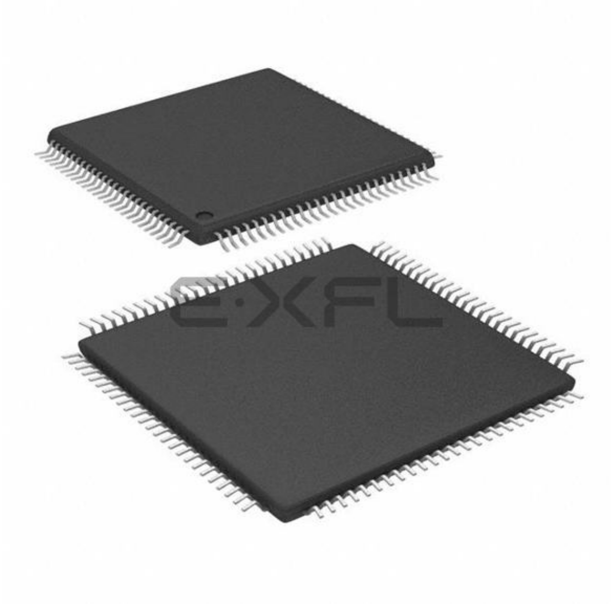
Microchip Technology - PIC24FJ64GB110-I/PT Datasheet



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	16-Bit
Speed	32MHz
Connectivity	I²C, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	83
Program Memory Size	64KB (22K 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
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64gb110-i-pt

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
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0[(1)]	R/W-0[(1)]	R/W-0[(1)]	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2[(2)]	IPL1[(2)]	IPL0[(2)]	RA	N	OV	Z	C
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 15-9 **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 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 of the result occurred
0 = No carry out from the Most Significant bit of the result occurred

Note 1: The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

2: The IPL 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.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® devices and improve data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word which contains the byte, using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the Least Significant Byte. The Most Significant Byte is not modified.

A sign-extend instruction (`SE`) is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users can clear the MSB of any W register by executing a zero-extend (`ZE`) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

4.2.3 NEAR DATA SPACE

The 8-Kbyte area between 0000h and 1FFFh is referred to as the near data space. Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. The remainder of the data space is addressable indirectly. Additionally, the whole data space is addressable using `MOV` instructions, which support Memory Direct Addressing with a 16-bit address field.

4.2.4 SFR SPACE

The first 2 Kbytes of the near data space, from 0000h to 07FFh, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '`0`'. A diagram of the SFR space, showing where SFRs are actually implemented, is shown in Table 4-2. Each implemented area indicates a 32-byte region where at least one address is implemented as an SFR. A complete listing of implemented SFRs, including their addresses, is shown in Tables 4-3 through 4-30.

TABLE 4-2: IMPLEMENTED REGIONS OF SFR DATA SPACE

	SFR Space Address							
	xx00	**xx20**	**xx40**	**xx60**	**xx80**	**xxA0**	**xxC0**	**xxE0**
000h	Core			ICN	Interrupts			—
100h	Timers		Capture			Compare		
200h	I^2C™	UART	SPI/UART	SPI/I^2C	SPI	UART	I/O	
300h	A/D	A/D/CTMU	—	—	—	—	—	—
400h	—	—	—	—	USB			—
500h	—	—	—	—	—	—	—	—
600h	PMP	RTC/Comp	CRC	—	PPS			—
700h	—	—	System	NVM/PMD	—	—	—	—

Legend: — = No implemented SFRs in this block

TABLE 6-1: RESET FLAG BIT OPERATION

Flag Bit	Setting Event	Clearing Event
TRAPR (RCON<15>)	Trap Conflict Event	POR
IOPUWR (RCON<14>)	Illegal Opcode or Uninitialized W Register Access	POR
CM (RCON<9>)	Configuration Mismatch Reset	POR
EXTR (RCON<7>)	MCLR Reset	POR
SWR (RCON<6>)	`RESET` Instruction	POR
WDTO (RCON<4>)	WDT Time-out	`PWRSAV` Instruction, POR
SLEEP (RCON<3>)	`PWRSAV #SLEEP` Instruction	POR
IDLE (RCON<2>)	`PWRSAV #IDLE` Instruction	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—

Note: All Reset flag bits may be set or cleared by the user software.

6.1 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen as shown in Table 6-2. If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. Refer to **Section 8.0 "Oscillator Configuration"** for further details.

TABLE 6-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)

Reset Type	Clock Source Determinant
POR	FNOSC Configuration bits (CW2<10:8>)
BOR	
MCLR	COSC Control bits (OSCCON<14:12>)
WDTO	
SWR	

6.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 6-3. Note that the system Reset signal, SYSRST, is released after the POR and PWRT delay times expire.

The time at which the device actually begins to execute code will also depend on the system oscillator delays, which include the Oscillator Start-up Timer (OST) and the PLL lock time. The OST and PLL lock times occur in parallel with the applicable SYSRST delay times.

The FSCM delay determines the time at which the FSCM begins to monitor the system clock source after the SYSRST signal is released.

REGISTER 7-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
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 **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 **T5IF:** Timer5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **T4IF:** Timer4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC4IF:** Output Compare Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC3IF:** Output Compare Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **Unimplemented:** Read as ‘0’

bit 7 **IC8IF:** Input Capture Channel 8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC7IF:** Input Capture Channel 7 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 **MI2C1IF:** Master I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SI2C1IF:** Slave I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-23: IPC6: INTERRUPT PRIORITY CONTROL REGISTER 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—
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 **T4IP<2:0>:** Timer4 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 **OC4IP<2:0>:** Output Compare 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 **OC3IP<2:0>:** Output Compare Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

9.2.2 IDLE MODE

Idle mode has these features:

- The CPU will stop executing instructions.
- The WDT is automatically cleared.
- The system clock source remains active. By default, all peripheral modules continue to operate normally from the system clock source, but can also be selectively disabled (see **Section 9.4 "Selective Peripheral Module Control"**).
- If the WDT or FSCM is enabled, the LPRC will also remain active.

The device will wake from Idle mode on any of these events:

- Any interrupt that is individually enabled.
- Any device Reset.
- A WDT time-out.

On wake-up from Idle, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction or the first instruction in the ISR.

9.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a PWRSAV instruction will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

9.3 Doze Mode

Generally, changing clock speed and invoking one of the power-saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a power-saving mode may stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV<11>). The ratio between peripheral and core clock speed is determined by the DOZE<2:0> bits (CLKDIV<14:12>). There are eight possible configurations, from 1:1 to 1:256, with 1:1 being the default.

It is also possible to use Doze mode to selectively reduce power consumption in event driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption while the CPU Idles, waiting for something to invoke an interrupt routine. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, interrupt events have no effect on Doze mode operation.

9.4 Selective Peripheral Module Control

Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume power. There may be cases where the application needs what these modes do not provide: the allocation of power resources to CPU processing with minimal power consumption from the peripherals.

PIC24F devices address this requirement by allowing peripheral modules to be selectively disabled, reducing or eliminating their power consumption. This can be done with two control bits:

- The Peripheral Enable bit, generically named, "XXXEN", located in the module's main control SFR.
- The Peripheral Module Disable (PMD) bit, generically named, "XXXMD", located in one of the PMD Control registers.

Both bits have similar functions in enabling or disabling their associated module. Setting the PMD bit for a module disables all clock sources to that module, reducing its power consumption to an absolute minimum. In this state, the control and status registers associated with the peripheral will also be disabled, so writes to those registers will have no effect and read values will be invalid. Many peripheral modules have a corresponding PMD bit.

In contrast, disabling a module by clearing its XXXEN bit disables its functionality, but leaves its registers available to be read and written to. This reduces power consumption, but not by as much as setting the PMD bit does. Most peripheral modules have an enable bit; exceptions include input capture, output compare and RTCC.

To achieve more selective power savings, peripheral modules can also be selectively disabled when the device enters Idle mode. This is done through the control bit of the generic name format, "XXXIDL". By default, all modules that can operate during Idle mode will do so. Using the disable on Idle feature allows further reduction of power consumption during Idle mode, enhancing power savings for extremely critical power applications.

10.3 Input Change Notification

The input change notification function of the I/O ports allows the PIC24FJ256GB110 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 81 external inputs that may be selected (enabled) for generating an interrupt request on a Change-Of-State.

Registers, CNEN1 through CNEN6, contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin has a both a weak pull-up and a weak pull-down connected to it. The pull-ups act as a current source that is connected to the pin, while the pull-downs act as a current sink that is connected to the pin. These eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups and pull-downs are separately enabled using the CNPU1 through CNPU6 registers (for pull-ups) and the CNPD1 through CNPD6 registers (for pull-downs). Each CN pin has individual control bits for its pull-up and pull-down. Setting a control bit enables the weak pull-up or pull-down for the corresponding pin.

When the internal pull-up is selected, the pin pulls up to V_{DD} – 0.7V (typical). Make sure that there is no external pull-up source when the internal pull-ups are enabled, as the voltage difference can cause a current path.

Note:	Pull-ups on change notification pins should always be disabled whenever the port pin is configured as a digital output.

10.4 Peripheral Pin Select

A major challenge in general purpose devices is providing the largest possible set of peripheral features while minimizing the conflict of features on I/O pins. In an application that needs to use more than one peripheral multiplexed on a single pin, inconvenient workarounds in application code or a complete redesign may be the only option.

The Peripheral Pin Select (PPS) feature provides an alternative to these choices by enabling the user's peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of any one of many digital peripherals to any one of these I/O pins. Peripheral Pin Select is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

10.4.1 AVAILABLE PINS

The Peripheral Pin Select feature is used with a range of up to 44 pins, depending on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the designation, "RPn" or "RPIn", in their full pin designation, where "n" is the remappable pin number. "RP" is used to designate pins that support both remappable input and output functions, while "RPI" indicates pins that support remappable input functions only.

PIC24FJ256GB110 family devices support a larger number of remappable input only pins than remappable input/output pins. In this device family, there are up to 32 remappable input/output pins, depending on the pin count of the particular device selected; these are numbered, RP0 through RP31. Remappable input only pins are numbered above this range, from RPI32 to RPI43 (or the upper limit for that particular device).

See Table 1-4 for a summary of pinout options in each package offering.

REGISTER 10-11: RPINR15: PERIPHERAL PIN SELECT INPUT REGISTER 15

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **IC9R<5:0>:** Assign Input Capture 9 (IC9) to Corresponding RPn or RPIn Pin bits

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

REGISTER 10-12: RPINR17: PERIPHERAL PIN SELECT INPUT REGISTER 17

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

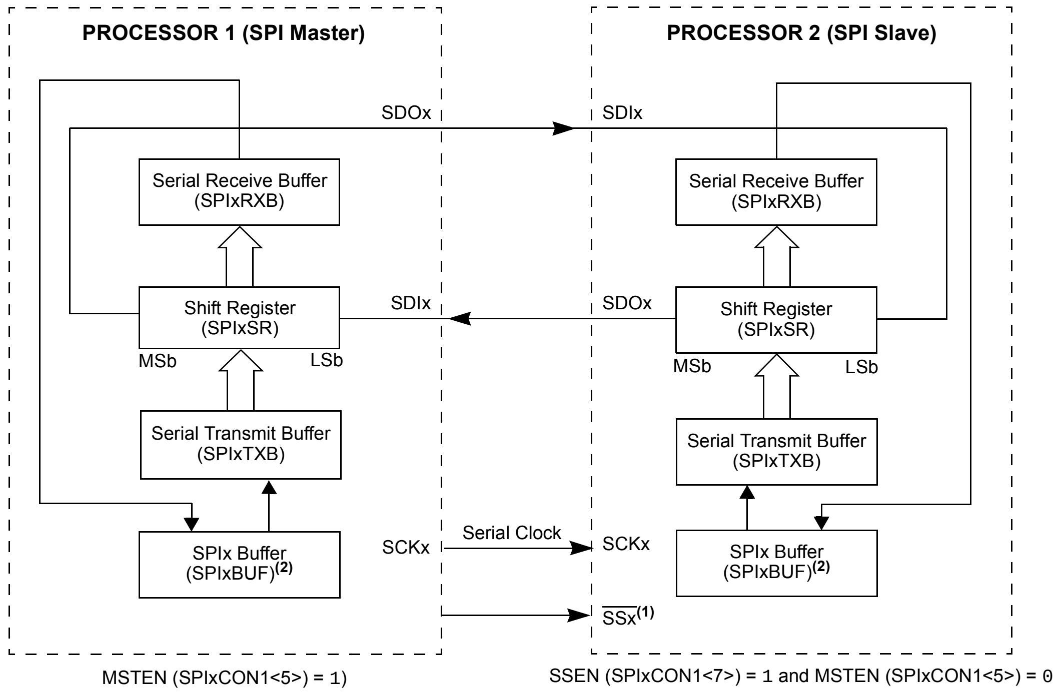
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **U3RXR<5:0>:** Assign UART3 Receive (U3RX) to Corresponding RPn or RPIn Pin bits

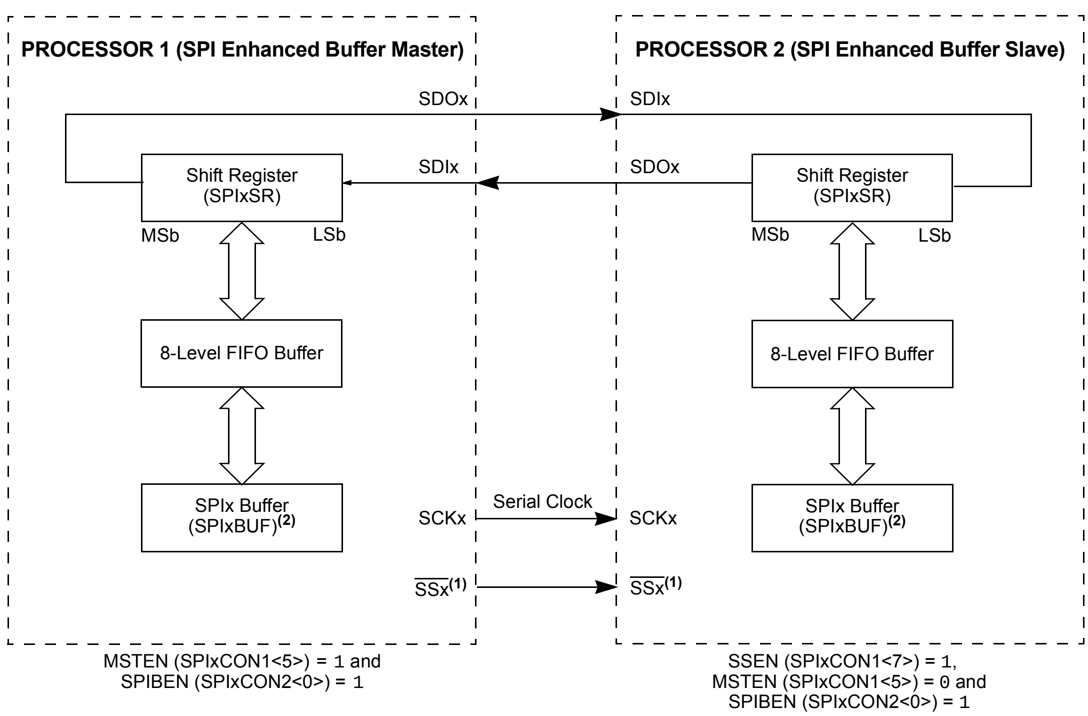
bit 7-0 **Unimplemented:** Read as ‘0’

FIGURE 15-3: SPI MASTER/SLAVE CONNECTION (STANDARD MODE)



Note 1: Using the $\overline{SSx}$ pin in Slave mode of operation is optional.
2: User must write transmit data to read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

FIGURE 15-4: SPI MASTER/SLAVE CONNECTION (ENHANCED BUFFER MODES)



Note 1: Using the $\overline{SSx}$ pin in Slave mode of operation is optional.
2: User must write transmit data to read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

16.0 INTER-INTEGRATED CIRCUIT (I^2C™)

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 *"PIC24F Family Reference Manual"*, **Section 24. "Inter-Integrated Circuit (I^2C™)"** (DS39702).

The Inter-Integrated Circuit (I^2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

The I^2C module supports these features:

- Independent master and slave logic
- 7-bit and 10-bit device addresses
- General call address, as defined in the I^2C protocol
- Clock stretching to provide delays for the processor to respond to a slave data request
- Both 100 kHz and 400 kHz bus specifications.
- Configurable address masking
- Multi-Master modes to prevent loss of messages in arbitration
- Bus Repeater mode, allowing the acceptance of all messages as a slave regardless of the address
- Automatic SCL

A block diagram of the module is shown in Figure 16-1.

16.1 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communications protocol for the device being communicated with. Typically, the sequence of events is as follows:

1. Assert a Start condition on SDAx and SCLx.
2. Send the I^2C device address byte to the slave with a write indication.
3. Wait for and verify an Acknowledge from the slave.
4. Send the first data byte (sometimes known as the command) to the slave.
5. Wait for and verify an Acknowledge from the slave.
6. Send the serial memory address low byte to the slave.
7. Repeat steps 4 and 5 until all data bytes are sent.
8. Assert a Repeated Start condition on SDAx and SCLx.
9. Send the device address byte to the slave with a read indication.
10. Wait for and verify an Acknowledge from the slave.
11. Enable master reception to receive serial memory data.
12. Generate an ACK or NACK condition at the end of a received byte of data.
13. Generate a Stop condition on SDAx and SCLx.

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5	**ACKDT:** Acknowledge Data bit (when operating as I^2C master. Applicable during master receive.) Value that will be transmitted when the software initiates an Acknowledge sequence. 1 = Sends NACK during Acknowledge 0 = Sends ACK during Acknowledge
bit 4	**ACKEN:** Acknowledge Sequence Enable bit (When operating as I^2C master. Applicable during master receive.) 1 = Initiates Acknowledge sequence on SDAx and SCLx pins and transmits ACKDT data bit. Hardware clear at end of master Acknowledge sequence. 0 = Acknowledge sequence not in progress
bit 3	**RCEN:** Receive Enable bit (when operating as I^2C master) 1 = Enables Receive mode for I^2C. Hardware clear at end of eighth bit of master receive data byte. 0 = Receives sequence not in progress
bit 2	**PEN:** Stop Condition Enable bit (when operating as I^2C master) 1 = Initiates Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence. 0 = Stop condition not in progress
bit 1	**RSEN:** Repeated Start Condition Enabled bit (when operating as I^2C master) 1 = Initiates Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of master Repeated Start sequence. 0 = Repeated Start condition not in progress
bit 0	**SEN:** Start Condition Enabled bit (when operating as I^2C master) 1 = Initiates Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence. 0 = Start condition not in progress

18.1 Hardware Configuration

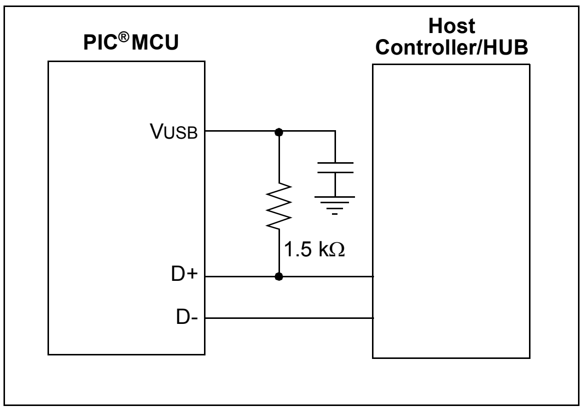
18.1.1 DEVICE MODE

18.1.1.1 D+ Pull-up Resistor

PIC24FJ256GB110 family devices have a built-in 1.5 kΩ resistor on the D+ line that is available when the microcontroller in operating in device mode. This is used to signal an external Host that the device is operating in Full Speed Device mode. It is engaged by setting the DPPULUP bit (U1OTGCON<7>).

Alternatively, an external resistor may be used on D+, as shown in Figure 18-2.

FIGURE 18-2: EXTERNAL PULL-UP FOR FULL-SPEED DEVICE MODE



18.1.1.2 Power Modes

Many USB applications will likely have several different sets of power requirements and configuration. The most common power modes encountered are:

- Bus Power Only,
- Self-Power Only and
- Dual Power with Self-Power Dominance.

Bus Power Only mode (Figure 18-3) is effectively the simplest method. All power for the application is drawn from the USB.

To meet the inrush current requirements of the USB 2.0 Specification, the total effective capacitance appearing across VBUS and ground must be no more than 10 μF.

In the USB Suspend mode, devices must consume no more than 2.5 mA from the 5V VBUS line of the USB cable. During the USB Suspend mode, the D+ or D- pull-up resistor must remain active, which will consume some of the allowed suspend current.

In Self-Power Only mode (Figure 18-4), the USB application provides its own power, with very little power being pulled from the USB. Note that an attach indication is added to indicate when the USB has been connected and the host is actively powering VBUS.

To meet compliance specifications, the USB module (and the D+ or D- pull-up resistor) should not be enabled until the host actively drives VBUS high. One of the 5.5V tolerant I/O pins may be used for this purpose.

The application should never source any current onto the 5V VBUS pin of the USB cable.

The Dual-power option with Self-Power Dominance (Figure 18-5) allows the application to use internal power primarily, but switch to power from the USB when no internal power is available. Dual-power devices must also meet all of the special requirements for inrush current and Suspend mode current previously described, and must not enable the USB module until VBUS is driven high.

FIGURE 18-3: BUS POWER ONLY

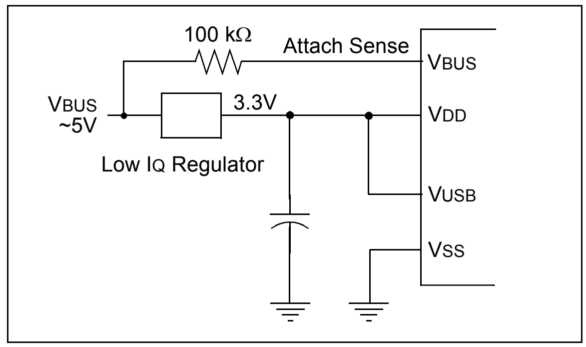


FIGURE 18-4: SELF-POWER ONLY

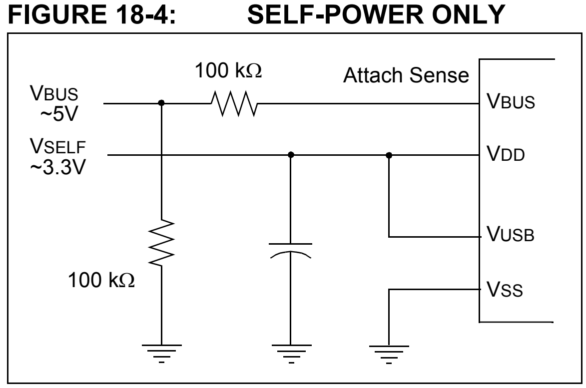
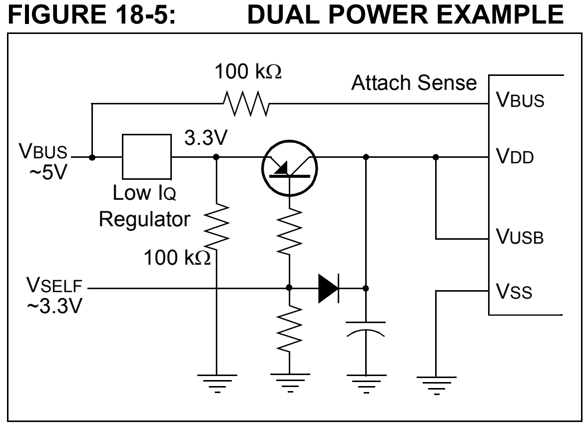


FIGURE 18-5: DUAL POWER EXAMPLE



18.1.2 HOST AND OTG MODES

18.1.2.1 D+ and D- Pull-down Resistors

PIC24FJ256GB110 family devices have built-in 15 kΩ pull-down resistor on the D+ and D- lines. These are used in tandem to signal to the bus that the microcontroller is operating in Host mode. They are engaged by setting the DPPULDWN and DMPULDWN bits (U1OTGCON<5,4>).

18.1.2.2 Power Configurations

In Host mode, as well as Host mode in On-the-Go operation, the USB 2.0 specification requires that the Host application supply power on VBUS. Since the microcontroller is running below VBUS and is not able to source sufficient current, a separate power supply must be provided.

When the application is always operating in Host mode, a simple circuit can be used to supply VBUS and regulate current on the bus (Figure 18-6). For OTG operation, it is necessary to be able to turn VBUS on or off as needed, as the microcontroller switches between Device and Host modes. A typical example using an external charge pump is shown in Figure 18-7.

FIGURE 18-6: HOST INTERFACE EXAMPLE

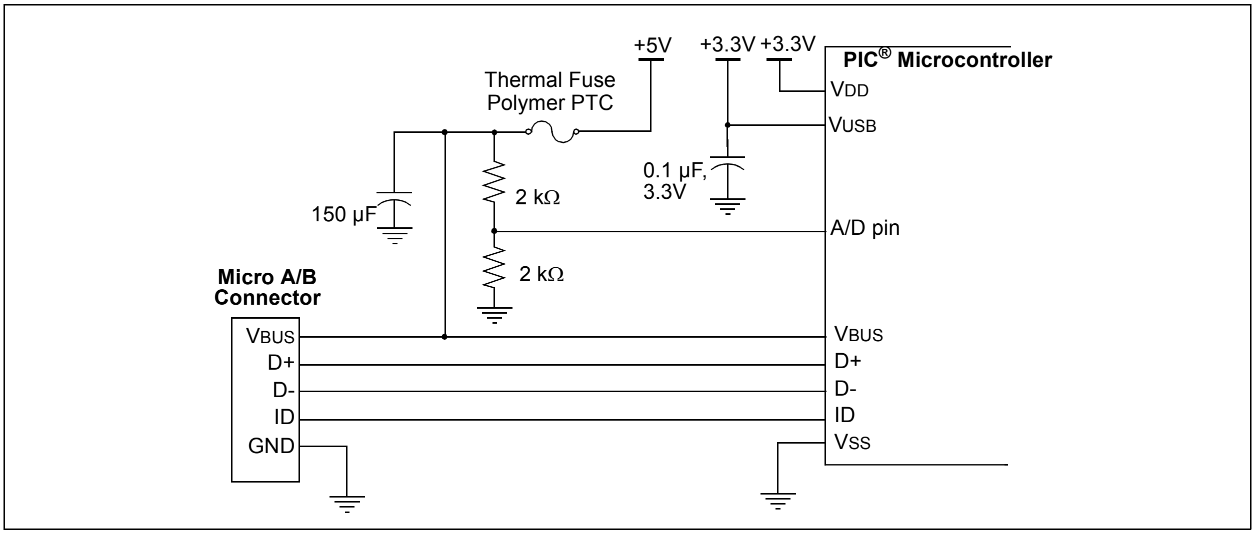
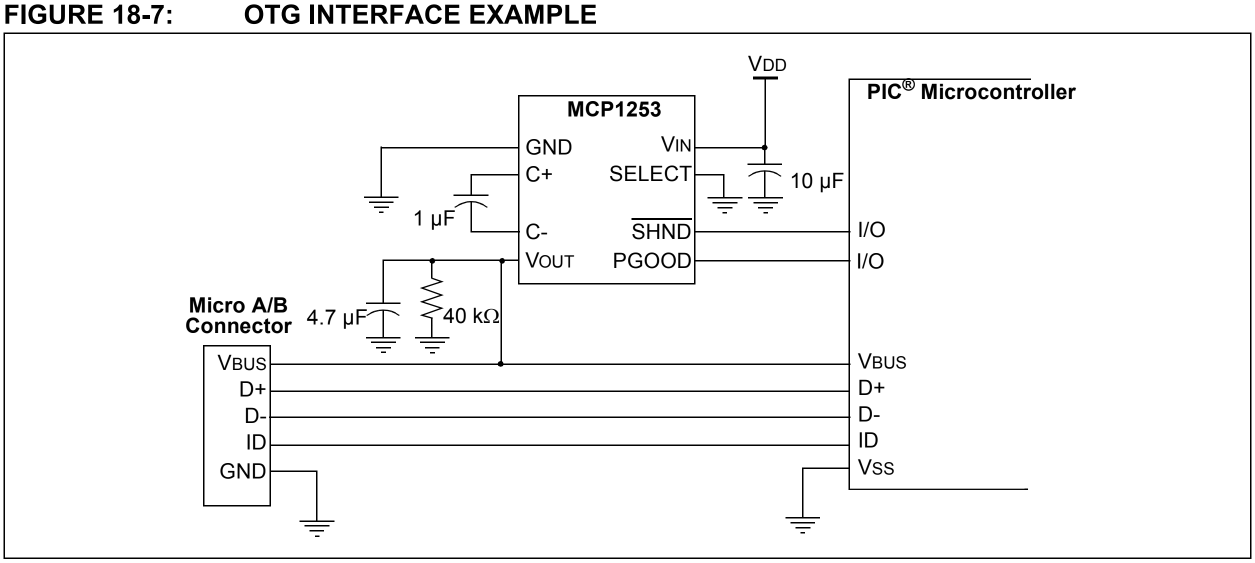


FIGURE 18-7: OTG INTERFACE EXAMPLE



18.1.2.3 VBUS Voltage Generation with External Devices

When operating as a USB host, either as an A-device in an OTG configuration or as an embedded host, VBUS must be supplied to the attached device. PIC24FJ256GB110 family devices have an internal VBUS boost assist to help generate the required 5V VBUS from the available voltages on the board. This is comprised of a simple PWM output to control a Switch mode power supply, and built-in comparators to monitor output voltage and limit current.

To enable voltage generation:

1. Verify that the USB module is powered (U1PWRC<0> = `1`) and that the VBUS discharge is disabled (U1OTGCON<0> = `0`).
2. Set the PWM period (U1PWMRRS<7:0>) and duty cycle (U1PWMRRS<15:8>) as required.
3. Select the required polarity of the output signal based on the configuration of the external circuit with the PWMPOL bit (U1PWMCON<9>).
4. Select the desired target voltage using the VBUSCHG bit (U1OTGCON<1>).
5. Enable the PWM counter by setting the CNTEN bit to ‘`1`’ (U1PWMCON<8>).
6. Enable the PWM module by setting the PWMEN bit to ‘`1`’ (U1PWMCON<15>).
7. Enable the VBUS generation circuit (U1OTGCON<3> = `1`).

Note:	This section describes the general process for VBUS voltage generation and control. Please refer to the “*PIC24F Family Reference Manual”* for additional examples.

18.1.3 USING AN EXTERNAL INTERFACE

Some applications may require the USB interface to be isolated from the rest of the system. PIC24FJ256GB110 family devices include a complete interface to communicate with and control an external USB transceiver, including the control of data line pull-ups and pull-downs. The VBUS voltage generation control circuit can also be configured for different VBUS generation topologies.

Please refer to the *“PIC24F Family Reference Manual”*, **Section 27. “USB On-The-Go (OTG)”** for information on using the external interface.

18.1.4 CALCULATING TRANSCEIVER POWER REQUIREMENTS

The USB transceiver consumes a variable amount of current depending on the characteristic impedance of the USB cable, the length of the cable, the VUSB supply voltage and the actual data patterns moving across the USB cable. Longer cables have larger capacitances and consume more total energy when switching output states. The total transceiver current consumption will be application-specific. Equation 18-1 can help estimate how much current actually may be required in full-speed applications.

Please refer to the *“PIC24F Family Reference Manual”*, **Section 27. “USB On-The-Go (OTG)”** for a complete discussion on transceiver power consumption.

EQUATION 18-1: ESTIMATING USB TRANSCEIVER CURRENT CONSUMPTION

$$I_{XCVR} = \frac{(40\text{ mA} \bullet V_{USB} \bullet P_{ZERO} \bullet P_{IN} \bullet L_{CABLE})}{(3.3V \bullet 5m)} + I_{PULLUP}$$

Legend: VUSB – Voltage applied to the VUSB pin in volts (3.0V to 3.6V).

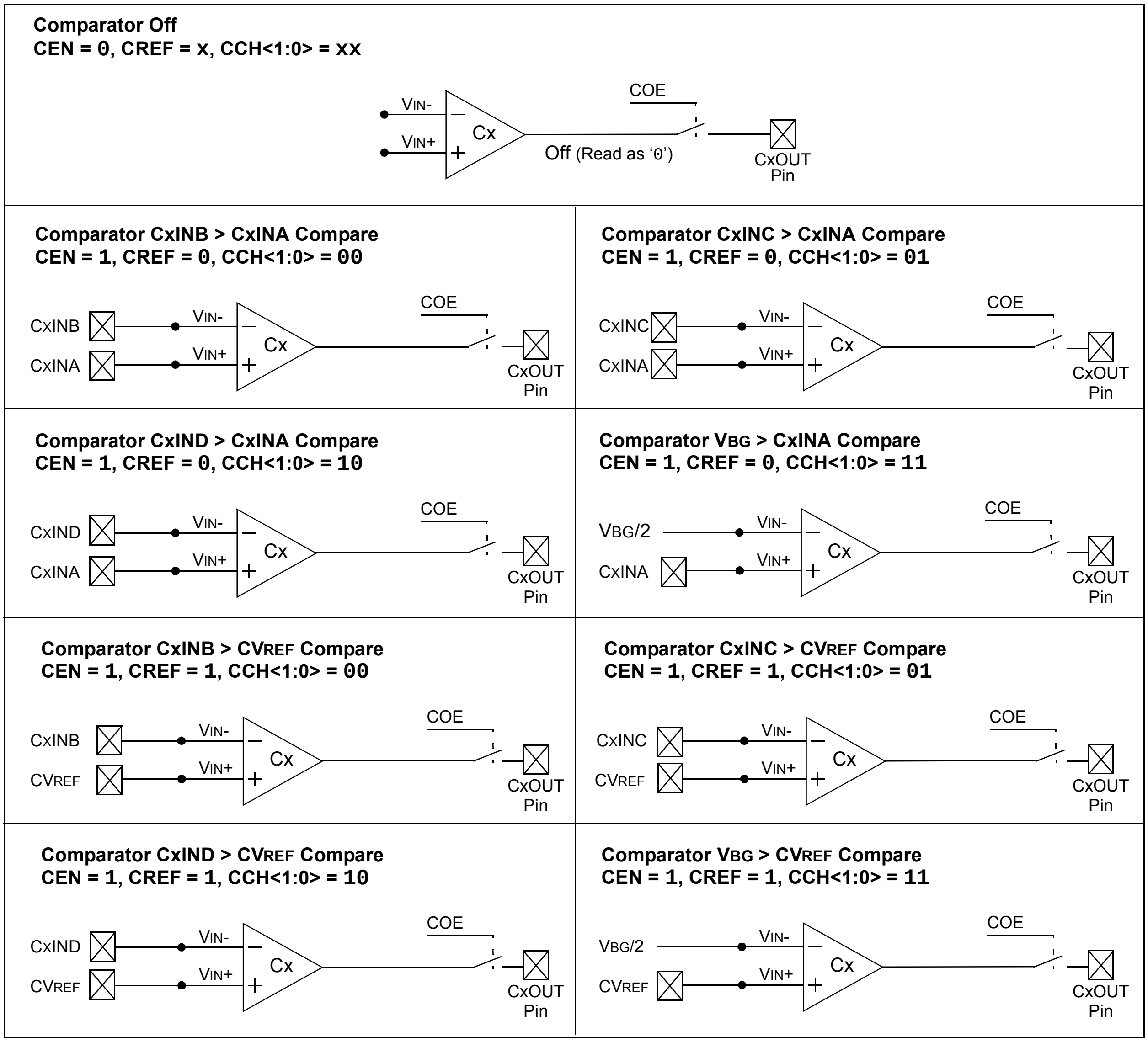
PZERO – Percentage (in decimal) of the IN traffic bits sent by the PIC® microcontroller that are a value of ‘`0`’.

PIN – Percentage (in decimal) of total bus bandwidth that is used for IN traffic.

LCABLE – Length (in meters) of the USB cable. The USB 2.0 Specification requires that full-speed applications use cables no longer than 5m.

IPULLUP – Current which the nominal, 1.5 kΩ pull-up resistor (when enabled) must supply to the USB cable.

FIGURE 23-2: INDIVIDUAL COMPARATOR CONFIGURATIONS



REGISTER 25-1: CTMUCON: CTMU CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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
EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12 **TGEN:** Time Generation Enable bit(1)
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 10 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 10 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 9 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 8 **CTTRIG:** Trigger Control bit
1 = Trigger output is enabled
0 = Trigger output is disabled

bit 7 **EDG2POL:** Edge 2 Polarity Select bit
1 = Edge 2 programmed for a positive edge response
0 = Edge 2 programmed for a negative edge response

bit 6-5 **EDG2SEL<1:0>:** Edge 2 Source Select bits
11 = CTED1 pin
10 = CTED2 pin
01 = OC1 module
00 = Timer1 module

bit 4 **EDG1POL:** Edge 1 Polarity Select bit
1 = Edge 1 programmed for a positive edge response
0 = Edge 1 programmed for a negative edge response

Note 1: If TGEN = 1, the CTEDGx inputs and CTPLS outputs must be assigned to available RPn pins before use. See **Section 10.4 “Peripheral Pin Select”** for more information.

27.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

27.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

27.4 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

27.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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

27.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C 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 IDE compatibility

TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
ULNK	ULNK		Unlink Frame Pointer	1	1	None
XOR	XOR	f	f = f .XOR. WREG	1	1	N, Z
	XOR	f,WREG	WREG = f .XOR. WREG	1	1	N, Z
	XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N, Z
	XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N, Z
	XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N, Z
ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C, Z, N

NOTES:

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent ________

RE: Reader Response

From: Name ____

Company ____

Address ____

City / State / ZIP / Country ______________________________

Telephone: (_______) _________ - _________ FAX: (______) _________ - _________

Application (optional):

Would you like a reply?____Y ____N

Device: PIC24FJ256GB110 Family Literature Number: DS39897C

Questions:

1. What are the best features of this document?

____

____

2. How does this document meet your hardware and software development needs?

____

____

3. Do you find the organization of this document easy to follow? If not, why?

____

____

4. What additions to the document do you think would enhance the structure and subject?

____

____

5. What deletions from the document could be made without affecting the overall usefulness?

____

____

6. Is there any incorrect or misleading information (what and where)?

____

____

7. How would you improve this document?

____

____

