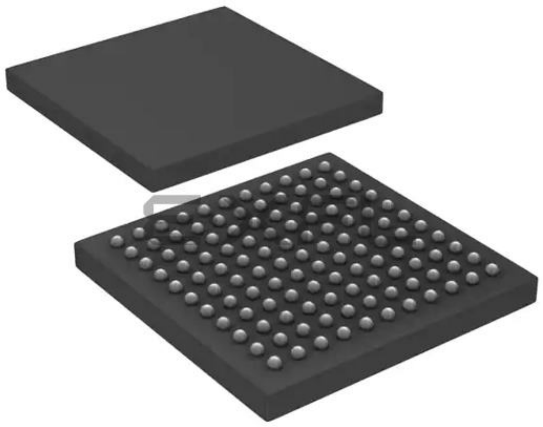
Microchip Technology - PIC24FJ128GC010T-I/BG Datasheet



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, IrDA, LINbus, PMP, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, LCD, LVD, POR, PWM, WDT
Number of I/O	85
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 50x12b, 2x16b; D/A 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	121-TFBGA
Supplier Device Package	121-TFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128gc010t-i-bg

3.2 CPU Control Registers

REGISTER 3-1: SR: ALU STATUS REGISTER[1]

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[2]	R/W-0[2]	R/W-0[2]	R-0	R/W-0	R/W-0	R/W-0,	R/W-0
IPL2[3]	IPL1[3]	IPL0[3]	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 (IPL) Status bits[2,3] 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 is in progress 0 = REPEAT loop is not in progress
bit 3	**N:** ALU Negative bit 1 = Result was negative 0 = Result was not 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 resulted in the ALU having a value of zero. 0 = An operation resulted in the ALU having a non-zero value.
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 of the result occurred

Note 1: ALU result flags are not affected for every operation. See Table 36-2 for details.
2: The IPLx Status bits are read-only when NSTDIS (INTCON1<15>) = 1.
3: The IPLx Status bits are concatenated with the IPL3 (CORCON<3>) bit to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

TABLE 4-25: 12-BIT PIPELINE A/D CONVERTER REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADTBL0	0300	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL1	0302	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL2	0304	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL3	0306	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL4	0308	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL5	030A	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL6	030C	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL7	030E	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL8	0310	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL9	0312	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL10	0314	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL11	0316	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL12	0318	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL13	031A	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL14	031C	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL15	031E	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL16	0320	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL17	0322	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL18	0324	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL19	0326	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL20	0328	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL21	032A	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL22	032C	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL23	032E	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL24	0330	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL25	0332	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL26	0334	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL27	0336	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL28	0338	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL29	033A	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL30	033C	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000
ADTBL31	033E	UCTMU	DIFF	—	—	—	—	—	—	—	ADCH6	ADCH5	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	0000

Legend: — = unimplemented, read as '0'; r = reserved, do not modify. Reset values are shown in hexadecimal.

5.0 DIRECT MEMORY ACCESS CONTROLLER (DMA)

Note: This data sheet summarizes the features of the PIC24FJ128GC010 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Direct Memory Access Controller (DMA)"** (DS39742) which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM.

The Direct Memory Access (DMA) controller is designed to service high data throughput peripherals operating on the SFR bus, allowing them to access data memory directly and alleviating the need for CPU-intensive management. By allowing these data-intensive peripherals to share their own data path, the main data bus is also deloaded, resulting in additional power savings.

The DMA controller functions both as a peripheral and a direct extension of the CPU. It is located on the microcontroller data bus, between the CPU and DMA controller-enabled peripherals, with direct access to SRAM. This partitions the SFR bus into two buses, allowing the DMA controller access to the DMA capable peripherals located on the new DMA SFR bus. The controller serves as a master device on the DMA SFR bus, controlling data flow from DMA capable peripherals.

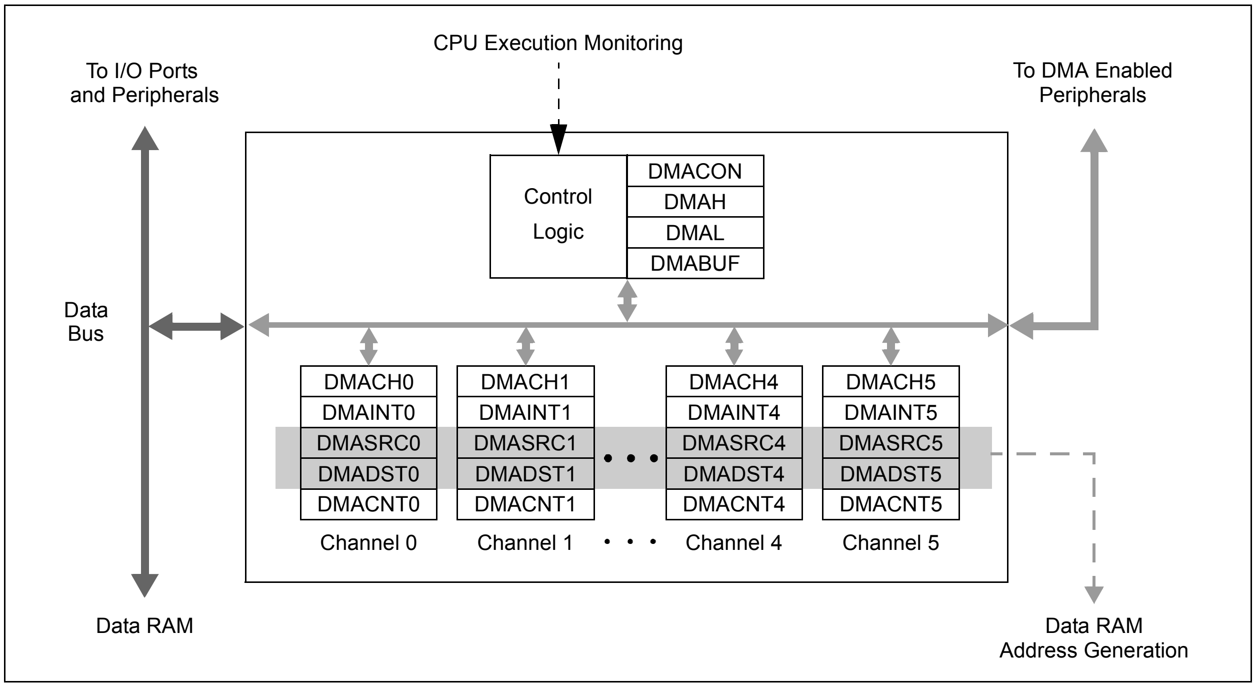
The controller also monitors CPU instruction processing directly, allowing it to be aware of when the CPU requires access to peripherals on the DMA bus, and automatically relinquishing control to the CPU as needed. The use of DMA increases the time the processor can execute code while the DMA is transferring data.

The DMA controller includes these features:

- Six Multiple Independent and Independently Programmable Channels
- Concurrent Operation with the CPU (no DMA caused Wait states)
- DMA Bus Arbitration
- Five Programmable Address modes
- Four Programmable Transfer modes
- Four Flexible Internal Data Transfer modes
- Byte or Word Support for Data Transfer
- 16-Bit Source and Destination Address Register for Each Channel, Dynamically Updated and Reloadable
- 16-Bit Transaction Count Register, Dynamically Updated and Reloadable
- Upper and Lower Address Limit Registers
- Counter Half-Full Level Interrupt
- Software Triggered Transfer
- Null Write mode for Symmetric Buffer Operations

A simplified block diagram of the DMA controller is shown if Figure 5-1.

FIGURE 5-1: DMA CONTROLLER FUNCTIONAL BLOCK DIAGRAM



EXAMPLE 6-2: ERASING A PROGRAM MEMORY BLOCK ('C' LANGUAGE CODE)

```
// C example
   unsigned long progAddr = 0x6000;             // Address of row to write
   unsigned int offset;
//Set up pointer to the first memory location to be written
   TBLPAG = progAddr>>16;                       // Initialize PM Page Boundary SFR
   offset = progAddr & 0xFFFF;                  // Initialize lower word of address
   __builtin_tblwtl(offset, 0x0000);            // Set base address of erase block
                                                // with dummy latch write
   NVMCON = 0x4042;                             // Initialize NVMCON
   asm("DISI #5");                              // Block all interrupts with priority <7
                                                // for next 5 instructions
   __builtin_write_NVM();                       // check function to perform unlock
                                                // sequence and set WR
```

EXAMPLE 6-3: LOADING THE WRITE BUFFERS

```
; Set up NVMCON for row programming operations
       MOV     #0x4001, W0                     ;
       MOV     W0, NVMCON                      ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0                     ;
       MOV     W0, TBLPAG                      ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0                     ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0, W2                 ;
       MOV     #HIGH_BYTE_0, W3                ;
       TBLWTL  W2, [W0]                        ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                      ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1, W2                 ;
       MOV     #HIGH_BYTE_1, W3                ;
       TBLWTL  W2, [W0]                        ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                      ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2, W2                 ;
       MOV     #HIGH_BYTE_2, W3                ;
       TBLWTL  W2, [W0]                        ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                      ; Write PM high byte into program latch
       •
       •
       •
; 63rd_program_word
       MOV     #LOW_WORD_63, W2                ;
       MOV     #HIGH_BYTE_63, W3               ;
       TBLWTL  W2, [W0]                        ; Write PM low word into program latch
       TBLWTH  W3, [W0]                        ; Write PM high byte into program latch
```

EXAMPLE 6-4: INITIATING A PROGRAMMING SEQUENCE

```
       DISI    #5                       ; Block all interrupts with priority <7
                                        ; for next 5 instructions
       MOV.B   #0x55, W0
       MOV     W0, NVMKEY               ; Write the 0x55 key
       MOV.B   #0xAA, W1                ;
       MOV     W1, NVMKEY               ; Write the 0xAA key
       BSET    NVMCON, #WR              ; Start the programming sequence
       NOP                              ; Required delays
       NOP
       BTSC    NVMCON, #15              ; and wait for it to be
       BRA     $-2                      ; completed
```

TABLE 8-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
A/D (12-Bit Pipeline)	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
A/D (Sigma-Delta)	105	0000E6h	0001E6h	IFS6<9>	IEC6<9>	IPC26<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU Event	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
DAC1	78	000080h	000180h	IFS4<14>	IEC4<14>	IPC19<10:8>
DAC2	79	000082h	000182h	IFS4<15>	IEC4<15>	IPC19<14:12>
DMA Channel 0	4	00001Ch	00011Ch	IFS0<4>	IEC0<4>	IPC1<2:0>
DMA Channel 1	14	000030h	000130h	IFS0<14>	IEC0<14>	IPC3<10:8>
DMA Channel 2	24	000044h	000144h	IFS1<8>	IEC1<8>	IPC6<2:0>
DMA Channel 3	36	00005Ch	00015Ch	IFS2<4>	IEC2<4>	IPC9<2:0>
DMA Channel 4	46	000070h	000170h	IFS2<14>	IEC2<14>	IPC11<10:8>
DMA Channel 5	61	00008Eh	00018Eh	IFS3<13>	IEC3<13>	IPC15<6:4>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
External Interrupt 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
External Interrupt 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
FRC Self-Tune	106	0000E8h	0001E8h	IFS6<10>	IEC6<10>	IPC26<10:8>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Capture 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
Input Capture 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
Input Capture 8	23	000042h	000142h	IFS1<7>	IEC1<7>	IPC5<14:12>
Input Capture 9	93	0000CEh	0001CEh	IFS5<13>	IEC5<13>	IPC23<6:4>
JTAG	117	0000FEh	0001FEh	IFS7<5>	IEC7<5>	IPC29<6:4>
Input Change Notification (ICN)	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
LCD Controller	100	0000DCh	0001DCh	IFS6<4>	IEC6<4>	IPC25<2:0>
High/Low-Voltage Detect (HLVD)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
Op Amp 1	103	0000E2h	0001E2h	IFS6<7>	IEC6<7>	IPC25<14:12>
Op Amp 2	104	0000E4h	0001E4h	IFS6<8>	IEC6<8>	IPC26<2:0>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Output Compare 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
Output Compare 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
Output Compare 8	44	00006Ch	00016Ch	IFS2<12>	IEC2<12>	IPC11<2:0>
Output Compare 9	92	0000CCh	0001CCh	IFS5<12>	IEC5<12>	IPC23<2:0>

REGISTER 8-17: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
DAC2IE	DAC1IE	CTMUIE	—	—	—	—	HLVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
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	**DAC2IE:** DAC Converter 2 Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 14	**DAC1IE:** DAC Converter 1 Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 13	**CTMUIE:** CTMU Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 12-9	**Unimplemented:** Read as ‘0’
bit 8	**HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 7-4	**Unimplemented:** Read as ‘0’
bit 3	**CRCIE:** CRC Generator Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 2	**U2ERIE:** UART2 Error Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 1	**U1ERIE:** UART1 Error Interrupt Enable bit 1 = Interrupt request is enabled 0 = Interrupt request is not enabled
bit 0	**Unimplemented:** Read as ‘0’

REGISTER 8-43: IPC25: INTERRUPT PRIORITY CONTROL REGISTER 25

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	AMP1IP2	AMP1IP1	AMP1IP0	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	LCDIP2	LCDIP1	LCDIP0
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	**AMP1IP<2:0>:** Op Amp 1 Interrupt Priority bits 111 = Interrupt is Priority 7 (highest priority interrupt) • • • 001 = Interrupt is Priority 1 000 = Interrupt source is disabled
bit 11-3	**Unimplemented:** Read as ‘0’
bit 2-0	**LCDIP<2:0>:** LCD Controller Interrupt Priority bits 111 = Interrupt is Priority 7 (highest priority interrupt) • • • 001 = Interrupt is Priority 1 000 = Interrupt source is disabled

REGISTER 8-44: IPC26: INTERRUPT PRIORITY CONTROL REGISTER 26

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	FSTIP2	FSTIP1	FSTIP0
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
—	SDA1IP2	SDA1IP1	SDA1IP0	—	AMP2IP2	AMP2IP1	AMP2IP0
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	**FSTIP<2:0>:** FRC Self-Tune 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	**SDA1IP<2:0>:** Sigma-Delta A/D Converter 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	**AMP2IP<2:0>:** Op Amp 2 Interrupt Priority bits 111 = Interrupt is Priority 7 (highest priority interrupt) • • • 001 = Interrupt is Priority 1 000 = Interrupt source is disabled

9.6.1 CONSIDERATIONS FOR USB OPERATION

When using the USB On-The-Go module in PIC24FJ128GC010 family devices, users must always observe these rules in configuring the system clock:

- The oscillator modes listed in Table 9-3 are the only oscillator configurations that permit USB operation. There is no provision to provide a separate external clock source to the USB module.
- For USB operation, the selected clock source (EC, HS or XT) must meet the USB clock tolerance requirements.
- When the FRCPLL Oscillator mode is used for USB applications, the FRC self-tune system should be used as well. While the FRC is accurate, the only two ways to ensure the level of accuracy required by the *"USB 2.0 Specification",* throughout the application's operating range, are either the self-tune system or manually changing the TUN<5:0> bits.
- The user must always ensure that the FRC source is configured to provide a frequency of 4 MHz or 8 MHz (RCDIV<2:0> = `001` or `000`) and that the USB PLL prescaler is configured appropriately.
- All other oscillator modes are available; however, USB operation is not possible when these modes are selected. They may still be useful in cases where other power levels of operation are desirable and the USB module is not needed (e.g., the application is Sleeping and waiting for a bus attachment).

9.7 Reference Clock Output

In addition to the CLKO output (FOSC/2) available in certain oscillator modes, the device clock in the PIC24FJ128GC010 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 9-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV<3:0> bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the Primary modes (EC, HS or XT); otherwise, if the POSCEN bit is not also set, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

9.8 Secondary Oscillator

9.8.1 BASIC SOSC OPERATION

PIC24FJ128GC010 family devices do not have to set the SOSCEN bit to use the Secondary Oscillator. Any module requiring the SOSC (such as RTCC, Timer1 or DSWDT) will automatically turn on the SOSC when the clock signal is needed. The SOSC, however, has a long start-up time (as long as 1 second).To avoid delays for peripheral start-up, the SOSC can be manually started using the SOSCEN bit.

To use the Secondary Oscillator, the SOSCSEL bit (CW3<8>) must be set to '`1`'. Programming the SOSCSEL bit to '`0`' configures the SOSC pins for Digital mode, enabling digital I/O functionality on the pins.

9.8.2 CRYSTAL SELECTION

The 32.768 kHz crystal used for the SOSC must have the following specifications in order to properly start up and run at the correct frequency:

- 12.5 pF loading capacitance
- 1.0 pF shunt capacitance
- A typical ESR of 50K; 70K maximum

In addition, the two external crystal loading capacitors should be in the range of 22-27 pF, which will be based on the PC board layout. The capacitors should be C0G, 5% tolerance and rated 25V or greater.

The accuracy and duty cycle of the SOSC can be measured on the REFO pin and is recommended to be in the range of 40-60% and accurate to ±0.65Hz.

Note: Do not enable the LCD Segment pin, SEG17, on RD0 when using the 64-pin package if the SOSC is used for time-sensitive applications. Avoid high-frequency traces adjacent to the SOSCO and SOSCI pins as this can cause errors in the SOSC frequency and/or duty cycle.

10.1.1 INSTRUCTION-BASED POWER-SAVING MODES

Three of the power-saving modes are entered through the execution of the PWRSAV instruction. Sleep mode stops clock operation and halts all code execution. Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. Deep Sleep mode stops clock operation, code execution and all peripherals, except RTCC and DSWDT. It also freezes I/O states and removes power to Flash memory, and may remove power to SRAM.

The assembly syntax of the PWRSAV instruction is shown in Example 10-1. Sleep and Idle modes are entered directly with a single assembler command. Deep Sleep requires an additional sequence to unlock and enable the entry into Deep Sleep, which is described in **Section 10.4.2 "Entering Deep Sleep Mode"**.

Note:	SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

When using the MPLAB® C compilers, there are two special power-saving instructions:

- Sleep();
- Idle();

These built-in functions are equivalent to the PWRSAV assembly instructions.

The features enabled with the low-voltage/retention regulator result in some changes to the way that Sleep and Deep Sleep modes behave. See **Section 10.3 "Sleep Mode"** and **Section 10.4 "Deep Sleep Mode"** for additional information.

10.1.1.1 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.

For Deep Sleep mode, interrupts that coincide with the execution of the PWRSAV instruction may be lost. If the low-voltage/retention regulator is not enabled, the microcontroller resets on leaving Deep Sleep and the interrupt will be lost. If the low-voltage/retention regulator is enabled, the microcontroller will exit Deep Sleep and the interrupt will then be handled.

Interrupts that occur during the Deep Sleep unlock sequence will interrupt the mandatory five-instruction cycle sequence timing and cause a failure to enter Deep Sleep. For this reason, it is recommended to disable all interrupts during the Deep Sleep unlock sequence.

EXAMPLE 10-1: PWRSAV INSTRUCTION SYNTAX

```
// Syntax to enter Sleep mode:
PWRSAV     #SLEEP_MODE       ; Put the device into SLEEP mode
//
//Synatx to enter Idle mode:
PWRSAV     #IDLE_MODE        ; Put the device into IDLE mode
//
// Syntax to enter Deep Sleep mode:
// First use the unlock sequence to set the DSEN bit (see Example 10-2)
BSET       DSCON, #DSEN      ;Enable Deep Sleep
BSET       DSCON, #DSEN      ; Enable Deep Sleep(repeat the command)
PWRSAV     #SLEEP_MODE       ; Put the device into Deep SLEEP mode
```

10.1.2 HARDWARE-BASED POWER-SAVING MODE

The hardware-based VBAT mode does not require any action by the user during code development. Instead, it is a hardware design feature that allows the microcontroller to retain critical data (using the DSGPRx registers) and maintains the RTCC when VDD is removed from the application. This is accomplished by supplying a backup power source to a specific power pin. VBAT mode is described in more detail in **Section 10.5 "VBAT Mode"**.

10.1.3 LOW-VOLTAGE/RETENTION REGULATOR

PIC24FJ128GC010 family devices incorporate a second on-chip voltage regulator, designed to provide power to select microcontroller features at 1.2V nominal. This regulator allows features, such as data RAM and the WDT, to be maintained in power-saving modes where they would otherwise be inactive, or maintain them at a lower power than would otherwise be the case.

The low-voltage/retention regulator is only available when Sleep or Deep Sleep modes are invoked. It is controlled by the $\overline{\text{LPCFG}}$ Configuration bit (CW1<10>) and in firmware by the RETEN bit (RCON<12>). $\overline{\text{LPCFG}}$ must be programmed (= `0`) and the RETEN bit must be set (= `1`) for the regulator to be enabled.

10.2 Idle Mode

Idle mode provides 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 10.8 "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 Interrupt Service Routine (ISR).

10.3 Sleep Mode

Sleep mode includes these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum provided that no I/O pin is sourcing current.
- The I/O pin directions and states are frozen.
- The Fail-Safe Clock Monitor does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode, if the WDT or RTCC with LPRC as the clock source, is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items, such as the Input Change Notification (ICN) on the I/O ports or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode.

The device will wake-up from Sleep mode on any of these events:

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

On wake-up from Sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

10.3.1 LOW-VOLTAGE/RETENTION SLEEP MODE

Low-Voltage/Retention Sleep mode functions as Sleep mode, with the same features and wake-up triggers. The difference is that the low-voltage/retention regulator allows Core Digital Logic Voltage (VCORE) to drop to 1.2V nominal. This permits an incremental reduction of power consumption over what would be required if VCORE was maintained at a 1.8V (minimum) level.

Low-Voltage Sleep mode requires a longer wake-up time than Sleep mode due to the additional time required to bring VCORE back to 1.8V (known as TREG). In addition, the use of the low-voltage/retention regulator limits the amount of current that can be sourced to any active peripherals, such as the RTCC, LCD, etc.

REGISTER 11-21: RPINR20: PERIPHERAL PIN SELECT INPUT REGISTER 20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
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	**SCK1R<5:0>:** Assign SPI1 Clock Input (SCK1IN) to Corresponding RPn or RPIn Pin bits
bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**SDI1R<5:0>:** Assign SPI1 Data Input (SDI1) to Corresponding RPn or RPIn Pin bits

REGISTER 11-22: RPINR21: PERIPHERAL PIN SELECT INPUT REGISTER 21

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
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	**U3CTSR<5:0>:** Assign UART3 Clear-to-Send ($\overline{\text{U3CTS}}$) to Corresponding RPn or RPIn Pin bits
bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**SS1R<5:0>:** Assign SPI1 Slave Select Input (SS1IN) to Corresponding RPn or RPIn Pin bits

REGISTER 11-38: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
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
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
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	**RP21R<5:0>:** RP21 Output Pin Mapping bits Peripheral Output Number n is assigned to pin, RP21 (see Table 11-4 for peripheral function numbers).
bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**RP20R<5:0>:** RP20 Output Pin Mapping bits Peripheral Output Number n is assigned to pin, RP20 (see Table 11-4 for peripheral function numbers).

REGISTER 11-39: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTER 11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
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
—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
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	**RP23R<5:0>:** RP23 Output Pin Mapping bits Peripheral Output Number n is assigned to pin, RP23 (see Table 11-4 for peripheral function numbers).
bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**RP22R<5:0>:** RP22 Output Pin Mapping bits Peripheral Output Number n is assigned to pin, RP22 (see Table 11-4 for peripheral function numbers).

19.3 USB Interrupts

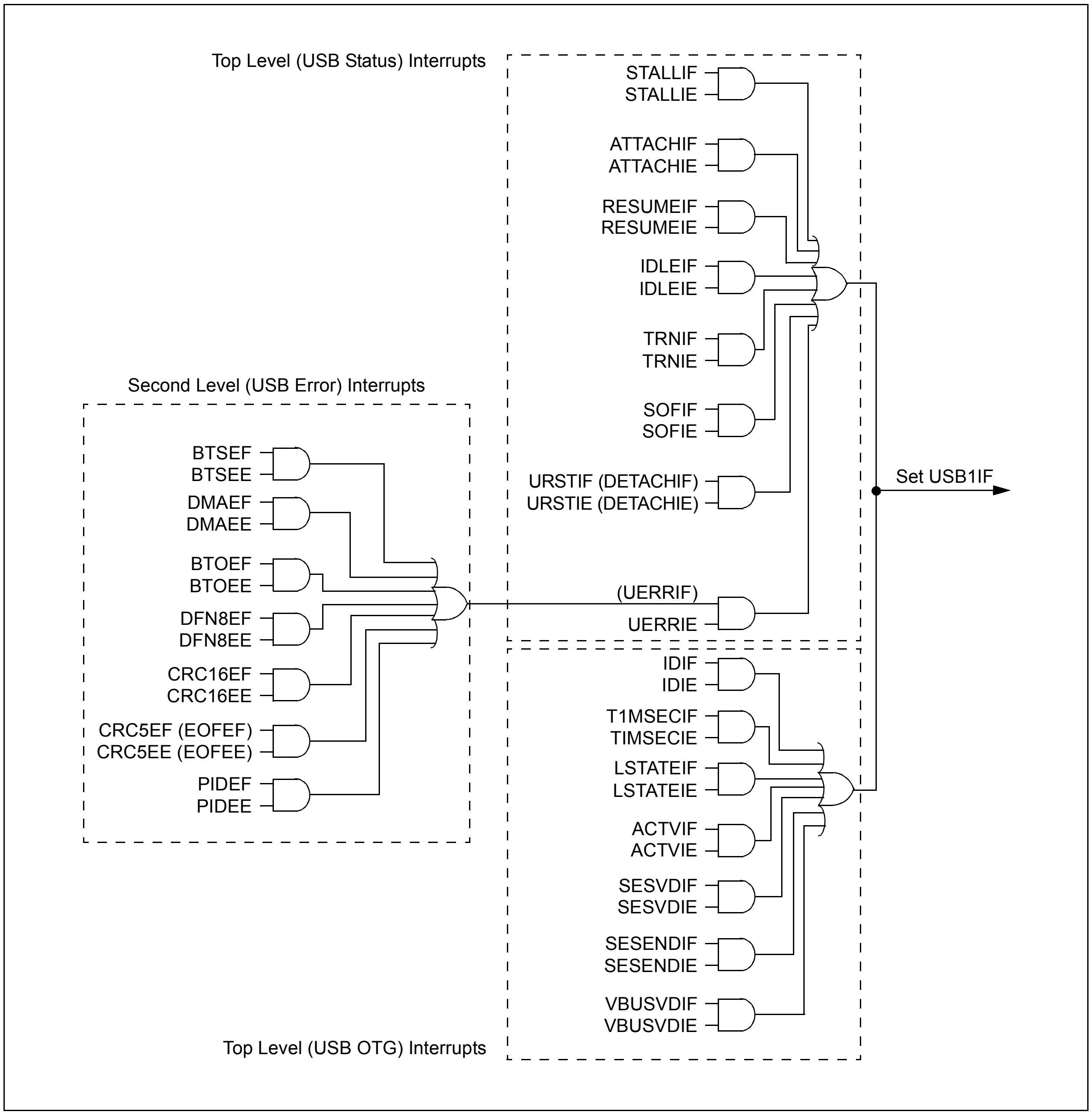
The USB OTG module has many conditions that can be configured to cause an interrupt. All interrupt sources use the same interrupt vector.

Figure 19-8 shows the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the U1IE and U1IR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the U1EIR and U1EIE registers.

An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level. Unlike the device-level interrupt flags in the IFSx registers, USB interrupt flags in the U1IR registers can only be cleared by writing a ‘1’ to the bit position.

Interrupts may be used to trap routine events in a USB transaction. Figure 19-9 provides some common events within a USB frame and their corresponding interrupts.

FIGURE 19-8: USB OTG INTERRUPT FUNNEL



19.7.2 USB INTERRUPT REGISTERS

REGISTER 19-14: U1OTGIR: USB OTG INTERRUPT STATUS REGISTER (HOST MODE ONLY)

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

R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	U-0	R/K-0, HS
IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDIF	SESENDIF	—	VBUSVDIF
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	K = Write ‘1’ to Clear 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 **IDIF:** ID State Change Indicator bit
1 = Change in ID state is detected
0 = No ID state change is detected

bit 6 **T1MSECIF:** 1 Millisecond Timer bit
1 = The 1 millisecond timer has expired
0 = The 1 millisecond timer has not expired

bit 5 **LSTATEIF:** Line State Stable Indicator bit
1 = USB line state (as defined by the SE0 and JSTATE bits) has been stable for 1 ms, but different from the last time
0 = USB line state has not been stable for 1 ms

bit 4 **ACTVIF:** Bus Activity Indicator bit
1 = Activity on the D+/D- lines or VBUS is detected
0 = No activity on the D+/D- lines or VBUS is detected

bit 3 **SESVDIF:** Session Valid Change Indicator bit
1 = VBUS has crossed VA_SESS_END (as defined in the *“USB 2.0 OTG Specification”*)[(1)]
0 = VBUS has not crossed VA_SESS_END

bit 2 **SESENDIF:** B-Device VBUS Change Indicator bit
1 = VBUS change on B-device is detected; VBUS has crossed VB_SESS_END (as defined in the *“USB 2.0 OTG Specification”*)[(1)]
0 = VBUS has not crossed VA_SESS_END

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **VBUSVDIF:** A-Device VBUS Change Indicator bit
1 = VBUS change on A-device is detected; VBUS has crossed VA_VBUS_VLD (as defined in the *“USB 2.0 OTG Specification”*)[(1)]
0 = No VBUS change on A-device is detected

Note 1: VBUS threshold crossings may either be rising or falling.

Note: Individual bits can only be cleared by writing a ‘1’ to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits, at the moment of the write, to become cleared.

REGISTER 19-19: U1EIR: USB ERROR INTERRUPT STATUS REGISTER

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

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS
BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
						EOFEF	
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	K = Write ‘1’ to Clear 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 **BTSEF:** Bit Stuff Error Flag bit
1 = Bit stuff error has been detected
0 = No bit stuff error has been detected

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **DMAEF:** DMA Error Flag bit
1 = A USB DMA error condition is detected; the data size indicated by the BD byte count field is less than the number of received bytes, the received data is truncated
0 = No DMA error

bit 4 **BTOEF:** Bus Turnaround Time-out Error Flag bit
1 = Bus turnaround time-out has occurred
0 = No bus turnaround time-out has occurred

bit 3 **DFN8EF:** Data Field Size Error Flag bit
1 = Data field was not an integral number of bytes
0 = Data field was an integral number of bytes

bit 2 **CRC16EF:** CRC16 Failure Flag bit
1 = CRC16 failed
0 = CRC16 passed

bit 1 For Device mode:
CRC5EF: CRC5 Host Error Flag bit
1 = Token packet is rejected due to CRC5 error
0 = Token packet is accepted (no CRC5 error)
For Host mode:
EOFEF: End-of-Frame (EOF) Error Flag bit
1 = End-of-Frame error has occurred
0 = End-of-Frame interrupt is disabled

bit 0 **PIDEF:** PID Check Failure Flag bit
1 = PID check failed
0 = PID check passed

Note: Individual bits can only be cleared by writing a ‘1’ to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits, at the moment of the write, to become cleared.

REGISTER 21-4: PMCON4: EPMP CONTROL REGISTER 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN<13:8>					
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
PTEN<7:3>					PTEN<2:0>		
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 **PTEN15:** PMA15 Port Enable bit
1 = PMA15 functions as either Address Line 15 or Chip Select 2
0 = PMA15 functions as port I/O

bit 14 **PTEN14:** PMA14 Port Enable bit
1 = PMA14 functions as either Address Line 14 or Chip Select 1
0 = PMA14 functions as port I/O

bit 13-3 **PTEN<13:3>:** EPMP Address Port Enable bits
1 = PMA<13:3> function as EPMP address lines
0 = PMA<13:3> function as port I/Os

bit 2-0 **PTEN<2:0>:** PMALU/PMALH/PMALL Strobe Enable bits
1 = PMA<2:0> function as either address lines or address latch strobes
0 = PMA<2:0> function as port I/Os

REGISTER 30-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0, HS	R-0, HSC
CON	COE	CPOL	—	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1(1)	EVPOL0(1)	—	CREF	—	—	CCH1	CCH0
bit 7							bit 0

Legend:	HS = Hardware Settable bit	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 **CON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled

bit 14 **COE:** Comparator Output Enable bit
1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only

bit 13 **CPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 12-10 **Unimplemented:** Read as '0'

bit 9 **CEVT:** Comparator Event bit
1 = Comparator event that is defined by EVPOL<1:0> has occurred; subsequent triggers and interrupts are disabled until the bit is cleared
0 = Comparator event has not occurred

bit 8 **COUT:** Comparator Output bit
When CPOL = 0:
1 = VIN+ > VIN-
0 = VIN+ < VIN-
When CPOL = 1:
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 7-6 **EVPOL<1:0>:** Trigger/Event/Interrupt Polarity Select bits(1)
11 = Trigger/event/interrupt is generated on any change of the comparator output (while CEVT = 0)
10 = Trigger/event/interrupt is generated on the high-to-low transition of the comparator output
01 = Trigger/event/interrupt is generated on the low-to-high transition of the comparator output
00 = Trigger/event/interrupt generation is disabled

bit 5 **Unimplemented:** Read as '0'

bit 4 **CREF:** Comparator Reference Select bits (non-inverting input)
1 = Non-inverting input connects to the internal CVREF voltage
0 = Non-inverting input connects to the CxINA pin

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

Note 1: If the EVPOL<1:0> bits are set to a value other than '00', the first interrupt generated will occur on any transition of COUT. Subsequent interrupts will occur based on the EVPOLx bits setting.

TABLE 36-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
RCALL	RCALL	Expr	Relative Call	1	2	None
	RCALL	Wn	Computed Call	1	2	None
REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
	REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
RESET	RESET		Software Device Reset	1	1	None
RETFIE	RETFIE		Return from Interrupt	1	3 (2)	None
RETLW	RETLW	#lit10,Wn	Return with Literal in Wn	1	3 (2)	None
RETURN	RETURN		Return from Subroutine	1	3 (2)	None
RLC	RLC	f	f = Rotate Left through Carry f	1	1	C, N, Z
	RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
	RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C, N, Z
RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N, Z
	RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N, Z
	RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC	RRC	f	f = Rotate Right through Carry f	1	1	C, N, Z
	RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
	RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N, Z
	RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
	RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N, Z
SE	SE	Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C, N, Z
SETM	SETM	f	f = FFFFh	1	1	None
	SETM	WREG	WREG = FFFFh	1	1	None
	SETM	Ws	Ws = FFFFh	1	1	None
SL	SL	f	f = Left Shift f	1	1	C, N, OV, Z
	SL	f,WREG	WREG = Left Shift f	1	1	C, N, OV, Z
	SL	Ws,Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
	SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N, Z
	SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N, Z
SUB	SUB	f	f = f – WREG	1	1	C, DC, N, OV, Z
	SUB	f,WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
	SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
	SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
	SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB	f	f = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	#lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SUBR	SUBR	f	f = WREG – f	1	1	C, DC, N, OV, Z
	SUBR	f,WREG	WREG = WREG – f	1	1	C, DC, N, OV, Z
	SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
	SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR	f	f = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SWAP	SWAP.b	Wn	Wn = Nibble Swap Wn	1	1	None
	SWAP	Wn	Wn = Byte Swap Wn	1	1	None

FIGURE 38-17: DEEP SLEEP IPD (+85°C AND -60°C) vs. VDD

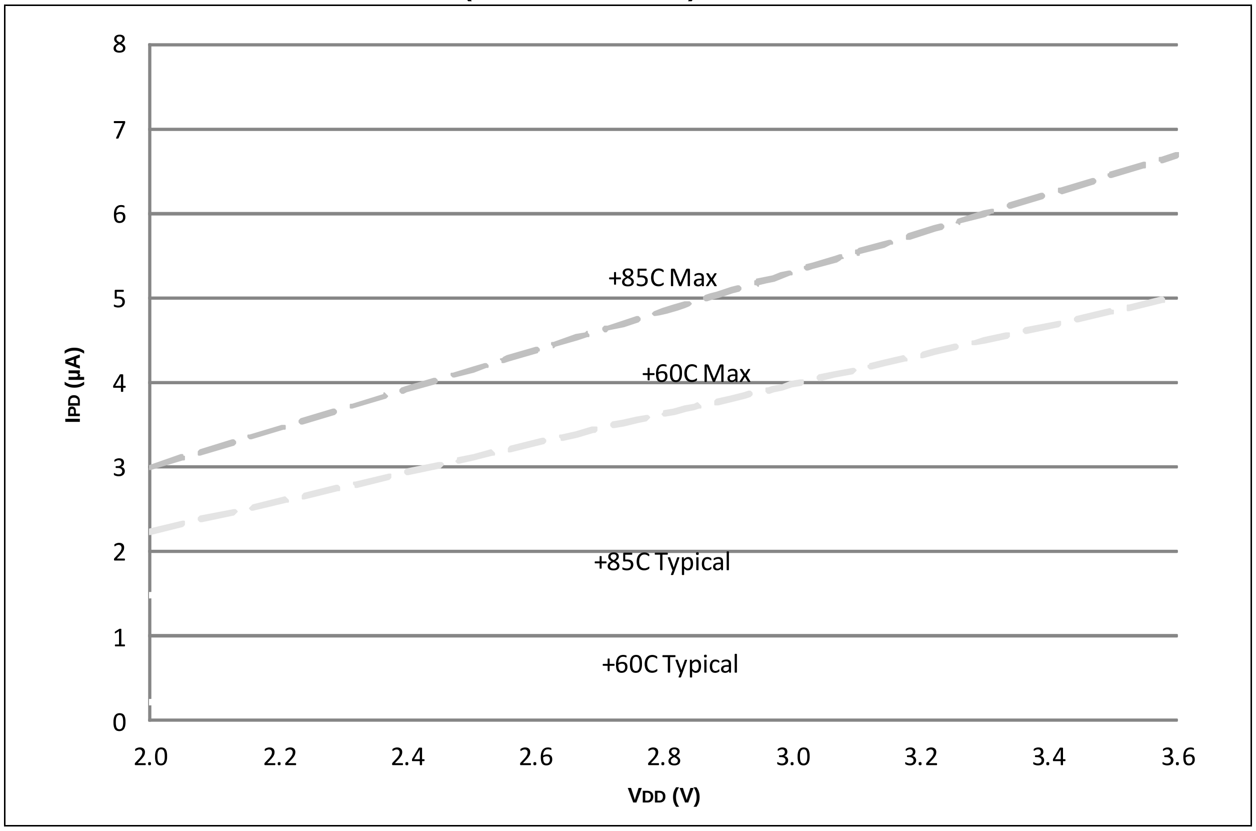
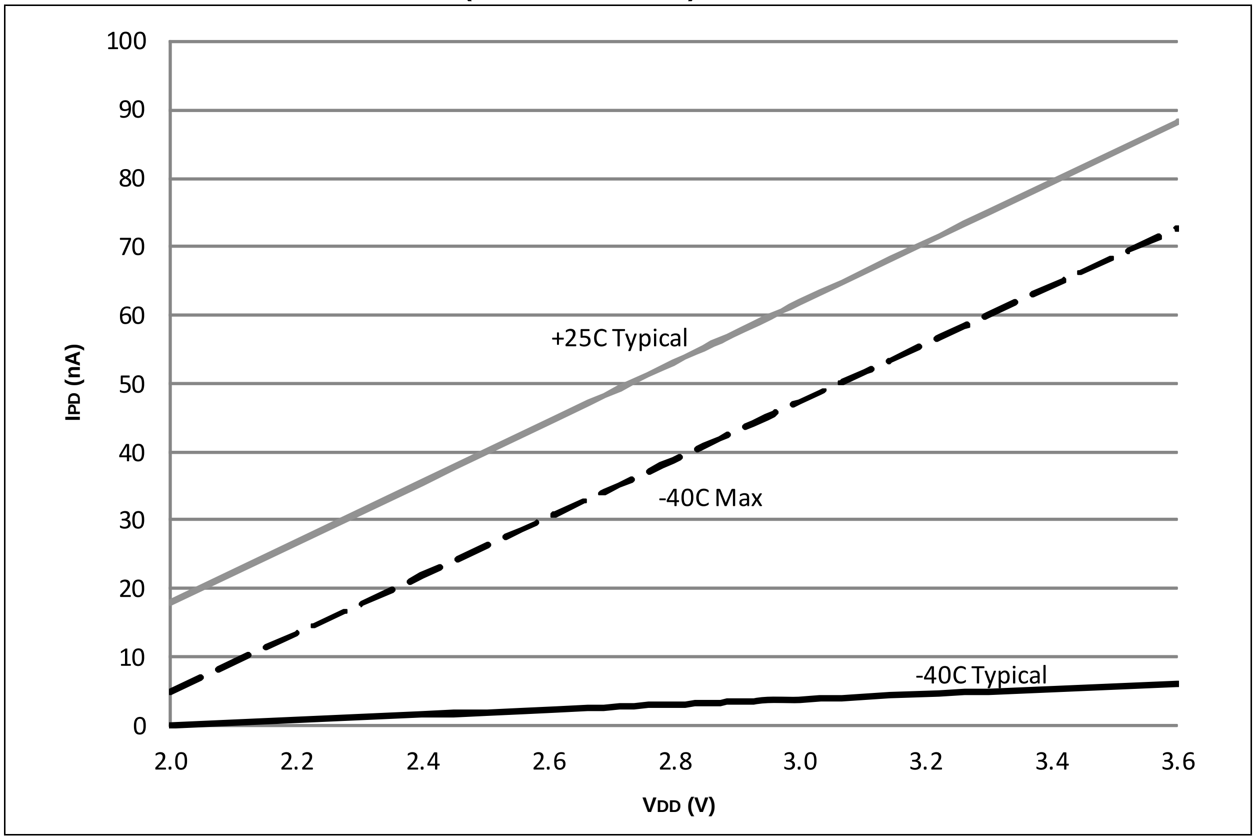


FIGURE 38-18: DEEP SLEEP IPD (+25°C AND -40°C) vs. VDD


IPD (nA)
+25C Typical
-40C Max
-40C Typical
VDD (V)