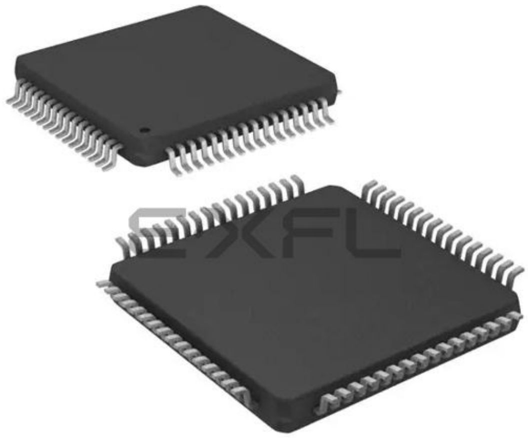
Microchip Technology - PIC24FJ128GA406-I/PT Datasheet



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	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, PMP/PSP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, LCD, LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b/12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128ga406-i-pt

TABLE 6: COMPLETE PIN FUNCTION DESCRIPTIONS FOR PIC24FJXXXGB412 (CONTINUED)

Pin	Function	Pin	Function
J1	SEG4/AN3/C2INA/IOCB3/RB3	K7	SEG8/AN14/**RP14**/CTED5/CTPLS/PMA1/PMALH/IOCB14/RB14
J2	SEG5/AN2/CTCMP/C2INB/**RP13**/CTED13/IOCB2/RB2	K8	VDD
J3	PGED2/SEG63/AN7/**RP7**/U6TX/IOCB7/RB7	K9	SEG39/**RP5**/IOCD15/RD15
J4	AVDD	K10	SEG12/**RP16**/USBID/IOCF3/RF3
J5	IOCH7/RH7	K11	SEG40/**RP30**/IOCF2/RF2
J6	TCK/IOCA1/RA1	L1	PGEC2/LCDBIAS3/AN6/**RP6**/IOCB6/RB6
J7	SEG18/AN12/U6RX/CTED2/PMA11/IOCB12/RB12	L2	SEG36/VREF-/CVREF-/PMA7/IOCA9/RA9
J8	IOCH9/RH9	L3	AVSS
J9	IOCH10/RH10	L4	SEG30/AN9/$\overline{\text{TMPR}}$/**RP9**/T1CK/IOCB9/RB9
J10	SEG41/**RP15**/IOCF8/RF8	L5	IOCH6/RH6
J11	D-/IOCG3/RG3	L6	SEG53/**RP31**/IOCF13/RF13
K1	PGEC1/SEG6/VREF-/CVREF-/AN1/AN1-/**RP1**/CTED12/IOCB1/RB1	L7	SEG19/AN13/SDI4/CTED1/PMA10/IOCB13/RB13
K2	PGED1/SEG7/VREF+/CVREF+/DVREF+/AN0/**RP0**/IOCB0/RB0	L8	SEG9/AN15/**RP29**/CTED6/PMA0/PMALL/IOCB15/RB15
K3	SEG37/VREF+/CVREF+/DVREF+/PMA6/IOCA10/RA10	L9	SEG38/**RPI43**/IOCD14/RD14
K4	SEG31/AN8/**RP8**/PWRGT/IOCB8/RB8	L10	SEG10/**RP10**/PMA9/IOCF4/RF4
K5	AN11/REFI1/$\overline{\text{SS4}}$/FSYNC4/PMA12/IOCB11/RB11	L11	SEG11/**RP17**/PMA8/IOCF5/RF5
K6	SEG54/**RPI32**/CTED7/PMA18/IOCF12/RF12		

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select functions.

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC24FJ64GA406
- PIC24FJ128GA406
- PIC24FJ256GA406
- PIC24FJ64GA410
- PIC24FJ128GA410
- PIC24FJ256GA410
- PIC24FJ64GA412
- PIC24FJ128GA412
- PIC24FJ256GA412
- PIC24FJ64GB406
- PIC24FJ128GB406
- PIC24FJ256GB406
- PIC24FJ64GB410
- PIC24FJ128GB410
- PIC24FJ256GB410
- PIC24FJ64GB412
- PIC24FJ128GB412
- PIC24FJ256GB412

The PIC24FJ256GA412/GB412 family expands the capabilities of the PIC24F family by adding a complete selection of advanced analog peripherals to its existing digital features. This combination, along with its ultra low-power features, Direct Memory Access (DMA) for peripherals, USB On-The-Go (OTG) and a built-in LCD Controller and driver, makes this family the new standard for mixed-signal PIC® microcontrollers in one economical and power-saving package.

1.1 Core Features

1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC® Digital Signal Controllers (DSCs). The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 32 Kbytes (data)
- A 16-element Working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32 by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as 'C'
- Operational performance up to 16 MIPS

1.1.2 XLP POWER-SAVING TECHNOLOGY

The PIC24FJ256GA412/GB412 family of devices incorporates a greatly expanded range of power-saving operating modes for the ultimate in power conservation. The new modes include:

- Retention Sleep, with essential circuits being powered from a separate low-voltage regulator
- Deep Sleep without RTCC, for the lowest possible power consumption under software control
- VBAT mode (with or without RTCC), to continue limited operation from a backup battery when VDD is removed

Many of these new low-power modes also support the continuous operation of the low-power, on-chip Real-Time Clock/Calendar (RTCC), making it possible for an application to keep time while the device is otherwise asleep.

Aside from these new features, the PIC24FJ256GA412/GB412 devices also include all of the legacy power-saving features of previous PIC24F microcontrollers, such as:

- On-the-Fly Clock Switching, allowing the selection of a lower power clock during run time
- Doze Mode Operation, for maintaining peripheral clock speed while slowing the CPU clock
- Instruction-Based Power-Saving Modes, for quick invocation of Idle and the many Sleep modes

1.1.3 DUAL PARTITION FLASH PROGRAM MEMORY

A brand new feature to the PIC24F family is the use of Dual Partition Flash program memory technology. This allows PIC24FJ256GA412/GB412 family devices a range of new operating options not available before:

- Dual Partition Operation, which can store two different applications in their own code partition, and allows for the support of robust bootloader applications and enhanced security
- Live Update Operation, which allows the main application to continue operation while the second Flash partition is being reprogrammed – all without adding Wait states to code execution
- Direct Run-Time Programming from Data RAM, with the option of data compression in the RAM image

PIC24FJ256GA412/GB412 family devices can also operate with their two Flash partitions as one large program memory, providing space for large and complex applications.

3.0 CPU

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 CPU, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"CPU with Extended Data Space (EDS)"** (DS39732). The information in this data sheet supersedes the information in the FRM.

The PIC24F CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set and a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M instructions of user program memory space. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the REPEAT instructions, which are interruptible at any point.

PIC24F devices have sixteen, 16-bit Working registers in the programmer's model. Each of the Working registers can act as a data, address or address offset register. The 16th Working register (W15) operates as a Software Stack Pointer (SSP) for interrupts and calls.

The lower 32 Kbytes of the Data Space (DS) can be accessed linearly. The upper 32 Kbytes of the Data Space are referred to as Extended Data Space to which the extended data RAM, EPMP memory space or program memory can be mapped.

The Instruction Set Architecture (ISA) has been significantly enhanced beyond that of the PIC18, but maintains an acceptable level of backward compatibility. All PIC18 instructions and addressing modes are supported, either directly, or through simple macros. Many of the ISA enhancements have been driven by compiler efficiency needs.

The core supports Inherent (no operand), Relative, Literal and Memory Direct Addressing modes, along with three groups of addressing modes. All modes support Register Direct and various Register Indirect modes. Each group offers up to seven addressing modes. Instructions are associated with predefined addressing modes depending upon their functional requirements.

For most instructions, the core is capable of executing a data (or program data) memory read, a Working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing trinary operations (that is, A + B = C) to be executed in a single cycle.

A high-speed, 17-bit x 17-bit multiplier has been included to significantly enhance the core arithmetic capability and throughput. The multiplier supports Signed, Unsigned and Mixed mode, 16-bit x 16-bit or 8-bit x 8-bit, integer multiplication. All multiply instructions execute in a single cycle.

The 16-bit ALU has been enhanced with integer divide assist hardware that supports an iterative non-restoring divide algorithm. It operates in conjunction with the REPEAT instruction looping mechanism and a selection of iterative divide instructions to support 32-bit (or 16-bit), divided by 16-bit, integer signed and unsigned division. All divide operations require 19 cycles to complete but are interruptible at any cycle boundary.

The PIC24F has a vectored exception scheme with up to 8 sources of non-maskable traps and up to 118 interrupt sources. Each interrupt source can be assigned to one of seven priority levels.

A block diagram of the CPU is shown in Figure 3-1.

3.1 Programmer's Model

The programmer's model for the PIC24F is shown in Figure 3-2. All registers in the programmer's model are memory-mapped and can be manipulated directly by instructions.

A description of each register is provided in Table 3-1. All registers associated with the programmer's model are memory-mapped.

4.3.5.2 Data Write into EDS

In order to write data to EDS space, such as in EDS reads, an Address Pointer is set up by loading the required EDS page number into the DSWPAG register and assigning the offset address to one of the W registers. Once the above assignment is done, then the EDS window is enabled by setting bit 15 of the Working register assigned with the offset address and the accessed location can be written.

Figure 4-8 illustrates how the EDS space address is generated for write operations.

When the MSb of EA is '1', the lower 9 bits of DSWPAG are concatenated to the lower 15 bits of EA to form a 24-bit EDS address for write operations. Example 4-2 shows how to write a byte, word and double-word to EDS.

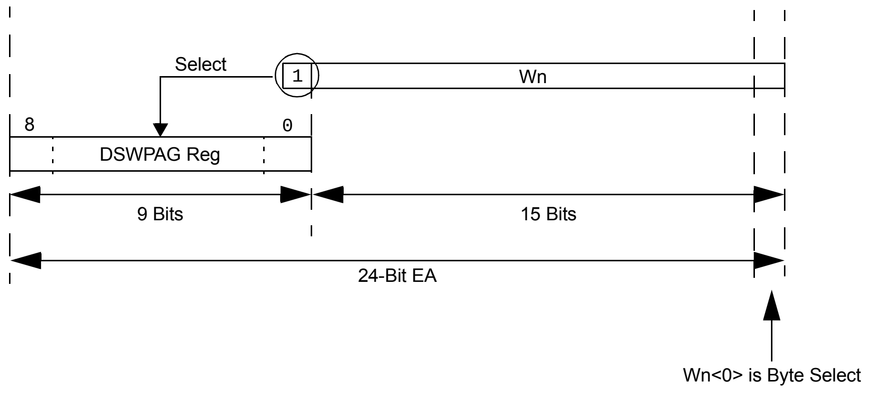
The DS Page registers (DSRPAG/DSWPAG) do not update automatically while crossing a page boundary when the rollover happens from 0xFFFF to 0x8000. While developing code in assembly, care must be taken to update the DS Page registers when an Address Pointer crosses the page boundary. The 'C' compiler keeps track of the addressing, and increments or decrements the DS Page registers accordingly, while accessing contiguous data memory locations.

Note 1: All write operations to EDS are executed in a single cycle.

2: Use of Read/Modify/Write operation on any EDS location under a REPEAT instruction is not supported. For example, BCLR, BSW, BTG, RLC f, RLNC f, RRC f, RRNC f, ADD f, SUB f, SUBR f, AND f, IOR f, XOR f, ASR f, ASL f.

3: Use the DSRPAG register while performing Read/Modify/Write operations.

FIGURE 4-8: EDS ADDRESS GENERATION FOR WRITE OPERATIONS



EXAMPLE 4-2: EDS WRITE CODE IN ASSEMBLY

```
; Set the EDS page where the data to be written
    mov     #0x0002, w0
    mov     w0, DSWPAG     ;page 2 is selected for write
    mov     #0x0800, w1    ;select the location (0x800) to be written
    bset    w1, #15        ;set the MSB of the base address, enable EDS mode

;Write a byte to the selected location
    mov     #0x00A5, w2
    mov     #0x003C, w3
    mov.b   w2, [w1++]     ;write Low byte
    mov.b   w3, [w1++]     ;write High byte

;Write a word to the selected location
    mov     #0x1234, w2    ;
    mov     w2, [w1]       ;

;Write a Double - word to the selected location
    mov     #0x1122, w2
    mov     #0x4455, w3
    mov.d   w2, [w1]       ;2 EDS writes
```

REGISTER 8-31: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC3IP2	IC3IP1	IC3IP0	—	DMA3IP2	DMA3IP1	DMA3IP0
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 **Unimplemented:** Read as ‘0’

bit 14-12 **IC5IP<2:0>:** Input Capture Channel 5 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **IC4IP<2:0>:** Input Capture Channel 4 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 **IC3IP<2:0>:** Input Capture Channel 3 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **DMA3IP<2:0>:** DMA Channel 3 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 7 **CLKLOCK:** Clock Selection Lock Enable bit

If FSCM is enabled (FCKSM1 = 1):
1 = Clock and PLL selections are locked
0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit

If FSCM is disabled (FCKSM1 = 0):
Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.

bit 6 **IOLOCK:** I/O Lock Enable bit(2)

1 = I/O lock is active
0 = I/O lock is not active

bit 5 **LOCK:** PLL Lock Status bit(3)

1 = PLL module is in lock or PLL module start-up timer is satisfied
0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled

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

bit 3 **CF:** Clock Fail Detect bit

1 = FSCM has detected a clock failure
0 = No clock failure has been detected

bit 2 **POSCEN:** Primary Oscillator Sleep Enable bit

1 = Primary Oscillator continues to operate during Sleep mode
0 = Primary Oscillator is disabled during Sleep mode

bit 1 **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit

1 = Enables Secondary Oscillator
0 = Disables Secondary Oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit

1 = Initiates an oscillator switch to a clock source specified by the NOSC<2:0> bits
0 = Oscillator switch is complete

Note 1: Reset values for these bits are determined by the FNOSCx Configuration bits.

2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1' once the IOLOCK bit is set, it cannot be cleared.

3: This bit also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

4: The default divisor of the postscaler is 2, which will generate a 4 MHz clock to the PLL module.

10.0 POWER-SAVING FEATURES

> **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, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Power-Saving Features with Deep Sleep"** (DS39727). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ256GA412/GB412 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked reduces consumed power.

PIC24FJ256GA412/GB412 family devices manage power consumption with five strategies:

- Instruction-Based Power Reduction Modes
- Hardware-Based Power Reduction Features
- Clock Frequency Control
- Software Controlled Doze Mode
- Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical application features, such as timing-sensitive communications.

10.1 Overview of Power-Saving Modes

In addition to full-power operation, otherwise known as Run mode, the PIC24FJ256GA412/GB412 family of devices offers three Instruction-Based Power-Saving modes and one Hardware-Based mode:

- Idle
- Sleep (Sleep and Low-Voltage Sleep)
- Deep Sleep (without retention)
- VBAT (with and without RTCC)

All four modes can be activated by powering down different functional areas of the microcontroller, allowing progressive reductions of operating and Idle power consumption. In addition, three of the modes can be tailored for more power reduction at a trade-off of some operating features. Table 10-1 lists all of the operating modes in order of increasing power savings. Table 10-2 summarizes how the microcontroller exits the different modes. Specific information is provided in the following sections.

TABLE 10-1: OPERATING MODES FOR PIC24FJ256GA412/GB412 FAMILY DEVICES

Mode	Entry	Active Systems				
		Core	Peripherals	Data RAM Retention	RTCC[1]	DSGPR0/ DSGPR1 Retention
Run (default)	N/A	Y	Y	Y	Y	Y
Idle	Instruction	N	Y	Y	Y	Y
Sleep:						
Sleep	Instruction	N	S[2]	Y	Y	Y
Low-Voltage Sleep	Instruction + RETEN bit	N	S[2]	Y	Y	Y
Deep Sleep:						
Deep Sleep	Instruction + DSEN bit	N	N	N	Y	Y
VBAT:						
with RTCC	Hardware	N	N	N	Y	Y

Note 1: If RTCC is otherwise enabled in firmware.

2: A select peripheral can operate during this mode from LPRC or some external clock.

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER[1] (CONTINUED)

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit

When TCS = 1:
1 = Synchronizes external clock input
0 = Does not synchronize external clock input

When TCS = 0:
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit

1 = Extended clock selected by the TECS<1:0> bits
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: Changing the value of T1CON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

2: The TMRCK input must also be assigned to an available RPn or RPIn pin. See **Section 11.5 “Peripheral Pin Select (PPS)”** for more information.

REGISTER 13-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER[1] (CONTINUED)

bit 3 **T32:** 32-Bit Timer Mode Select bit[3]

1 = Timerx and Timery form a single 32-bit timer
0 = Timerx and Timery act as two 16-bit timers
In 32-bit mode, T3CON control bits do not affect 32-bit timer operation.

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timerx Clock Source Select bit[2]

1 = Timer source is selected by TECS<1:0>
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

2: If TCS = 1 and TECS<1:0> = x1, the selected external timer input (TMRCK or TxCK) must be configured to an available RPn/RPIn pin. For more information, see **Section 11.5 “Peripheral Pin Select (PPS)”**.

3: In T4CON, the T45 bit is implemented instead of T32 to select 32-bit mode. In 32-bit mode, the T3CON or T5CON control bits do not affect 32-bit timer operation.

REGISTER 14-1: CCPxCON1L: CCPx CONTROL 1 LOW REGISTERS (CONTINUED)

bit 3-0 **MOD<3:0>:** CCPx Mode Select bits

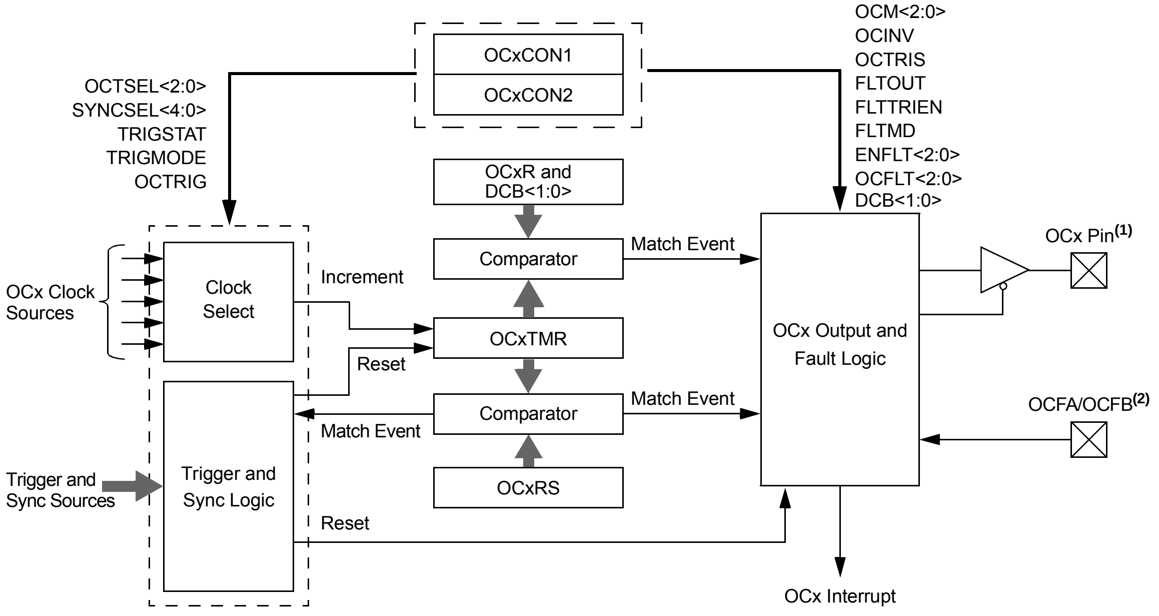
For CCSEL = 1 (Input Capture Modes):
1xxx = Reserved
011x = Reserved
0101 = Capture every 16th rising edge
0100 = Capture every 4th rising edge
0011 = Capture every rising and falling edge
0010 = Capture every falling edge
0001 = Capture every rising edge
0000 = Capture every rising and falling edge (Edge Detect mode)

For CCSEL = 0 (Output Compare/Timer Modes):
1111 = External Input mode: Pulse generator is disabled, source is selected by ICS<2:0>
1110 = Reserved
110x = Reserved
10xx = Reserved
0111 = Variable Frequency Pulse mode
0110 = Center-Aligned Pulse Compare mode, buffered
0101 = Dual Edge Compare mode, buffered
0100 = Dual Edge Compare mode
0011 = 16-Bit/32-Bit Single Edge mode, toggles output on compare match
0010 = 16-Bit/32-Bit Single Edge mode, drives output low on compare match
0001 = 16-Bit/32-Bit Single Edge mode, drives output high on compare match
0000 = 16-Bit/32-Bit Timer mode, output functions are disabled

TABLE 14-5: CLC CLOCK SOURCE SELECTION (CLKSEL<2:0> = 011)

MCCPx/SCCPx Module	MCCP1	SCCP2	SCCP3	SCCP4	SCCP5	SCCP6	SCCP7
CLC Module for Clock Source	1	2	3	1	2	3	4

FIGURE 16-1: OUTPUT COMPARE x BLOCK DIAGRAM (16-BIT MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. For more information, see **Section 11.5 "Peripheral Pin Select (PPS)"**.

2: The OCFA/OCFB Fault inputs must be assigned to an available RPn/RPIn pin before use. For more information, see **Section 11.5 "Peripheral Pin Select (PPS)"**.

16.2 Compare Operations

In Compare mode (Figure 16-1), the output compare module can be configured for single-shot or continuous pulse generation. It can also repeatedly toggle an output pin on each timer event.

To set up the module for compare operations:

1. Configure the OCx output for one of the available Peripheral Pin Select pins.
2. Calculate the required values for the OCxR and (for Double Compare modes) OCxRS Duty Cycle registers:
   a) Determine the instruction clock cycle time. Take into account the frequency of the external clock to the timer source (if one is used) and the timer prescaler settings.
   b) Calculate the time to the rising edge of the output pulse relative to the timer start value (0000h).
   c) Calculate the time to the falling edge of the pulse based on the desired pulse width and the time to the rising edge of the pulse.
3. Write the rising edge value to OCxR and the falling edge value to OCxRS.
4. Set the Timer Period register, PRy, to a value equal to or greater than the value in OCxRS.
5. Set the OCM<2:0> bits for the appropriate compare operation ('0xx').
6. For Trigger mode operations, set OCTRIG to enable Trigger mode. Set or clear TRIGMODE to configure trigger operation and TRIGSTAT to select a hardware or software trigger. For Synchronous mode, clear OCTRIG.
7. Set the SYNCSEL<4:0> bits to configure the trigger or synchronization source. If free-running timer operation is required, set the SYNCSELx bits to '00000' (no sync/trigger source).
8. Select the time base source with the OCTSEL<2:0> bits. If necessary, set the TON bits for the selected timer, which enables the compare time base to count. Synchronous mode operation starts as soon as the time base is enabled; Trigger mode operation starts after a trigger source event occurs.

18.2 Setting Baud Rate When Operating as a Bus Master

To compute the Baud Rate Generator reload value, use Equation 18-1.

EQUATION 18-1: COMPUTING BAUD RATE RELOAD VALUE[1,2]

$$F_{SCL} = \frac{F_{CY}}{2 \cdot (BRG + 2)}$$

or

$$BRG = \left(\frac{F_{CY}}{2 \cdot F_{SCL}}\right) - 2$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system-level parameters. The actual clock rate should be measured in its intended application.

18.3 Slave Address Masking

The I2CxMSK register (Register 18-4) designates address bit positions as “don’t care” for both 7-Bit and 10-Bit Addressing modes. Setting a particular bit location (= `1`) in the I2CxMSK register causes the slave module to respond, whether the corresponding address bit value is a ‘`0`’ or a ‘`1`’. For example, when I2CxMSK is set to ‘`0010000000`’, the slave module will detect both addresses, ‘`0000000000`’ and ‘`0010000000`’.

To enable address masking, the Intelligent Peripheral Management Interface (IPMI) must be disabled by clearing the STRICT bit (I2CxCONH<11>).

Note: As a result of changes in the I²C protocol, the addresses in Table 18-2 are reserved and will not be Acknowledged in Slave mode. This includes any address mask settings that include any of these addresses.

TABLE 18-1: I2Cx CLOCK RATES[1,2]

Required System FSCL	FCY	I2CxBRG Value		Actual FSCL
		(Decimal)	(Hexadecimal)	
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system-level parameters. The actual clock rate should be measured in its intended application.

TABLE 18-2: I2Cx RESERVED ADDRESSES[1]

Slave Address	R/$\overline{W}$ Bit	Description
`0000 000`	`0`	General Call Address[2]
`0000 000`	`1`	Start Byte
`0000 001`	`x`	CBus Address
`0000 01x`	`x`	Reserved
`0000 1xx`	`x`	HS Mode Master Code
`1111 0xx`	`x`	10-Bit Slave Upper Byte[3]
`1111 1xx`	`x`	Reserved

Note 1: The address bits listed here will never cause an address match independent of address mask settings.

2: This address will be Acknowledged only if GCEN = `1`.

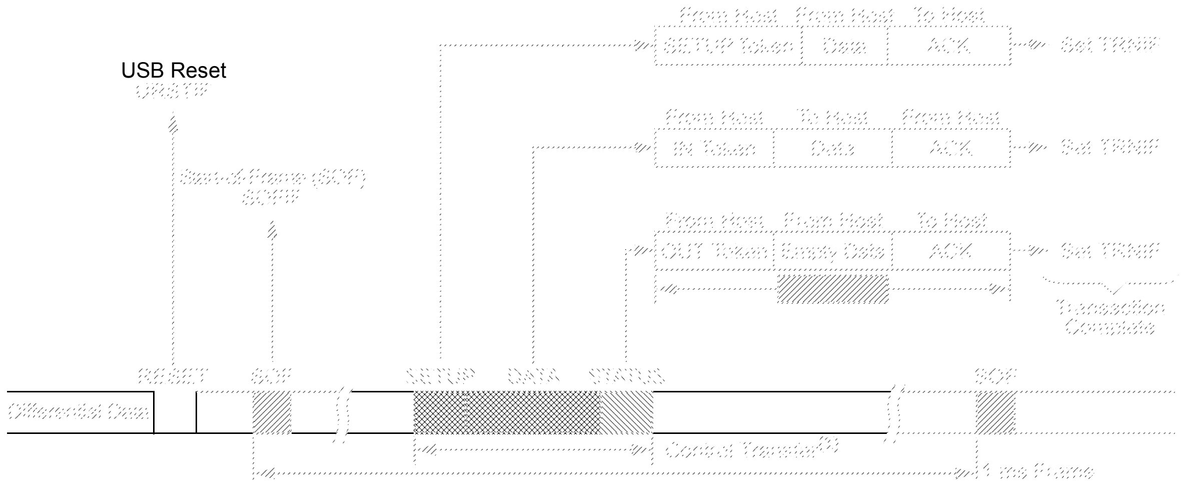
3: A match on this address can only occur on the upper byte in 10-Bit Addressing mode.

20.3.1 CLEARING USB OTG INTERRUPTS

Unlike device-level interrupts, the USB OTG interrupt status flags are not freely writable in software. All USB OTG flag bits are implemented as hardware settable only bits. Additionally, these bits can only be cleared in software by writing a '1' to their locations (i.e., performing a MOV type instruction). Writing a '0' to a flag bit (i.e., a BCLR instruction) has no effect.

> **Note:** Throughout this data sheet, a bit that can only be cleared by writing a '1' to its location is referred to as "Write 1 to Clear". In register descriptions, this function is indicated by the descriptor, "K".

FIGURE 20-9: EXAMPLE OF A USB TRANSACTION AND INTERRUPT EVENTS



Note 1: The control transfer shown here is only an example showing events that can occur for every transaction. Typical control transfers will spread across multiple frames.

20.4 Device Mode Operation

The following section describes how to perform a common Device mode task. In Device mode, USB transfers are performed at the transfer level. The USB module automatically performs the status phase of the transfer.

20.4.1 ENABLING DEVICE MODE

1. Reset the Ping-Pong Buffer Pointers by setting, then clearing, the Ping-Pong Buffer Reset bit, PPBRST (U1CON<1>).
2. Disable all interrupts (U1IE and U1EIE = 00h).
3. Clear any existing interrupt flags by writing FFh to U1IR and U1EIR.
4. Verify that VBUS is present (non-OTG devices only).
5. Enable the USB module by setting the USBEN bit (U1CON<0>).
6. Set the OTGEN bit (U1OTGCON<2>) to enable OTG operation.
7. Enable the endpoint zero buffer to receive the first setup packet by setting the EPRXEN and EPHSHK bits for Endpoint 0 (U1EP0<3,0> = 1).
8. Power up the USB module by setting the USBPWR bit (U1PWRC<0>).
9. Enable the D+ pull-up resistor to signal an attach by setting the DPPULUP bit (U1OTGCON<7>).

20.5.2 COMPLETE A CONTROL TRANSACTION TO A CONNECTED DEVICE

1. Follow the procedure described in **Section 20.5.1 “Enable Host Mode and Discover a Connected Device”** to discover a device.
2. Set up the Endpoint Control register for bidirectional control transfers by writing 0Dh to U1EP0 (this sets the EPCONDIS, EPTXEN and EPHSHK bits).
3. Place a copy of the device framework setup command in a memory buffer. See Chapter 9 of the *“USB 2.0 Specification”* for information on the device framework command set.
4. Initialize the Buffer Descriptor (BD) for the current (even or odd) TX EP0 to transfer the eight bytes of command data for a device framework command (i.e., `GET DEVICE DESCRIPTOR`):
   a) Set the BD Data Buffer Address (BD0ADR) to the starting address of the 8-byte memory buffer containing the command.
   b) Write 8008h to BD0STAT (this sets the UOWN bit and sets a byte count of 8).
5. Set the USB device address of the target device in the address register (U1ADDR<6:0>). After a USB bus Reset, the device USB address will be zero. After enumeration, it will be set to another value between 1 and 127.
6. Write D0h to U1TOK; this is a SETUP token to Endpoint 0, the target device’s default control pipe. This initiates a SETUP token on the bus, followed by a data packet. The device handshake is returned in the PID field of BD0STAT after the packets are complete. When the USB module updates BD0STAT, a Token Complete Interrupt Flag is asserted (the TRNIF flag is set). This completes the setup phase of the setup transaction, as referenced in Chapter 9 of the *“USB 2.0 Specification”*.
7. To initiate the data phase of the setup transaction (i.e., get the data for the `GET DEVICE DESCRIPTOR` command), set up a buffer in memory to store the received data.
8. Initialize the current (even or odd) RX or TX (RX for IN, TX for OUT) EP0 BD to transfer the data.
   a) Write C040h to BD0STAT. This sets the UOWN, configures Data Toggle (DTS) to DATA1 and sets the byte count to the length of the data buffer (64 or 40h in this case).
   b) Set BD0ADR to the starting address of the data buffer.
9. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device’s default control pipe (e.g., write 90h to U1TOK for an IN token for a `GET DEVICE DESCRIPTOR` command). This initiates an IN token on the bus, followed by a data packet from the device to the host. When the data packet completes, the BD0STAT is written and a Token Complete Interrupt Flag is asserted (the TRNIF flag is set). For control transfers with a single packet data phase, this completes the data phase of the setup transaction, as referenced in Chapter 9 of the *“USB 2.0 Specification”*. If more data needs to be transferred, return to Step 8.
10. To initiate the status phase of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current (even or odd) TX EP0 BD to transfer the status data:
    a) Set the BDT buffer address field to the start address of the data buffer.
    b) Write 8000h to BD0STAT (set UOWN bit, configure DTS to DATA0 and set byte count to 0).
12. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device’s default control pipe (e.g., write 01h to U1TOK for an OUT token for a `GET DEVICE DESCRIPTOR` command). This initiates an OUT token on the bus, followed by a zero length data packet from the host to the device. When the data packet completes, the BD is updated with the handshake from the device and a Token Complete Interrupt Flag is asserted (the TRNIF flag is set). This completes the status phase of the setup transaction, as described in Chapter 9 of the *“USB 2.0 Specification”*.

Note:	Only one control transaction can be performed per frame.

TABLE 23-1: MODULE-SPECIFIC INPUT DATA SOURCES

Bit Field Value		Input Source			
		CLC1	CLC2	CLC3	CLC4
DS4<2:0>	`011`	SDI1	SDI2	SDI3	SDI4
	`001`	CLC2 Output	CLC1 Output	CLC4 Output	CLC3 Output
DS3<2:0>	`100`	U1RX	U2RX	U3RX	U4RX
	`011`	SDO1	SDO2	SDO3	SDO4
	`001`	CLC1 Output	CLC2 Output	CLC3 Output	CLC4 Output
DS2<2:0>	`011`	U1TX	U2TX	U3TX	U4TX
	`001`	CLC2 Output	CLC1 Output	CLC4 Output	CLC3 Output

REGISTER 23-4: CLCxGLSL: CLCx GATE LOGIC INPUT SELECT LOW REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N
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
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
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 **G2D4T:** Gate 2 Data Source 4 True Enable bit
`1` = Data Source 4 inverted signal is enabled for Gate 2
`0` = Data Source 4 inverted signal is disabled for Gate 2

bit 14 **G2D4N:** Gate 2 Data Source 4 Negated Enable bit
`1` = Data Source 4 inverted signal is enabled for Gate 2
`0` = Data Source 4 inverted signal is disabled for Gate 2

bit 13 **G2D3T:** Gate 2 Data Source 3 True Enable bit
`1` = Data Source 3 inverted signal is enabled for Gate 2
`0` = Data Source 3 inverted signal is disabled for Gate 2

bit 12 **G2D3N:** Gate 2 Data Source 3 Negated Enable bit
`1` = Data Source 3 inverted signal is enabled for Gate 2
`0` = Data Source 3 inverted signal is disabled for Gate 2

bit 11 **G2D2T:** Gate 2 Data Source 2 True Enable bit
`1` = Data Source 2 inverted signal is enabled for Gate 2
`0` = Data Source 2 inverted signal is disabled for Gate 2

bit 10 **G2D2N:** Gate 2 Data Source 2 Negated Enable bit
`1` = Data Source 2 inverted signal is enabled for Gate 2
`0` = Data Source 2 inverted signal is disabled for Gate 2

bit 9 **G2D1T:** Gate 2 Data Source 1 True Enable bit
`1` = Data Source 1 inverted signal is enabled for Gate 2
`0` = Data Source 1 inverted signal is disabled for Gate 2

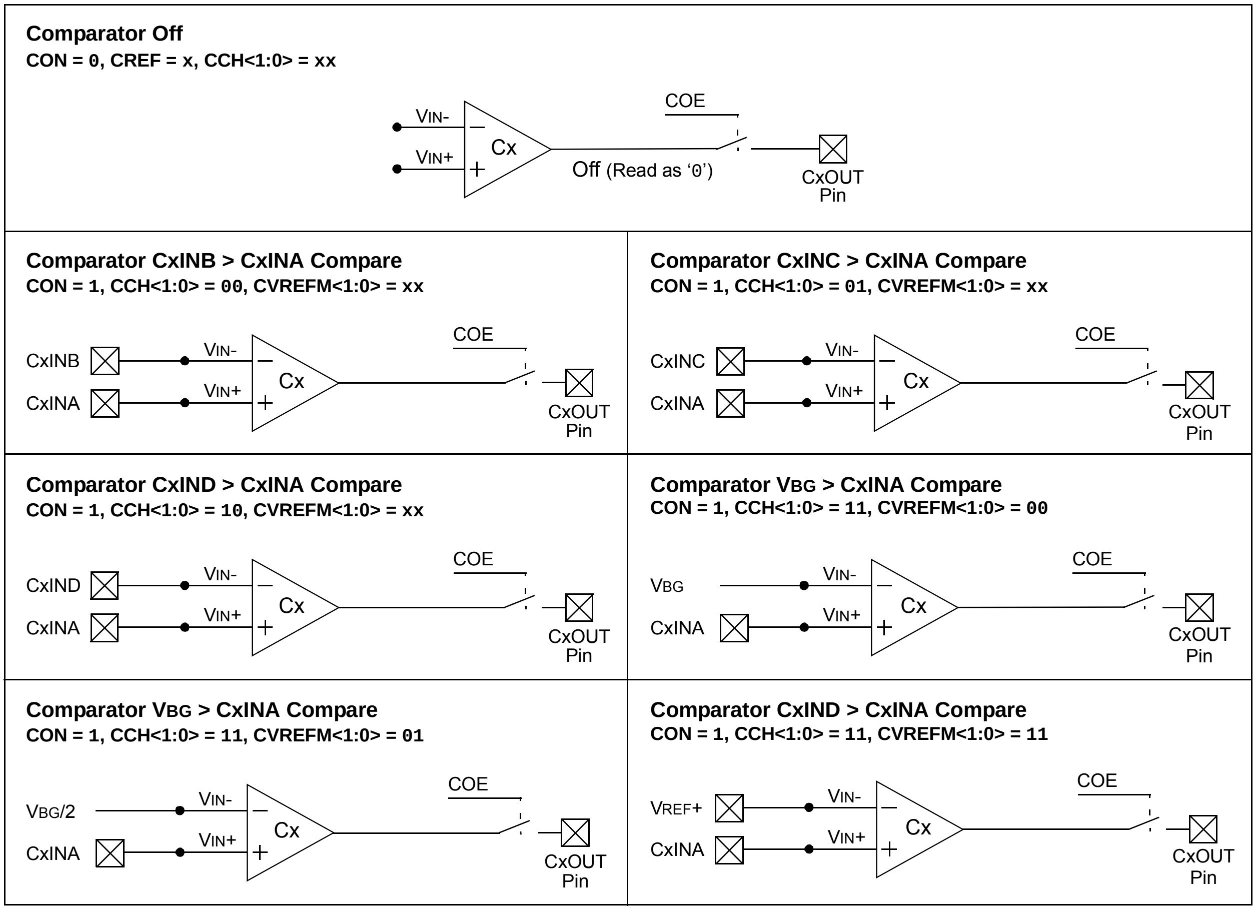
25.4 Encrypting Data

1. If not already set, set the CRYON bit.
2. Configure the CPHRSEL, CPHRMODx, KEYMODx and KEYSRCx bits as desired to select the proper mode and key length.
3. Set OPMOD<3:0> to '0000'.
4. If a software key is being used, write it to the CRYKEY register. It is only necessary to write the lowest *n* bits of CRYKEY for a key length of *n*, as all unused CRYKEY bits are ignored.
5. Read the KEYFAIL bit. If this bit is '1', an illegal configuration has been selected and the encrypt operation will NOT be performed.
6. Write the data to be encrypted to the appropriate CRYTXT register. For a single DES encrypt operation, it is only necessary to write the lowest 64 bits. However, for data less than the block size (64 bits for DES, 128 bits for AES), it is the responsibility of the software to properly pad the upper bits within the block.
7. Set the CRYGO bit.
8. In ECB and CBC modes, set the FREEIE bit (CRYCONL<10>) to enable the optional CRYTXTA interrupt to indicate when the next plaintext block can be loaded.
9. Poll the CRYGO bit until it is cleared or wait for the CRYDNIF module interrupt (DONEIE must be set). If other Cryptographic Engine interrupts are enabled, it will be necessary to poll the CRYGO bit to verify the interrupt source.
10. Read the encrypted block from the appropriate CRYTXT register.
11. Repeat Steps 5 through 8 to encrypt further blocks in the message with the same key.

25.5 Decrypting Data

1. If not already set, set the CRYON bit.
2. Configure the CPHRSEL, CPHRMODx, KEYMODx and KEYSRCx bits as desired to select the proper mode and key length.
3. Set OPMOD<3:0> to '0001'.
4. If a software key is being used, write the CRYKEY register. It is only necessary to write the lowest *n* bits of CRYKEY for a key length of *n*, as all unused CRYKEY bits are ignored.
5. If an AES-ECB or AES-CBC mode decryption is being performed, you must first perform an AES decryption key expansion operation.
6. Read the KEYFAIL status bit. If this bit is '1', an illegal configuration has been selected and the encrypt operation will not be performed.
7. Write the data to be decrypted into the appropriate Text/Data register. For a DES decrypt operation, it is only necessary to write the lowest 64 bits of CRYTXTB.
8. Set the CRYGO bit.
9. If this is the first decrypt operation after a Reset, or if a key storage program operation was performed after the last decrypt operation, or if the KEYMODx or KEYSRCx fields are changed, the engine will perform a new key expansion operation. This will result in extra clock cycles for the decrypt operation, but will otherwise be transparent to the application (i.e., the CRYGO bit will be cleared only after the key expansion and the decrypt operation have completed).
10. In ECB and CBC modes, set the FREEIE bit (CRYCONL<10>) to enable the optional CRYTXTA interrupt to indicate when the next plaintext block can be loaded.
11. Poll the CRYGO bit until it is cleared or wait for the CRYDNIF module interrupt (DONEIE must be set). If other Cryptographic Engine interrupts are enabled, it will be necessary to poll the CRYGO bit to verify the interrupt source.
12. Read the decrypted block out of the appropriate Text/Data register.
13. Repeat Steps 6 through 10 to encrypt further blocks in the message with the same key.

FIGURE 29-2: INDIVIDUAL COMPARATOR CONFIGURATIONS WHEN CREF = 0



Comparator Off
CON = 0, CREF = x, CCH<1:0> = xx

Comparator CxINB > CxINA Compare
CON = 1, CCH<1:0> = 00, CVREFM<1:0> = xx

Comparator CxINC > CxINA Compare
CON = 1, CCH<1:0> = 01, CVREFM<1:0> = xx

Comparator CxIND > CxINA Compare
CON = 1, CCH<1:0> = 10, CVREFM<1:0> = xx

Comparator VBG > CxINA Compare
CON = 1, CCH<1:0> = 11, CVREFM<1:0> = 00

Comparator VBG > CxINA Compare
CON = 1, CCH<1:0> = 11, CVREFM<1:0> = 01

Comparator CxIND > CxINA Compare
CON = 1, CCH<1:0> = 11, CVREFM<1:0> = 11

33.3 Watchdog Timer (WDT)

For PIC24FJ256GA412/GB412 family devices, the WDT is driven by the LPRC Oscillator. When the WDT is enabled, the clock source is also enabled.

The nominal WDT clock source from LPRC is 31 kHz. This feeds a prescaler that can be configured for either 5-bit (divide-by-32) or 7-bit (divide-by-128) operation. The prescaler is set by the FWPSA Configuration bit. With a 31 kHz input, the prescaler yields a nominal WDT Time-out period (TWDT) of 1 ms in 5-bit mode or 4 ms in 7-bit mode.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the WDTPS<3:0> Configuration bits (FWDT<3:0>), which allows the selection of a total of 16 settings, from 1:1 to 1:32,768. Using the prescaler and postscaler time-out periods, ranges from 1 ms to 131 seconds can be achieved.

The WDT, prescaler and postscaler are reset:

- On any device Reset
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSCx bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- When a PWRSAV instruction is executed (i.e., Sleep or Idle mode is entered)
- When the device exits Sleep or Idle mode to resume normal operation
- By a CLRWDT instruction during normal execution

If the WDT is enabled, it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake the device and code execution will continue from where the PWRSAV instruction was executed. The corresponding SLEEP or IDLE (RCON<3:2>) bit will need to be cleared in software after the device wakes up.

The WDT Flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

Note:	The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

33.3.1 WINDOWED OPERATION

The Watchdog Timer has an optional Fixed Window mode of operation. In this Windowed mode, CLRWDT instructions can only reset the WDT during the last 1/4 of the programmed WDT period. A CLRWDT instruction executed before that window causes a WDT Reset, similar to a WDT time-out.

Windowed WDT mode is enabled by programming the WINDIS Configuration bit (FWDT<7>) to ‘0’.

33.3.2 CONTROL REGISTER

The WDT is enabled or disabled by the FWDTEN<1:0> Configuration bits. When the Configuration bits, FWDTEN<1:0> = 11, the WDT is always enabled.

The WDT can be optionally controlled in software when the Configuration bits, FWDTEN<1:0> = 10. When FWDTEN<1:0> = 00, the Watchdog Timer is always disabled. The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user to enable the WDT for critical code segments and disable the WDT during non-critical segments for maximum power savings.

FIGURE 33-2: WDT BLOCK DIAGRAM

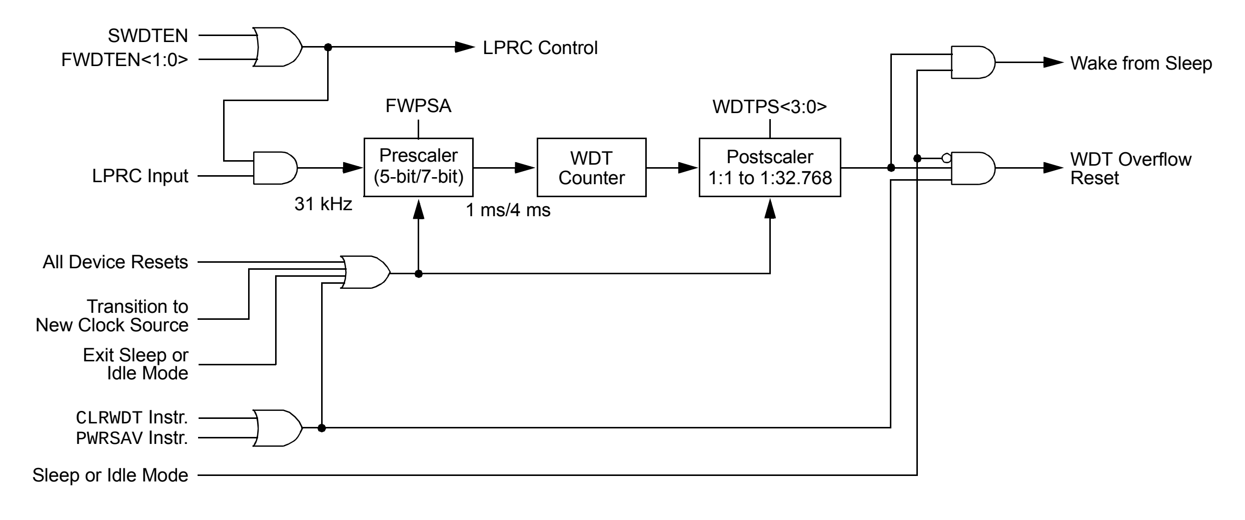


TABLE 36-14: USB ON-THE-GO MODULE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
Operating Voltage							
DUS01	VUSB3V3	USB Supply Voltage	Greater of: 3.0 or (VDD – 0.3V)	3.3	3.6	V	USB module enabled
			(VDD – 0.3V)(1)	—	3.6	V	USB disabled, RG2/RG3 are unused and externally pulled low or left in a high-impedance state
			(VDD – 0.3V)	VDD	3.6	V	USB disabled, RG2/RG3 are used as general purpose I/Os

Note 1: The VUSB3V3 pin may also be left in a high-impedance state under these conditions. However, if the voltage floats below (VDD – 0.3V), this may result in higher IPD currents than specified. The preferred method is to tie the VUSB pin to VDD, even if the USB module is not used.

TABLE 36-15: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)								
Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
DC18	VHLVD	HLVD Voltage on VDD Transition	HLVDL<3:0> = 0100(1)	3.45	—	3.73	V	
			HLVDL<3:0> = 0101	3.30	—	3.57	V	
			HLVDL<3:0> = 0110	3.00	—	3.25	V	
			HLVDL<3:0> = 0111	2.80	—	3.03	V	
			HLVDL<3:0> = 1000	2.67	—	2.92	V	
			HLVDL<3:0> = 1001	2.45	—	2.70	V	
			HLVDL<3:0> = 1010	2.33	—	2.60	V	
			HLVDL<3:0> = 1011	2.21	—	2.49	V	
			HLVDL<3:0> = 1100	2.11	—	2.38	V	
			HLVDL<3:0> = 1101	2.10	—	2.25	V	
			HLVDL<3:0> = 1110	2.00	—	2.15	V	
DC101	VTHL	HLVD Voltage on LVDIN Pin Transition	HLVDL<3:0> = 1111	—	1.20	—	V	

Note 1: Trip points for values of HLVD<3:0>, from '0000' to '0011', are not implemented.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support