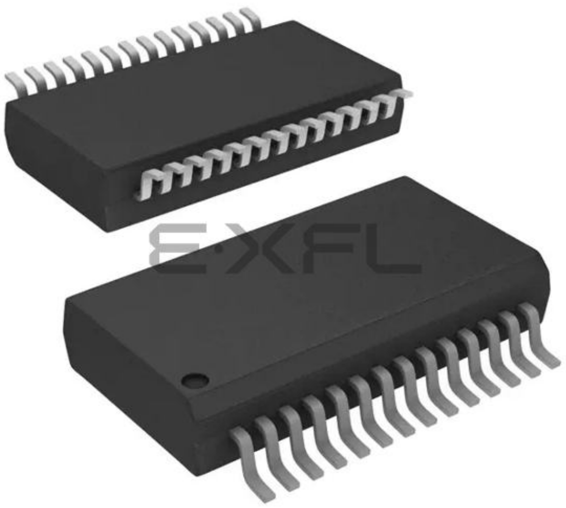


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	23
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5V
Data Converters	A/D 19x10b/12b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic24fv08km202-e-ss

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

3.2 CPU Control Registers

REGISTER 3-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HSC
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0, HSC(1)	R/W-0, HSC(1)	R/W-0, HSC(1)	R-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
IPL2(2)	IPL1(2)	IPL0(2)	RA	N	OV	Z	C
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable 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 15-9 **Unimplemented:** Read as ‘0’

bit 8 **DC:** ALU Half Carry/Borrow bit
1 = A carry-out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred
0 = No carry-out from the 4th or 8th low-order bit of the result has occurred

bit 7-5 **IPL<2:0>:** CPU Interrupt Priority Level Status bits(1,2)
111 = CPU Interrupt Priority Level is 7 (15); user interrupts are disabled
110 = CPU Interrupt Priority Level is 6 (14)
101 = CPU Interrupt Priority Level is 5 (13)
100 = CPU Interrupt Priority Level is 4 (12)
011 = CPU Interrupt Priority Level is 3 (11)
010 = CPU Interrupt Priority Level is 2 (10)
001 = CPU Interrupt Priority Level is 1 (9)
000 = CPU Interrupt Priority Level is 0 (8)

bit 4 **RA:** REPEAT Loop Active bit
1 = REPEAT loop in progress
0 = REPEAT loop not in progress

bit 3 **N:** ALU Negative bit
1 = Result was negative
0 = Result was non-negative (zero or positive)

bit 2 **OV:** ALU Overflow bit
1 = Overflow occurred for signed (2’s complement) arithmetic in this arithmetic operation
0 = No overflow has occurred

bit 1 **Z:** ALU Zero bit
1 = An operation, which effects the Z bit, has set it at some time in the past
0 = The most recent operation, which effects the Z bit, has cleared it (i.e., a non-zero result)

bit 0 **C:** ALU Carry/$\overline{\text{Borrow}}$ bit
1 = A carry-out from the Most Significant bit (MSb) of the result occurred
0 = No carry-out from the Most Significant bit (MSb) of the result occurred

Note 1: The IPLx Status bits are read-only when NSTDIS (INTCON1<15>) = 1.
2: The IPL<2:0> Status bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

4.3.2 DATA ACCESS FROM PROGRAM MEMORY AND DATA EEPROM MEMORY USING TABLE INSTRUCTIONS

The `TBLRDL` and `TBLWTL` instructions offer a direct method of reading or writing the lower word of any address within the program memory without going through Data Space. It also offers a direct method of reading or writing a word of any address within data EEPROM memory. The `TBLRDH` and `TBLWTH` instructions are the only method to read or write the upper 8 bits of a program space word as data.

> **Note:** The `TBLRDH` and `TBLWTH` instructions are not used while accessing data EEPROM memory.

The PC is incremented by 2 for each successive 24-bit program word. This allows program memory addresses to directly map to Data Space addresses. Program memory can thus be regarded as two 16-bit, word-wide address spaces, residing side by side, each with the same address range. `TBLRDL` and `TBLWTL` access the space which contains the least significant data word, and `TBLRDH` and `TBLWTH` access the space which contains the upper data byte.

Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

1. `TBLRDL` (Table Read Low): In Word mode, it maps the lower word of the program space location (P<15:0>) to a data address (D<15:0>).

   In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when byte select is '`1`'; the lower byte is selected when it is '`0`'.

2. `TBLRDH` (Table Read High): In Word mode, it maps the entire upper word of a program address (P<23:16>) to a data address. Note that D<15:8>, the 'phantom' byte, will always be '`0`'.

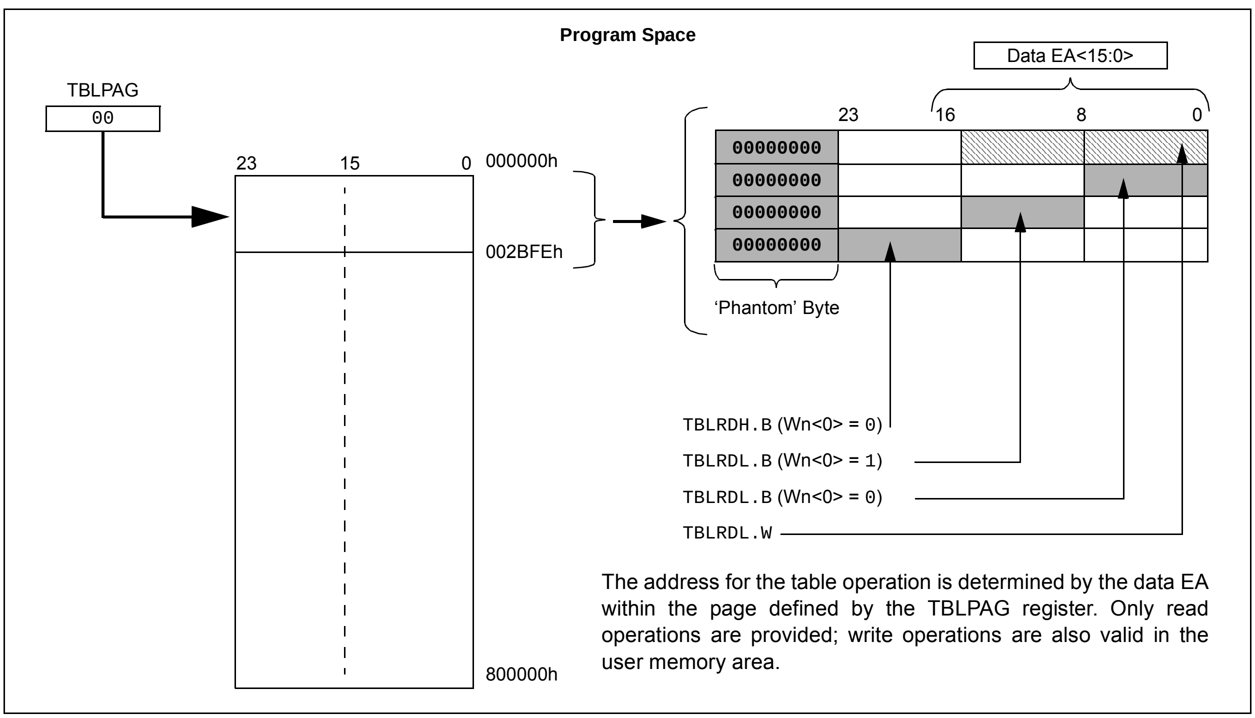
   In Byte mode, it maps the upper or lower byte of the program word to D<7:0> of the data address, as above. Note that the data will always be '`0`' when the upper 'phantom' byte is selected (Byte Select = `1`).

In a similar fashion, two table instructions, `TBLWTH` and `TBLWTL`, are used to write individual bytes or words to a program space address. The details of their operation are explained in **Section 5.0 "Flash Program Memory"**.

For all table operations, the area of program memory space to be accessed is determined by the Table Memory Page Address register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user and configuration spaces. When TBLPAG<7> = `0`, the table page is located in the user memory space. When TBLPAG<7> = `1`, the page is located in configuration space.

> **Note:** Only Table Read operations will execute in the configuration memory space, and only then, in implemented areas, such as the Device ID. Table Write operations are not allowed.

FIGURE 4-6: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



REGISTER 8-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	U-0	U-0
U2TXIF	U2RXIF	INT2IF	CCT4IF	CCT3IF	—	—	—
bit 15							bit 8

U-0	R/W-0, HS	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
—	CCP5IF	—	INT1IF	CNIF	CMIF	BCL1IF	SSP1IF
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 15 **U2TXIF:** UART2 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **U2RXIF:** UART2 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **INT2IF:** External Interrupt 2 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **CCT4IF:** Capture/Compare 4 Timer Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **CCT3IF:** Capture/Compare 3 Timer Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10-7 **Unimplemented:** Read as ‘0’

bit 6 **CCP5IF:** Capture/Compare 5 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **INT1IF:** External Interrupt 1 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **CNIF:** Input Change Notification Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **CMIF:** Comparator Interrupt Flag Status Bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **BCL1IF:** MSSP1 I^2C™ Bus Collision Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SI2C1IF:** MSSP1 SPI/I^2C Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 8-30: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
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 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **U2ERIP<2:0>:** UART2 Error Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **U1ERIP<2:0>:** UART1 Error Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 9-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5(1)	TUN4(1)	TUN3(1)	TUN2(1)	TUN1(1)	TUN0(1)
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 15-6 **Unimplemented:** Read as ‘0’

bit 5-0 **TUN<5:0>:** FRC Oscillator Tuning bits(1)

011111 = Maximum frequency deviation
011110
•
•
•
000001
000000 = Center frequency, oscillator is running at factory calibrated frequency
111111
•
•
•
100001
100000 = Minimum frequency deviation

Note 1: Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

11.2.2 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation, and a read operation of the same port. Typically, this instruction would be a NOP.

11.3 Input Change Notification (ICN)

The Input Change Notification function of the I/O ports allows the PIC24FXXXXX family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 37 external signals (CN0 through CN36) that may be selected (enabled) for generating an interrupt request on a Change-of-State.

There are six control registers associated with the CN module. The CNEN1 and CNEN3 registers contain the interrupt enable control bits for each of the CNx input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CNx pin also has a weak pull-up/pull-down connected to it. The pull-ups act as a current source that is connected to the pin. The pull-downs act as a current sink to eliminate the need for external resistors when push button or keypad devices are connected.

On any pin, only the pull-up resistor or the pull-down resistor should be enabled, but not both of them. If the push button or the keypad is connected to V_{DD}, enable the pull-down, or if they are connected to V_{SS}, enable the pull-up resistors. The pull-ups are enabled separately using the CNPU1 and CNPU3 registers, which contain the control bits for each of the CNx pins.

Setting any of the control bits enables the weak pull-ups for the corresponding pins. The pull-downs are enabled separately using the CNPD1 and CNPD3 registers, which contain the control bits for each of the CNx pins. Setting any of the control bits enables the weak pull-downs for the corresponding pins.

When the internal pull-up is selected, the pin uses V_{DD} as the pull-up source voltage. When the internal pull-down is selected, the pins are pulled down to V_{SS} by an internal resistor. Make sure that there is no external pull-up source/pull-down sink when the internal pull-ups/pull-downs are enabled.

Note:	Pull-ups and pull-downs on Change Notification (CN) pins should always be disabled whenever the port pin is configured as a digital output.

EXAMPLE 11-1: PORT WRITE/READ EXAMPLE

```
MOV   0xFF00, W0;          //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
MOV   W0, TRISB;
NOP;                       //Delay 1 cycle
BTSS  PORTB, #13;          //Next Instruction

Equivalent ‘C’ Code
TRISB = 0xFF00;            //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
NOP();                     //Delay 1 cycle
if(PORTBbits.RB13 == 1)    // execute following code if PORTB pin 13 is set.
{
}
```

REGISTER 14-4: SSPxCON1: MSSPx CONTROL REGISTER 1 (I²C™ MODE)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
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 15-8 **Unimplemented:** Read as '0'

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:
1 = A write to the SSPxBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)
0 = No collision

In Slave Transmit mode:
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

In Receive mode (Master or Slave modes):
This is a "don't care" bit.

bit 6 **SSPOV:** Master Synchronous Serial Port Receive Overflow Indicator bit

In Receive mode:
1 = A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)
0 = No overflow

In Transmit mode:
This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(1)

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins
0 = Disables the serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCLx Release Control bit

In Slave mode:
1 = Releases clock
0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:
Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(2)

1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1011 = I²C Firmware Controlled Master mode (Slave Idle)
1000 = I²C Master mode, Clock = FOSC/(2 * ([SSPxADD] + 1))(3)
0111 = I²C Slave mode, 10-bit address
0110 = I²C Slave mode, 7-bit address

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.
2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.
3: SSPxADD values of 0, 1 or 2 are not supported when the Baud Rate Generator is used with I²C mode.

15.2 Transmitting in 8-Bit Data Mode

1. Set up the UARTx:
   a) Write the appropriate values for data, parity and Stop bits.
   b) Write the appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UARTx.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
4. Write the data byte to the lower byte of the UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with the next rising edge of the baud clock.
5. Alternately, the data byte may be transferred while UTXEN = 0, and then, the user may set UTXEN. This will cause the serial bit stream to begin immediately, because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated as per interrupt control bit, UTXISELx.

15.3 Transmitting in 9-Bit Data Mode

1. Set up the UARTx (as described in **Section 15.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UARTx.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
4. Write UxTXREG as a 16-bit value only.
5. A word write to UxTXREG triggers the transfer of the 9-bit data to the TSR. The serial bit stream will start shifting out with the first rising edge of the baud clock.
6. A transmit interrupt will be generated as per the setting of control bit, UTXISELx.

15.4 Break and Sync Transmit Sequence

The following sequence will send a message frame header, made up of a Break, followed by an Auto-Baud Sync byte.

1. Configure the UARTx for the desired mode.
2. Set UTXEN and UTXBRK – this sets up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write '55h' to UxTXREG – loads the Sync character into the transmit FIFO.
5. After the Break has been sent, the UTXBRK bit is reset by hardware. The Sync character now transmits.

15.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UARTx (as described in **Section 15.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UARTx.
3. A receive interrupt will be generated when one or more data characters have been received, as per interrupt control bit, URXISELx.
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read UxRXREG.

The act of reading the UxRXREG character will move the next character to the top of the receive FIFO, including a new set of PERR and FERR values.

15.6 Operation of $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Control Pins

UARTx Clear-To-Send ($\overline{\text{UxCTS}}$) and Request-To-Send ($\overline{\text{UxRTS}}$) are the two hardware controlled pins that are associated with the UARTx module. These two pins allow the UARTx to operate in Simplex and Flow Control modes. They are implemented to control the transmission and reception between the Data Terminal Equipment (DTE). The UEN<1:0> bits in the UxMODE register configure these pins.

15.7 Infrared Support

The UARTx module provides two types of infrared UARTx support: one is the IrDA clock output to support an external IrDA encoder and decoder device (legacy module support), and the other is the full implementation of the IrDA encoder and decoder.

As the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is '0'.

15.7.1 EXTERNAL IrDA SUPPORT – IrDA CLOCK OUTPUT

To support external IrDA encoder and decoder devices, the UxBCLK pin (same as the $\overline{\text{UxRTS}}$ pin) can be configured to generate the 16x baud clock. When UEN<1:0> = 11, the UxBCLK pin will output the 16x baud clock if the UARTx module is enabled; it can be used to support the IrDA codec chip.

15.7.2 BUILT-IN IrDA ENCODER AND DECODER

The UARTx has full implementation of the IrDA encoder and decoder as part of the UARTx module. The built-in IrDA encoder and decoder functionality is enabled using the IREN bit (UxMODE<12>). When enabled (IREN = 1), the receive pin (UxRX) acts as the input from the infrared receiver. The transmit pin (UxTX) acts as the output to the infrared transmitter.

16.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the Real-Time Clock and Calendar, refer to the *"PIC24F Family Reference Manual"*, **"Real-Time Clock and Calendar (RTCC)"** (DS39696).

The RTCC provides the user with a Real-Time Clock and Calendar (RTCC) function that can be calibrated.

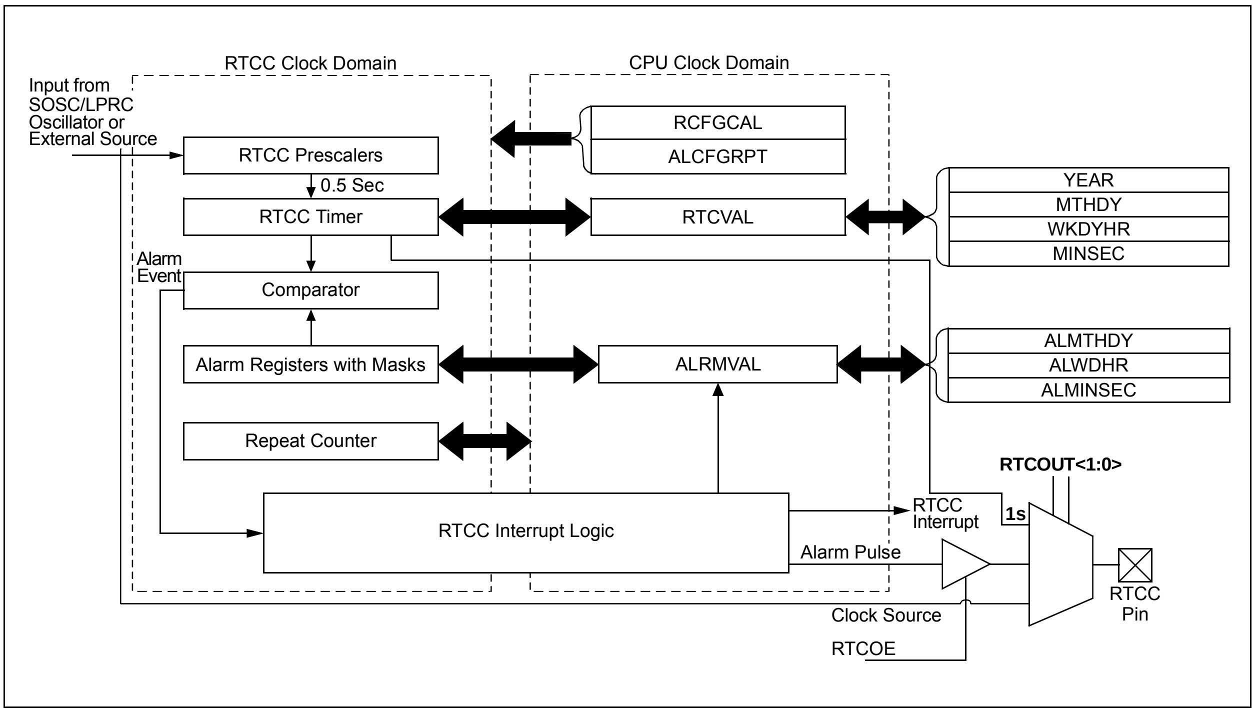
Key features of the RTCC module are:

- Operates in Sleep and Retention Sleep modes
- Selectable clock source
- Provides hours, minutes and seconds using 24-hour format
- Visibility of one half second period
- Provides calendar – weekday, date, month and year
- Alarm-configurable for half a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week, one month or one year
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat chime
- Year 2000 to 2099 leap year correction
- BCD format for smaller software overhead
- Optimized for long term battery operation
- User calibration of the 32.768 kHz clock crystal/32K INTRC frequency with periodic auto-adjust
- Optimized for long term battery operation
- Fractional second synchronization
- Calibration to within ±2.64 seconds error per month
- Calibrates up to 260 ppm of crystal error
- Ability to periodically wake-up external devices without CPU intervention (external power control)
- Power control output for external circuit control
- Calibration takes effect every 15 seconds
- Runs from any one of the following:
  - External Real-Time Clock of 32.768 kHz
  - Internal 31.25 kHz LPRC Clock
  - 50 Hz or 60 Hz External Input

16.1 RTCC Source Clock

The user can select between the SOSC crystal oscillator, LPRC internal oscillator or an external 50 Hz/60 Hz power line input as the clock reference for the RTCC module. This gives the user an option to trade off system cost, accuracy and power consumption, based on the overall system needs.

FIGURE 16-1: RTCC BLOCK DIAGRAM



16.2 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

16.2.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through corresponding register pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTRx bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 16-1).

By writing the RTCVALH byte, the RTCC Pointer value, the RTCPTR<1:0> bits decrement by one until they reach '00'. Once they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 16-1: RTCVAL REGISTER MAPPING

RTCPTR<1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTRx bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 16-2).

By writing the ALRMVALH byte, the ALRMPTR<1:0> bits (Alarm Pointer value) decrement by one until they reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL, until the pointer value is manually changed.

TABLE 16-2: ALRMVAL REGISTER MAPPING

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVALH<15:8>	ALRMVALL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	PWCSTAB	PWCSAMP

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes, the ALRMPTR<1:0> value will be decremented. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

> **Note:** This only applies to read operations and not write operations.

16.2.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RCFGCAL<13>) must be set (see Example 16-1 and Example 16-2).

> **Note:** To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN. Therefore, it is recommended that code follow the procedure in Example 16-2.

16.2.3 SELECTING RTCC CLOCK SOURCE

There are four reference source clock options that can be selected for the RTCC using the RTCCLK<1:0> bits (RTCPWC<11:10>): 00 = Secondary Oscillator, 01 = LPRC, 10 = 50 Hz External Clock and 11 = 60 Hz External Clock.

EXAMPLE 16-1: SETTING THE RTCWREN BIT IN ASSEMBLY

```
push    w7              ; Store W7 and W8 values on the stack.
push    w8
disi    #5              ; Disable interrupts until sequence is complete.
mov     #0x55, w7       ; Write 0x55 unlock value to NVMKEY.
mov     w7, NVMKEY
mov     #0xAA, w8       ; Write 0xAA unlock value to NVMKEY.
mov     w8, NVMKEY
bset    RCFGCAL, #13    ; Set the RTCWREN bit.
pop     w8              ; Restore the original W register values from the stack.
pop     w7
```

EXAMPLE 16-2: SETTING THE RTCWREN BIT IN 'C'

```
//This builtin function executes implements the unlock sequence and sets
//the RTCWREN bit.
__builtin_write_RTCWEN();
```

NOTES:

REGISTER 20-1: DACxCON: DACx CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
DACEN	—	DACSIDL	DACSLP	DACFM	—	SRDIS	DACTRIG
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACOE	DACTSEL4	DACTSEL3	DACTSEL2	DACTSEL1	DACTSEL0	DACREF1	DACREF0
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 15 **DACEN:** DACx Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **DACSIDL:** DACx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **DACSLP:** DACx Enable Peripheral During Sleep bit
1 = DACx continues to output the most recent value of DACxDAT during Sleep mode
0 = DACx is powered down in Sleep mode; DACxOUT pin is controlled by the TRISx and LATx bits

bit 11 **DACFM:** DACx Data Format Select bit
1 = Data is left justified (data stored in DACxDAT<15:8>)
0 = Data is right justified (data stored in DACxDAT<7:0>)

bit 10 **Unimplemented:** Read as ‘0’

bit 9 **SRDIS:** Soft Reset Disable bit
1 = DACxCON and DACxDAT SFRs reset only on a POR or BOR Reset
0 = DACxCON and DACxDAT SFRs reset on any type of device Reset

bit 8 **DACTRIG:** DACx Trigger Input Enable bit
1 = Analog output value updates when the selected (by DACTSEL<4:0>) event occurs
0 = Analog output value updates as soon as DACxDAT is written (DAC Trigger is ignored)

bit 7 **DACOE:** DACx Output Enable bit
1 = DACx output pin is enabled and driven on the DACxOUT pin
0 = DACx output pin is disabled, DACx output is available internally to other peripherals only

Note 1: BGBUF1 voltage is configured by BUFREF<1:0> (BUFCON0<1:0>).

REGISTER 25-8: DEVID: DEVICE ID REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
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 23-16 **Unimplemented:** Read as ‘0’

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits
01000101 = PIC24FV16KM204 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits
00011111 = PIC24FV16KM204
00011011 = PIC24FV16KM202
00010111 = PIC24FV08KM204
00010011 = PIC24FV08KM202
00001111 = PIC24FV16KM104
00001011 = PIC24FV16KM102
00000011 = PIC24FV08KM102
00000001 = PIC24FV08KM101

00011110 = PIC24F16KM204
00011010 = PIC24F16KM202
00010110 = PIC24F08KM204
00010010 = PIC24F08KM202
00001110 = PIC24F16KM104
00001010 = PIC24F16KM102
00000010 = PIC24F08KM102
00000000 = PIC24F08KM101

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.11 Demonstration/Development Boards, Evaluation Kits and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

26.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

TABLE 27-4: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions: 1.8V to 3.6V (PIC24F16KM204) 2.0V to 5.5V (PIC24FV16KM204) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended								
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
DC18	VHLVD	HLVD Voltage on VDD Transition	HLVDL<3:0> = 0000(2)	—	—	1.90	V	
			HLVDL<3:0> = 0001	1.88	—	2.13	V	
			HLVDL<3:0> = 0010	2.09	—	2.35	V	
			HLVDL<3:0> = 0011	2.25	—	2.53	V	
			HLVDL<3:0> = 0100	2.35	—	2.62	V	
			HLVDL<3:0> = 0101	2.55	—	2.84	V	
			HLVDL<3:0> = 0110	2.80	—	3.10	V	
			HLVDL<3:0> = 0111	2.95	—	3.25	V	
			HLVDL<3:0> = 1000	3.09	—	3.41	V	
			HLVDL<3:0> = 1001	3.27	—	3.59	V	
			HLVDL<3:0> = 1010(1)	3.46	—	3.79	V	
			HLVDL<3:0> = 1011(1)	3.62	—	4.01	V	
			HLVDL<3:0> = 1100(1)	3.91	—	4.26	V	
			HLVDL<3:0> = 1101(1)	4.18	—	4.55	V	
			HLVDL<3:0> = 1110(1)	4.49	—	4.87	V	

Note 1: These trip points should not be used on PIC24FXXKMXXX devices.
2: This trip point should not be used on PIC24FVXXKMXXX devices.

TABLE 27-5: BOR TRIP POINTS

Standard Operating Conditions: 1.8V to 3.6V (PIC24F16KM204) 2.0V to 5.5V (PIC24FV16KM204) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended								
Param No.	**Sym**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
DC15		BOR Hysteresis		—	5	—	mV	
DC19		BOR Voltage on VDD Transition	BORV<1:0> = 00	—	—	—	—	Valid for LPBOR (**Note 1**)
			BORV<1:0> = 01	2.90	3	3.38	V	
			BORV<1:0> = 10	2.53	2.7	3.07	V	
			BORV<1:0> = 11	1.75	1.85	2.05	V	**(Note 2)**
			BORV<1:0> = 11	1.95	2.05	2.16	V	**(Note 3)**

Note 1: LPBOR re-arms the POR circuit but does not cause a BOR.
2: This is valid for PIC24F (3.3V) devices.
3: This is valid for PIC24FV (5V) devices.

TABLE 27-7: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V (PIC24F16KMXXX) 2.0V to 5.5V (PIC24FV16KMXXX) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Parameter No.	**Device**	**Typical**	**Max**	**Units**	**Conditions**	
Idle Current (IIDLE)						
DC40	PIC24FV16KMXXX	120	200	μA	2.0V	0.5 MIPS, FOSC = 1 MHz(1)
		160	430	μA	5.0V	
	PIC24F16KMXXX	50	100	μA	1.8V	
		90	370	μA	3.3V	
DC42	PIC24FV16KMXXX	165	—	μA	2.0V	1 MIPS, FOSC = 2 MHz(1)
		260	—	μA	5.0V	
	PIC24F16KMXXX	95	—	μA	1.8V	
		180	—	μA	3.3V	
DC44	PIC24FV16KMXXX	3.1	6.5	mA	5.0V	16 MIPS, FOSC = 32 MHz(1)
	PIC24F16KMXXX	2.9	6.0	mA	3.3V	
DC46	PIC24FV16KMXXX	0.65	—	mA	2.0V	FRC (4 MIPS), FOSC = 8 MHz
		1.0	—	mA	5.0V	
	PIC24F16KMXXX	0.55	—	mA	1.8V	
		1.0	—	mA	3.3V	
DC50	PIC24FV16KMXXX	42	200	μA	2.0V	LPRC (15.5 KIPS), FOSC = 31 kHz
		65	225	μA	5.0V	
	PIC24F16KMXXX	2.2	18	μA	1.8V	
		4.0	40	μA	3.3V	

Legend: Unshaded rows represent PIC24F16KMXXX devices and shaded rows represent PIC24FV16KMXXX devices.

Note 1: The oscillator is in External Clock mode (FOSCSEL<2:0> = 010, FOSC<1:0> = 00).

TABLE 27-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V (PIC24F16KM204) 2.0V to 5.5V (PIC24FV16KM204) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Parameter No.	**Device**	**Typical(1)**	**Max**	**Units**	**Conditions**		
Power-Down Current (IPD)							
DC60	PIC24FV16KMXXX	6.0	—	μA	-40°C	2.0V	Sleep Mode(2)
			8.0		+25°C		
			8.5		+60°C		
			9.0		+85°C		
			15.0		+125°C		
		6.0	—	μA	-40°C	5.0V	
			8.0		+25°C		
			9.0		+60°C		
			10.0		+85°C		
			15.0		+125°C		
	PIC24F16KMXXX	0.025	—	μA	-40°C	1.8V	
			0.80		+25°C		
			1.5		+60°C		
			2.0		+85°C		
			7.5		+125°C		
		0.040	—	μA	-40°C	3.3V	
			1.0		+25°C		
			2.0		+60°C		
			3.0		+85°C		
			7.5		+125°C		
DC61	PIC24FV16KMXXX	0.25	—	μA	+85°C	2.0V	Low-Voltage Sleep Mode(2)
			7.5		+125°C		
		0.35	3.0	μA	+85°C	5.0V	
			7.5		+125°C		

Legend: Unshaded rows represent PIC24F16KMXXX devices and shaded rows represent PIC24FV16KMXXX devices.

Note 1: Data in the Typical column is at 3.3V, +25°C (PIC24F16KMXXX) or 5.0V, +25°C (PIC24FV16KMXXX) unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low. PMSLP is set to '0' and WDT, etc., are all switched off.

3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 27-15: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < T_A < +85°C (unless otherwise stated) -40°C ≤ T_A ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	V_{BG}	Band Gap Reference Voltage	0.973	1.024	1.075	V	V_{DD} > 4.5V for 4*V_{BG} reference V_{DD} > 2.3V for 2*V_{BG} reference
	T_{BG}	Band Gap Reference Start-up Time	—	1	—	ms	
	V_{RGOUT}	Regulator Output Voltage	3.1	3.3	3.6	V	
	C_{EFC}	External Filter Capacitor Value	4.7	10	—	µF	Series resistance < 3 Ohm recommended; < 5 Ohm is required.
	V_{LVR}	Low-Voltage Regulator Output Voltage	—	2.6	—	V	

TABLE 27-16: CTMU CURRENT SOURCE SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V (PIC24F16KM204)** **2.0V to 5.5V (PIC24FV16KM204)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended					
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Comments**	**Conditions**
	I_{OUT1}	CTMU Current Source, Base Range	—	550	—	nA	CTMUCON1L<1:0> = 01	2.5V < V_{DD} < V_{DDMAX}
	I_{OUT2}	CTMU Current Source, 10x Range	—	5.5	—	µA	CTMUCON1L<1:0> = 10	
	I_{OUT3}	CTMU Current Source, 100x Range	—	55	—	µA	CTMUCON1L<1:0> = 11	
	I_{OUT4}	CTMU Current Source, 1000x Range	—	550	—	µA	CTMUCON1L<1:0> = 00 **(Note 2)**	
	V_F	Temperature Diode Forward Voltage	—	.76	—	V		
	V_Δ	Voltage Change per Degree Celsius	—	1.6	—	mV/°C		

Note 1: Nominal value at the center point of the current trim range (CTMUCON1L<7:2> = 000000). On PIC24F16KM parts, the current output is limited to the typical current value when I_{OUT4} is chosen.

2: Do not use this current range with a temperature sensing diode.

TABLE 27-26: COMPARATOR TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
300	TRESP	Response Time*(1)	—	150	400	ns	
301	TMC2OV	Comparator Mode Change to Output Valid*	—	—	10	µs	

* Parameters are characterized but not tested.

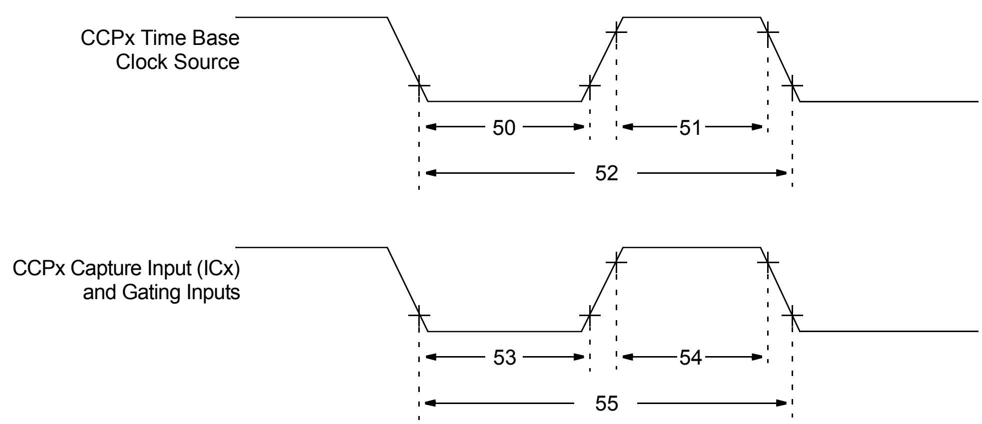
Note 1: Response time is measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from VSS to VDD.

TABLE 27-27: COMPARATOR VOLTAGE REFERENCE SETTLING TIME SPECIFICATIONS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
VR310	TSET	Settling Time(1)	—	—	10	µs	

Note 1: Settling time is measured while CVRSS = `1` and the CVR<3:0> bits transition from '`0000`' to '`1111`'.

FIGURE 27-10: CAPTURE/COMPARE/PWM TIMINGS (MCCPx, SCCPx MODULES)



Note: Refer to Figure 27-5 for load conditions.

TABLE 27-28: CAPTURE/COMPARE/PWM REQUIREMENTS (MCCPx, SCCPx MODULES)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
50	TCLKL	CCPx Time Base Clock Source Low Time	TCY/2	—	ns	
51	TCLKH	CCPx Time Base Clock Source High Time	TCY/2	—	ns	
52	TCLK	CCPx Time Base Clock Source Period	TCY	—	ns	
53	TCCL	CCPx Capture or Gating Input Low Time	TCLK	—	ns	
54	TCCH	CCPx Capture or Gating Input High Time	TCLK	—	ns	
55	TCCP	CCPx Capture or Gating Input Period	2 * TCLK/N	—	ns	N = Prescale Value (1, 4 or 16)