampled.

### 20.3.12 SPI CLOCK SPEED AND MODULE INTERACTIONS

Because MSSP1 and MSSP2 are independent modules, they can operate simultaneously at different data rates. Setting the SSPM<3:0> bits of the SSPx-CON1 register determines the rate for the corresponding module.

An exception is when both modules use Timer2 as a time base in Master mode. In this instance, any changes to the Timer2 module's operation will affect both MSSPx modules equally. If different bit rates are required for each module, the user should select one of the other three time base options for one of the modules.

**REGISTER 20-6: SSPxSTAT: MSSPx STATUS REGISTER ($I^2C$ MODE)**

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P(1) | S(1) | R/$\overline{W}$(2,3) | UA | BF |
| 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 **SMP:** Slew Rate Control bit

In Master or Slave mode:
1 = Slew rate control is disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control is enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit

In Master or Slave mode:
1 = Enables SMBus-specific inputs
0 = Disables SMBus-specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit

In Master mode:
Reserved.

In Slave mode:
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit(1)

1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last

bit 3 **S:** Start bit(1)

1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit(2,3)

In Slave mode:
1 = Read
0 = Write

In Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress

bit 1 **UA:** Update Address bit (10-Bit Slave mode only)

1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:
1 = SSPxBUF is full
0 = SSPxBUF is empty

In Receive mode:
1 = SSPxBUF is full (does not include the $\overline{\text{ACK}}$ and Stop bits)
0 = SSPxBUF is empty (does not include the $\overline{\text{ACK}}$ and Stop bits)

**Note 1:** This bit is cleared on Reset and when SSPEN is cleared.

**2:** This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{\text{ACK}}$ bit.

**3:** ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Active mode.

## 22.3 A/D Module Configuration

All of the registers described in the previous section must be configured for module operation to be fully defined. An effective approach is first, to describe the signals and sequences for the particular application. Typically, it is an iterative process to assign signals to port pins, to establish timing methods and to organize a scanning scheme, as well as to integrate the whole process with the software design.

The various configuration and control functions of the module are distributed throughout the module's six control registers. Control functions can be broadly sorted into four groups: input, timing, conversion and output. Table 22-1 shows the register location of control or Status bits by register.

**TABLE 22-1: A/D MODULE FUNCTIONS BY REGISTERS AND BITS**

| A/D Function | Register(s) | Specific Bits |
|---|---|---|
| Input | ADCON2H/L | PVCFG<1:0>, NVCFG, OFFCAL, CSCNA, ALTS |
| | ADCHS0H/L | CH0NB<2:0>, CH0SB<4:0>, CH0NA<2:0>, CH0SA<4:0> |
| | ADCSS0H/L | CSS<15:0> |
| | ADCSS1H/L | CSS<30:16> |
| | ADCTMEN0H/L | CTMEN<15:0> |
| | ADCTMEN1H/L | CTMEN<31:16> |
| Conversion | ADCON1H/L | ADON, SSRC<3:0>, ASAM, SAMP, DONE, MODE12 |
| | ADCON2H/L | SMPI<4:0> |
| | ADCON3H/L | EXTSAM |
| | ADCON5H/L | ASEN, LPENA, ASINTMD<1:0> |
| Timing | ADCON3H/L | ADRC, SAMC<4:0>, ADCS<7:0> |
| Output | ADCON1H/L | FORM<1:0> |
| | ADCON2H/L | BUFS, BUFM, BUFREGEN |
| | ADCON5H/L | WM<1:0>, CM<1:0> |

**Note:** Do not write to the SSRCx, BUFS, SMPIx, BUFM and ALTS bits, or the ADCON3H/L, and ADCSS0H/L registers, while ADON = 1; otherwise, indeterminate conversion data may result.

The following steps should be followed for performing an A/D conversion:

1. Configure the A/D module:
   - Select the output resolution (if configurable)
   - Select the voltage reference source to match the expected range on analog inputs
   - Select the analog conversion clock to match the desired data rate with a processor clock
   - Determine how sampling will occur
   - Set the multiplexer input assignments
   - Select the desired sample/conversion sequence
   - Select the output data format
   - Select the output value destination
   - Select the number of readings per interrupt
2. Configure the A/D interrupt (if required):
   - Clear the ADIF bit
   - Select the A/D interrupt priority
3. Turn on the A/D module.

The options for each configuration step are described in the subsequent sections.

### 22.3.1 SELECTING THE RESOLUTION

The MODE12 bit (ADCON1H<3>) controls output resolution. Setting this bit selects 12-bit resolution.

### 22.3.2 SELECTING THE VOLTAGE REFERENCE SOURCE

The voltage references for A/D conversions are selected using the PVCFG<1:0> and NVCFG0 control bits (ADCON2H<7:5>). The upper voltage reference (VR+) may be $AV_{DD}$, the external $V_{REF}+$ or an internal band gap reference voltage. The lower voltage reference (VR-) may be $AV_{SS}$ or the $V_{REF}-$ input pin. The available options vary between device families.

The external voltage reference pins may be shared with the AN2 and AN3 inputs on low pin count devices. The A/D Converter can still perform conversions on these pins when they are shared with the $V_{REF}+$ and $V_{REF}-$ input pins.

### 22.7.4 THRESHOLD DETECT INTERRUPTS

The A/D module can generate an interrupt and set the ADIF flag based on Threshold Detect operation. This is based on completion of a Threshold Detect sequence and/or the occurrence of a valid comparison. When Threshold Detect is enabled (ASENA = 1), A/D module interrupt generation is governed by the ASINTMDx bits (ADCON5H<1:0>), superseding any configuration implemented by the SMPIx bits (ADCON2L<6:2>). For information on alternative interrupt settings, refer to **Section 22.6.1 "Number of Conversions Per Interrupt"**.

The Threshold Detect interrupt is configured by the ASINTMD<1:0> bits (ADCON5H<1:0>). Options include interrupt after a scan sequence, interrupt after a scan sequence with a valid match, interrupt after a valid match (without waiting for the sequence to end) or no interrupt.

### 22.7.5 COMPARISON MODE EXAMPLES

The following examples show the effect of valid comparisons on the results buffer and the

Compare Hit registers. In each figure, changes within the registers are indicated in bold.

For the sake of simplicity, the examples assume a device with only 16 analog inputs. Devices with a greater number of channels, and thus, larger results buffers and two Compare Hit registers, will function in a similar fashion.

| | |
|---|---|
| **Note:** | When using any comparison mode, always use channel indexed buffer storage (BUFREGEN = 1). Otherwise, the threshold values for other channels may be overwritten, resulting in unpredictable comparisons. |

#### 22.7.5.1 Simple Comparisons (Greater And Less Than Results)

When the Compare Mode bits, CM<1:0> (ADCON5L<1:0>), are programmed as '0x', the converter compares the sampled value to see if it is greater than (CM<1:0> = 01), or less than (CM<1:0> = 00), the threshold value in the buffer location. If the condition is met, both of the following occur:

- The Compare Hit bit (CHHn) for the corresponding channel is set.
- If the Write Mode bits, WM<1:0> (ADCON5L<3:2>), are programmed to '01', the converted value is written to the buffer, replacing the threshold value. If WM<1:0> = 10, the converted value is discarded.

The changes to the result buffer and the Compare Hit register are shown in Figure 22-13. Note that they are the same for both types of simple comparison.

**REGISTER 23-1: CMxCON: COMPARATOR CONTROL x REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| CON | COE | CPOL | EVPOL1 | EVPOL0 | CREF | CCH1 | CCH0 |
| 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 **CON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled

bit 6 **COE:** Comparator Output Enable bit
1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only

bit 5 **CPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 4-3 **EVPOL<1:0>:** Interrupt Polarity Select bits
11 = Interrupt generation on any change of the output(1)
10 = Interrupt generation only on high-to-low transition of the output
01 = Interrupt generation only on low-to-high transition of the output
00 = Interrupt generation is disabled

bit 2 **CREF:** Comparator Reference Select bit (non-inverting input)
1 = Non-inverting input connects to internal $CV_{REF}$ voltage
0 = Non-inverting input connects to CxINA pin

bit 1-0 **CCH<1:0>:** Comparator Channel Select bits
11 = Inverting input of comparator connects to $V_{BG}$
10 = Inverting input of comparator connects to C2INB pin
01 = Inverting input of comparator connects to CxINC pin
00 = Inverting input of comparator connects to CxINx pin(2)

**Note 1:** The CMPxIF is automatically set any time this mode is selected and must be cleared by the application after the initial configuration.

**2:** Comparator 1 and Comparator 3 use C2INB as an input to the inverting terminal. Comparator 2 uses C2IND as an input to the inverting terminal.

The USB Specification limits the power taken from the bus. Each device is ensured 100 mA at approximately 5V (one unit load). Additional power may be requested, up to a maximum of 500 mA.

Note that power above one unit load is a request and the host or hub is not obligated to provide the extra current. Thus, a device capable of consuming more than one unit load must be able to maintain a low-power configuration of a one unit load or less, if necessary.

The USB Specification also defines a Suspend mode. In this situation, current must be limited to 500 μA, averaged over one second. A device must enter a suspend state after 3 ms of inactivity (i.e., no SOF tokens for 3 ms). A device entering Suspend mode must drop current consumption within 10 ms after suspend. Likewise, when signaling a wake-up, the device must signal a wake-up within 10 ms of drawing current above the suspend limit.

### 27.9.5 ENUMERATION

When the device is initially attached to the bus, the host enters an enumeration process in an attempt to identify the device. Essentially, the host interrogates the device, gathering information, such as power consumption, data rates and sizes, protocol and other descriptive information; descriptors contain this information. A typical enumeration process would be as follows:

1. USB Reset – Reset the device. Thus, the device is not configured and does not have an address (Address 0).
2. Get Device Descriptor – The host requests a small portion of the device descriptor.
3. USB Reset – Reset the device again.
4. Set Address – The host assigns an address to the device.
5. Get Device Descriptor – The host retrieves the device descriptor, gathering info, such as manufacturer, type of device, maximum control packet size.
6. Get configuration descriptors.
7. Get any other descriptors.
8. Set a configuration.

The exact enumeration process depends on the host.

### 27.9.6 DESCRIPTORS

There are eight different standard descriptor types, of which, five are most important for this device.

#### 27.9.6.1 Device Descriptor

The device descriptor provides general information, such as manufacturer, product number, serial number, the class of the device and the number of configurations. There is only one device descriptor.

#### 27.9.6.2 Configuration Descriptor

The configuration descriptor provides information on the power requirements of the device and how many different interfaces are supported when in this configuration. There may be more than one configuration for a device (i.e., low-power and high-power configurations).

#### 27.9.6.3 Interface Descriptor

The interface descriptor details the number of endpoints used in this interface, as well as the class of the interface. There may be more than one interface for a configuration.

#### 27.9.6.4 Endpoint Descriptor

The endpoint descriptor identifies the transfer type (**Section 27.9.3 “Transfers”**) and direction, and some other specifics for the endpoint. There may be many endpoints in a device and endpoints may be shared in different configurations.

#### 27.9.6.5 String Descriptor

Many of the previous descriptors reference one or more string descriptors. String descriptors provide human readable information about the layer (**Section 27.9.1 “Layered Framework”**) they describe. Often these strings show up in the host to help the user identify the device. String descriptors are generally optional to save memory and are encoded in a unicode format.

### 27.9.7 BUS SPEED

Each USB device must indicate its bus presence and speed to the host. This is accomplished through a 1.5 kΩ resistor, which is connected to the bus at the time of the attachment event.

Depending on the speed of the device, the resistor either pulls up the D+ or D- line to 3.3V. For a low-speed device, the pull-up resistor is connected to the D- line. For a full-speed device, the pull-up resistor is connected to the D+ line.

### 27.9.8 CLASS SPECIFICATIONS AND DRIVERS

USB specifications include class specifications, which operating system vendors optionally support. Examples of classes include: Audio, Mass Storage, Communications and Human Interface (HID). In most cases, a driver is required at the host side to ‘talk’ to the USB device. In custom applications, a driver may need to be developed. Fortunately, drivers are available for most common host systems for the most common classes of devices. Thus, these drivers can be reused.

### 28.2.2 CONTROL REGISTER

Register 28-17 shows the RCON2 register. This is a readable and writable register which contains a control bit that allows software to override the WDT Enable Configuration bit, but only if the Configuration bit has disabled the WDT.

**REGISTER 28-17: RCON2: RESET CONTROL REGISTER 2**

| R/W, HS-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| EXTR(1) | — | SWDTEN(2) | — | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| HS = Hardware Settable bit | | | |
| 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 **EXTR:** External Reset ($\overline{MCLR}$) Pin bit(1)
- 1 = A Master Clear (pin) Reset has occurred
- 0 = A Master Clear (pin) Reset has not occurred

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

bit 5 **SWDTEN:** Software Controlled Watchdog Timer Enable bit(2)
- 1 = Watchdog Timer is on
- 0 = Watchdog Timer is off

bit 4-0 **Unimplemented**: Read as '0'

**Note 1:** This bit is set in hardware; it can be cleared in software.

**2:** This bit has no effect unless the Configuration bits, WDTEN<1:0> = 10.

**BTG** **Bit Toggle f**

Syntax: BTG f, b {,a}

Operands: $0 \le f \le 255$
$0 \le b < 7$
$a \in [0,1]$

Operation: $(\overline{f<b>}) \rightarrow f<b>$

Status Affected: None

Encoding:

| 0111 | bbba | ffff | ffff |
|---|---|---|---|

Description: Bit ‘b’ in data memory location, ‘f’, is inverted.

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 $f \le 95$ (5Fh). See **Section 29.2.3 “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: `BTG PORTC, 4, 0`

Before Instruction:
PORTC = 0111 0101 [75h]
After Instruction:
PORTC = 0110 0101 [65h]

**BOV** **Branch if Overflow**

Syntax: BOV n

Operands: $-128 \le n \le 127$

Operation: if Overflow bit is ‘1’,
$(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| 1110 | 0100 | nnnn | nnnn |
|---|---|---|---|

Description: If the Overflow bit is ‘1’, then the program will branch.

The 2’s complement number ‘2n’ is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal ‘n’ | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal ‘n’ | Process Data | No operation |

Example:
```
HERE    BOV  Jump
```
Before Instruction
PC = address (HERE)
After Instruction
If Overflow = 1;
PC = address (Jump)
If Overflow = 0;
PC = address (HERE + 2)

**CPFSGT** **Compare f with W, Skip if f > W**

Syntax: CPFSGT f {,a}

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

Operation: (f) – (W),
skip if (f) > (W)
(unsigned comparison)

Status Affected: None

Encoding:

| 0110 | 010a | ffff | ffff |
|---|---|---|---|

Description: Compares the contents of data memory location 'f' to the contents of the W by performing an unsigned subtraction.

If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

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 $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
**Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE      CPFSGT REG, 0
NGREATER  :
GREATER   :
```
Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG > W;
PC = Address (GREATER)
If REG ≤ W;
PC = Address (NGREATER)

**CPFSLT** **Compare f with W, Skip if f < W**

Syntax: CPFSLT f {,a}

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

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

| 0110 | 000a | ffff | ffff |
|---|---|---|---|

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.

If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

Words: 1

Cycles: 1(2)
**Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | No operation |

If skip:

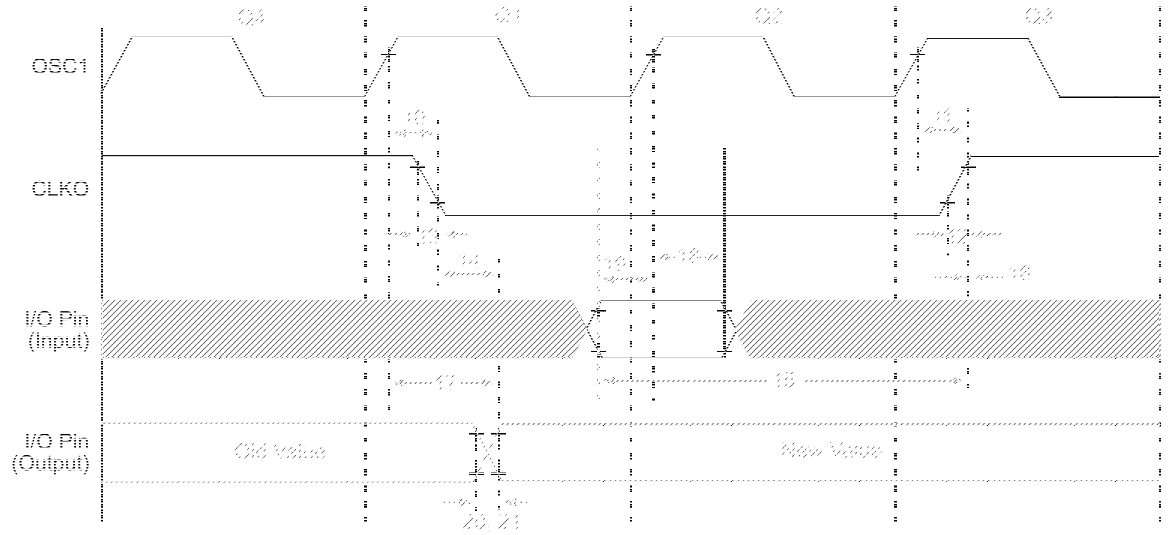
| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```
Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG < W;
PC = Address (LESS)
If REG ≥ W;
PC = Address (NLESS)

**FIGURE 30-4: CLKO AND I/O TIMING**




**Note:** Refer to Figure 30-2 for load conditions.

**TABLE 30-23: CLKO AND I/O TIMING REQUIREMENTS**

| Param No. | Symbol | Characteristic | Min. | Typ. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 10 | TOSH2CKL | OSC1 ↑ to CLKO ↓ | — | 75 | 200 | ns | **(Note 1)** |
| 11 | TOSH2CKH | OSC1 ↑ to CLKO ↑ | — | 75 | 200 | ns | **(Note 1)** |
| 12 | TCKR | CLKO Rise Time | — | 15 | 30 | ns | **(Note 1)** |
| 13 | TCKF | CLKO Fall Time | — | 15 | 30 | ns | **(Note 1)** |
| 14 | TCKL2IOV | CLKO ↓ to Port Out Valid | — | — | 0.5 TCY + 20 | ns | |
| 15 | TIOV2CKH | Port In Valid before CLKO ↑ | 0.25 TCY + 25 | — | — | ns | |
| 16 | TCKH2IOI | Port In Hold after CLKO ↑ | 0 | — | — | ns | |
| 17 | TOSH2IOV | OSC1 ↑ (Q1 cycle) to Port Out Valid | — | 50 | 150 | ns | |
| 18 | TOSH2IOI | OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time) | 100 | — | — | ns | |
| 19 | TIOV2OSH | Port Input Valid to OSC1 ↑ (I/O in setup time) | 0 | — | — | ns | |
| 20 | TIOR | Port Output Rise Time | — | 10 | 25 | ns | |
| 21 | TIOF | Port Output Fall Time | — | 10 | 25 | ns | |
| 22† | TINP | INTx Pin High or Low Time | 20 | — | — | ns | |
| 23† | TRBP | RB<7:4> Change INTx High or Low Time | TCY | — | — | ns | |

† These parameters are asynchronous events not related to any internal clock edges.

**Note 1:** Measurements are taken in EC mode, where CLKO output is 4 x TOSC.