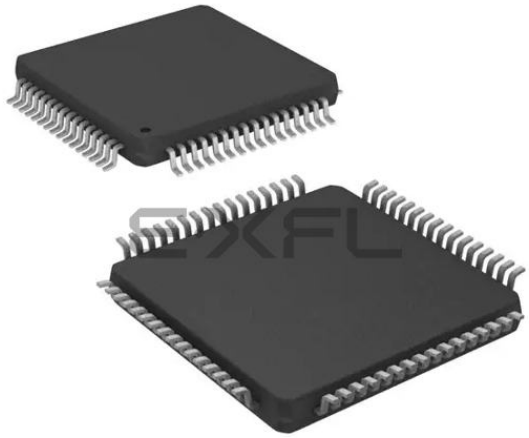


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

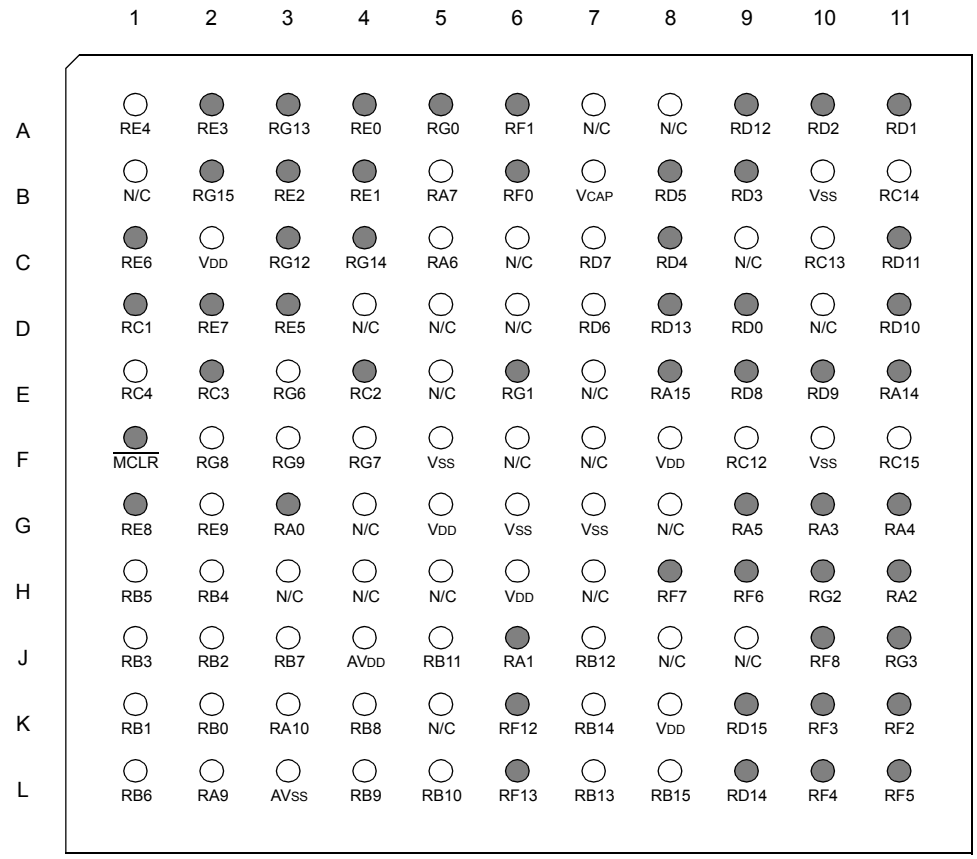
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, DMA, HLVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	512KB (170K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K 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	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj512gb606t-i-pt

Pin Diagrams(1) (Continued)

PIC24FJXXXGA610 121-Pin BGA



Legend: See Table 6 for a complete description of pin functions. Pinouts are subject to change.

Note 1: Gray shading indicates 5.5V tolerant input pins.

TABLE 1-3: PIC24FJ1024GA610/GB610 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Pin Function	Pin Number/Grid Locator						I/O	Input Buffer	Description
	GA606 64-Pin QFN/TQFP/ QFP	GB606 64-Pin QFN/ TQFP/QFP	GA610 100-Pin TQFP/ QFP	GB610 100-Pin TQFP/ QFP	GA612 121-Pin BGA	GB612 121-Pin BGA			
U6RX	27	27	41	41	J7	J7	I	ST	UART6 Receive Input
U6TX	18	18	27	27	J3	J3	O	DIG	UART6 Transmit Output
USBID	—	33	—	51	—	K10	I	ST	USB OTG ID Input
$\overline{\text{USBOEN}}$	—	12	—	21	—	H2	O	DIG	USB Output Enable (active-low)
VBUS	—	34	—	54	—	H8	I	—	VBUS Supply Detect
VCAP	56	56	85	85	B7	B7	P	—	External Filter Capacitor Connection (regulator enabled)
VDD	10,26,38	10,26,38	2,16,37, 46,62	2,16,37, 46,62	C2,F8, G5,H6, K8	C2,F8, G5,H6, K8	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins
VREF+	16	16	25,29	25,29	K2,K3	K2,K3	I	ANA	Comparator and A/D Reference Voltage (high) Input
VREF-	15	15	24,28	24,28	K1,L2	K1,L2	I	ANA	Comparator and A/D Reference Voltage (low) Input
VSS	9,25,41	9,25,41	15,36,45, 65,75	15,36,45, 65,75	B10,F5, F10,G6, G7	B10,F5, F10,G6, G7	P	—	Ground Reference for Peripheral Digital Logic and I/O Pins
VUSB3V3	—	35	—	55	—	H9	P	—	3.3V VUSB

Legend: TTL = TTL input buffer
ANA = Analog level input/output
DIG = Digital input/output
ST = Schmitt Trigger input buffer
I^2C = I^2C/SMBus input buffer
XCVR = Dedicated Transceiver

TABLE 4-13: EDS MEMORY ADDRESS WITH DIFFERENT PAGES AND ADDRESSES

DSRPAG (Data Space Read Register)	DSWPAG (Data Space Write Register)	Source/Destination Address while Indirect Addressing	24-Bit EA Pointing to EDS	Comment
x(1)	x(1)	0000h to 1FFFh	000000h to 001FFFh	Near Data Space(2)
		2000h to 7FFFh	002000h to 007FFFh	
001h	001h	8000h to FFFFh	008000h to 00FFFEh	EPMP Memory Space
002h	002h		010000h to 017FFEh	
003h • • • • • 1FFh	003h • • • • • 1FFh		018000h to 0187FEh • • • • FF8000h to FFFFFEh	
000h	000h		Invalid Address	Address Error Trap(3)

Note 1: If the source/destination address is below 8000h, the DSRPAG and DSWPAG registers are not considered.

2: This Data Space can also be accessed by Direct Addressing.

3: When the source/destination address is above 8000h and DSRPAG/DSWPAG are '0', an address error trap will occur.

4.2.6 SOFTWARE STACK

Apart from its use as a Working register, the W15 register in PIC24F devices is also used as a Software Stack Pointer (SSP). The pointer always points to the first available free word and grows from lower to higher addresses. It pre-decrements for stack pops and post-increments for stack pushes, as shown in Figure 4-7. Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

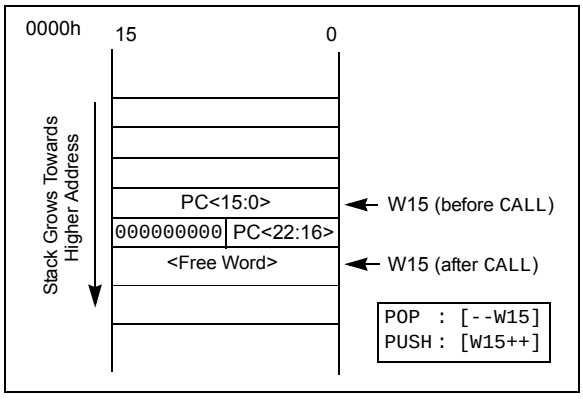
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

The Stack Pointer Limit Value register (SPLIM), associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' as all stack operations must be word-aligned. Whenever an EA is generated using W15 as a source or destination pointer, the resulting address is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal, and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 2000h in RAM, initialize the SPLIM with the value, 1FFEh.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0800h. This prevents the stack from interfering with the SFR space.

A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

FIGURE 4-7: CALL STACK FRAME



5.1 Summary of DMA Operations

The DMA Controller is capable of moving data between addresses according to a number of different parameters. Each of these parameters can be independently configured for any transaction; in addition, any or all of the DMA channels can independently perform a different transaction at the same time. Transactions are classified by these parameters:

- Source and destination (SFRs and data RAM)
- Data size (byte or word)
- Trigger source
- Transfer mode (One-Shot, Repeated or Continuous)
- Addressing modes (fixed address or address blocks, with or without address increment/decrement)

In addition, the DMA Controller provides channel priority arbitration for all channels.

5.1.1 SOURCE AND DESTINATION

Using the DMA Controller, data may be moved between any two addresses in the Data Space. The SFR space (0000h to 07FFh), or the data RAM space (0800h to FFFFh), can serve as either the source or the destination. Data can be moved between these areas in either direction or between addresses in either area. The four different combinations are shown in Figure 5-2.

If it is necessary to protect areas of data RAM, the DMA Controller allows the user to set upper and lower address boundaries for operations in the Data Space above the SFR space. The boundaries are set by the DMAH and DMAL Limit registers. If a DMA channel attempts an operation outside of the address boundaries, the transaction is terminated and an interrupt is generated.

5.1.2 DATA SIZE

The DMA Controller can handle both 8-bit and 16-bit transactions. Size is user-selectable using the SIZE bit (DMACHn<1>). By default, each channel is configured for word-sized transactions. When byte-sized transactions are chosen, the LSb of the source and/or destination address determines if the data represents the upper or lower byte of the data RAM location.

5.1.3 TRIGGER SOURCE

The DMA Controller can use any one of the device's interrupt sources to initiate a transaction. The DMA Trigger sources are listed in reverse order of their natural interrupt priority and are shown in Table 5-1.

Since the source and destination addresses for any transaction can be programmed independently of the Trigger source, the DMA Controller can use any Trigger to perform an operation on any peripheral. This also allows DMA channels to be cascaded to perform more complex transfer operations.

5.1.4 TRANSFER MODE

The DMA Controller supports four types of data transfers, based on the volume of data to be moved for each Trigger.

- One-Shot: A single transaction occurs for each Trigger.
- Continuous: A series of back-to-back transactions occur for each Trigger; the number of transactions is determined by the DMACNTn transaction counter.
- Repeated One-Shot: A single transaction is performed repeatedly, once per Trigger, until the DMA channel is disabled.
- Repeated Continuous: A series of transactions are performed repeatedly, one cycle per Trigger, until the DMA channel is disabled.

All transfer modes allow the option to have the source and destination addresses, and counter value automatically reloaded after the completion of a transaction. Repeated mode transfers do this automatically.

5.1.5 ADDRESSING MODES

The DMA Controller also supports transfers between single addresses or address ranges. The four basic options are:

- Fixed-to-Fixed: Between two constant addresses
- Fixed-to-Block: From a constant source address to a range of destination addresses
- Block-to-Fixed: From a range of source addresses to a single, constant destination address
- Block-to-Block: From a range to source addresses to a range of destination addresses

The option to select auto-increment or auto-decrement of source and/or destination addresses is available for Block Addressing modes.

In addition to the four basic modes, the DMA Controller also supports Peripheral Indirect Addressing (PIA) mode, where the source or destination address is generated jointly by the DMA Controller and a PIA capable peripheral. When enabled, the DMA channel provides a base source and/or destination address, while the peripheral provides a fixed range offset address.

For PIC24FJ1024GA610/GB610 family devices, the 12-bit A/D Converter module is the only PIA capable peripheral. Details for its use in PIA mode are provided in **Section 25.0 "12-Bit A/D Converter with Threshold Detect"**.

TABLE 8-2: INTERRUPT VECTOR DETAILS (CONTINUED)

Interrupt Source	IRQ #	IVT Address	Interrupt Bit Location		
			Flag	Enable	Priority
I2C1BC – I2C1 Bus Collision	84	0000BCh	IFS5<4>	IEC5<4>	I2C1BCInterrupt
I2C2BC – I2C2 Bus Collision	85	0000BEh	IFS5<5>	IEC5<5>	I2C2BCInterrupt
USB1 – USB1 Interrupt	86	0000C0h	IFS5<6>	IEC5<6>	USB1Interrupt
U4E – UART4 Error	87	0000C2h	IFS5<7>	IEC5<7>	U4ErrInterrupt
U4RX – UART4 Receiver	88	0000C4h	IFS5<8>	IEC5<8>	U4RXInterrupt
U4TX – UART4 Transmitter	89	0000C6h	IFS5<9>	IEC5<9>	U4TXInterrupt
SPI3 – SPI3 General	90	0000C8h	IFS5<10>	IEC5<10>	SPI3Interrupt
SPI3TX – SPI3 Transfer Done	91	0000CAh	IFS5<11>	IEC5<11>	SPI3TXInterrupt
—	92	92	—	—	—
—	93	93	—	—	—
CCP3 – Capture/Compare 3	94	0000D0h	IFS5<14>	IEC5<14>	CCP3Interrupt
CCP4 – Capture/Compare 4	95	0000D2h	IFS5<15>	IEC5<15>	CCP4Interrupt
CLC1 – Configurable Logic Cell 1	96	0000D4h	IFS6<0>	IEC6<0>	CLC1Interrupt
CLC2 – Configurable Logic Cell 2	97	0000D6h	IFS6<1>	IEC6<1>	CLC2Interrupt
CLC3 – Configurable Logic Cell 3	98	0000D8h	IFS6<2>	IEC6<2>	CLC3Interrupt
CLC4 – Configurable Logic Cell 4	99	0000DAh	IFS6<3>	IEC6<3>	CLC4Interrupt
—	100	—	—	—	—
CCT1 – Capture/Compare Timer1	101	0000DEh	IFS6<5>	IEC6<5>	CCT1Interrupt
CCT2 – Capture/Compare Timer2	102	0000E0h	IFS6<6>	IEC6<6>	CCT2Interrupt
—	103	—	—	—	—
—	104	—	—	—	—
—	105	—	—	—	—
FST – FRC Self-Tuning Interrupt	106	0000E8h	IFS6<10>	IEC6<10>	FSTInterrupt
—	107	—	—	—	—
—	108	—	—	—	—
I2C3BC – I2C3 Bus Collision	109	0000EEh	IFS6<13>	IEC6<13>	I2C3BCInterrupt
RTCCTS – Real-Time Clock Timestamp	110	0000F0h	IFS6<14>	IEC6<14>	RTCCTSInterrupt
U5RX – UART5 Receiver	111	0000F2h	IFS6<15>	IEC6<15>	U5RXInterrupt
U5TX – UART5 Transmitter	112	0000F4h	IFS7<0>	IEC7<0>	U5TXInterrupt
U5E – UART5 Error	113	0000F6h	IFS7<1>	IEC7<1>	U5ErrInterrupt
U6RX – UART6 Receiver	114	0000F8h	IFS7<2>	IEC7<2>	U6RXInterrupt
U6TX – UART6 Transmitter	115	0000FAh	IFS7<3>	IEC7<3>	U6TXInterrupt
U6E – UART6 Error	116	0000FCh	IFS7<4>	IEC7<4>	U6ErrInterrupt
JTAG – JTAG	117	0000FEh	IFS7<5>	IEC7<5>	JTAGInterrupt

9.8 Secondary Oscillator

9.8.1 BASIC SOSC OPERATION

PIC24FJ1024GA610/GB610 family devices do not have to set the SOSCEN bit to use the Secondary Oscillator. Any module requiring the SOSC (such as RTCC or Timer1) 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 (FOSC<3>) 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 when in High-Power mode:

- 12.5 pF loading capacitance
- 1.0 pF shunt capacitance
- A typical ESR of 35K; 50K maximum

In addition, the two external crystal loading capacitors should be in the range of 18-22 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 ±0.65 Hz.

9.8.3 LOW-POWER SOSC OPERATION

The Secondary Oscillator can operate in two distinct levels of power consumption based on device configuration. In Low-Power mode, the oscillator operates in a low drive strength, low-power state. By default, the oscillator uses a higher drive strength, and therefore, requires more power. Low-Power mode is selected by Configuration bit, SOSCHP (FDEVOPT1<3>). The lower drive strength of this mode makes the SOSC more sensitive to noise and requires a longer start-up time. This mode can be used with lower load capacitance crystals (6 pF-9 pF) having higher ESR ratings (50K-80K) to reduce Sleep current in the RTCC. When Low-Power mode is used, care must be taken in the design and layout of the SOSC circuit to ensure that the oscillator starts up and oscillates properly. PC board layout issues, stray capacitance and other factors will need to be carefully controlled in order for the crystal to operate.

For 32-bit cascaded operation, these steps are also necessary:

1. Set the OC32 bits for both registers (OCyCON2<8> and OCxCON2<8>). Enable the even numbered module first to ensure the modules will start functioning in unison.
2. Clear the OCTRIG bit of the even module (OCyCON2<7>), so the module will run in Synchronous mode.
3. Configure the desired output and Fault settings for OCy.
4. Force the output pin for OCx to the output state by clearing the OCTRIS bit.
5. If Trigger mode operation is required, configure the Trigger options in OCx by using the OCTRIG (OCxCON2<7>), TRIGMODE (OCxCON1<3>) and SYNCSEL<4:0> (OCxCON2<4:0>) bits.
6. Configure the desired Compare or PWM mode of operation (OCM<2:0>) for OCy first, then for OCx.

Depending on the output mode selected, the module holds the OCx pin in its default state and forces a transition to the opposite state when OCxR matches the timer. In Double Compare modes, OCx is forced back to its default state when a match with OCxRS occurs. The OCxIF interrupt flag is set after an OCxR match in Single Compare modes and after each OCxRS match in Double Compare modes.

Single-Shot pulse events only occur once, but may be repeated by simply rewriting the value of the OCxCON1 register. Continuous pulse events continue indefinitely until terminated.

15.3 Pulse-Width Modulation (PWM) Mode

In PWM mode, the output compare module can be configured for edge-aligned or center-aligned pulse waveform generation. All PWM operations are double-buffered (buffer registers are internal to the module and are not mapped into SFR space).

To configure the output compare module for PWM operation:

1. Configure the OCx output for one of the available Peripheral Pin Select pins if available on the OC module you are using. Otherwise, configure the dedicated OCx output pins.
2. Calculate the desired duty cycles and load them into the OCxR register.
3. Calculate the desired period and load it into the OCxRS register.
4. Select the current OCx as the synchronization source by writing 0x1F to the SYNCSEL<4:0> bits (OCxCON2<4:0>) and '0' to the OCTRIG bit (OCxCON2<7>).
5. Select a clock source by writing to the OCTSEL<2:0> bits (OCxCON1<12:10>).
6. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
7. Select the desired PWM mode in the OCM<2:0> bits (OCxCON1<2:0>).
8. Appropriate Fault inputs may be enabled by using the ENFLT<2:0> bits as described in Register 15-1.
9. If a timer is selected as a clock source, set the selected timer prescale value. The selected timer's prescaler output is used as the clock input for the OCx timer, and not the selected timer output.

Note:	This peripheral contains input and output functions that may need to be configured by the Peripheral Pin Select. See **Section 11.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2(2)	ENFLT1(2)
bit 15							bit 8

R/W-0	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0(2)	OCFLT2(2,3)	OCFLT1(2,4)	OCFLT0(2,4)	TRIGMODE	OCM2(1)	OCM1(1)	OCM0(1)
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **OCSIDL:** Output Compare x Stop in Idle Mode Control bit
- 1 = Output Compare x halts in CPU Idle mode
- 0 = Output Compare x continues to operate in CPU Idle mode

bit 12-10 **OCTSEL<2:0>:** Output Compare x Timer Select bits
- 111 = Peripheral clock (FCY)
- 110 = Reserved
- 101 = Reserved
- 100 = Timer1 clock (only synchronous clock is supported)
- 011 = Timer5 clock
- 010 = Timer4 clock
- 001 = Timer3 clock
- 000 = Timer2 clock

bit 9 **ENFLT2:** Fault Input 2 Enable bit(2)
- 1 = Fault 2 (Comparator 1/2/3 out) is enabled(3)
- 0 = Fault 2 is disabled

bit 8 **ENFLT1:** Fault Input 1 Enable bit(2)
- 1 = Fault 1 (OCFB pin) is enabled(4)
- 0 = Fault 1 is disabled

bit 7 **ENFLT0:** Fault Input 0 Enable bit(2)
- 1 = Fault 0 (OCFA pin) is enabled(4)
- 0 = Fault 0 is disabled

bit 6 **OCFLT2:** Output Compare x PWM Fault 2 (Comparator 1/2/3) Condition Status bit(2,3)
- 1 = PWM Fault 2 has occurred
- 0 = No PWM Fault 2 has occurred

bit 5 **OCFLT1:** Output Compare x PWM Fault 1 (OCFB pin) Condition Status bit(2,4)
- 1 = PWM Fault 1 has occurred
- 0 = No PWM Fault 1 has occurred

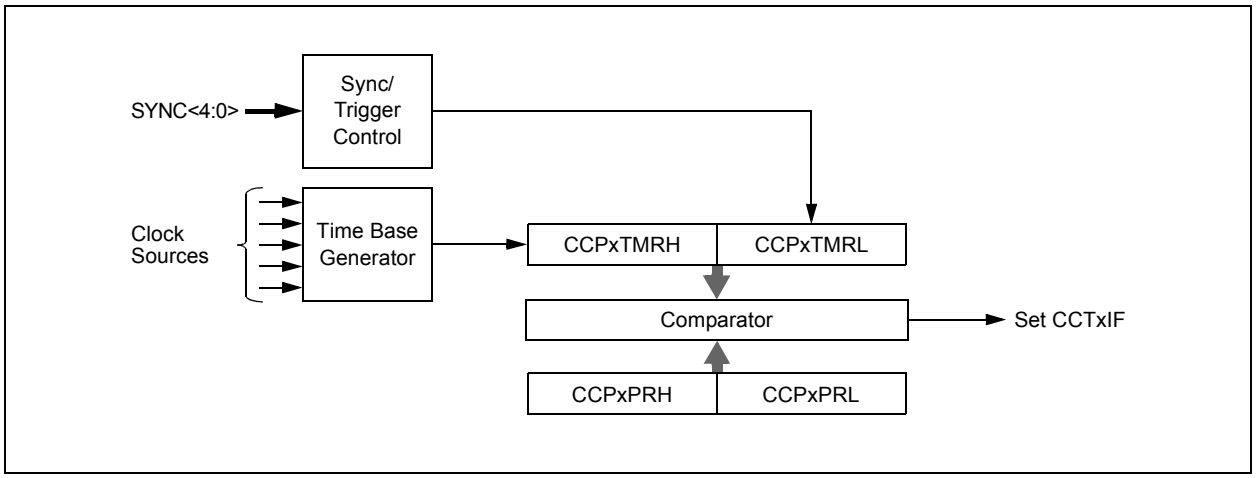
Note 1: The OCx output must also be configured to an available RPn pin. For more information, see **Section 11.4 "Peripheral Pin Select (PPS)"**.

2: The Fault input enable and Fault status bits are valid when OCM<2:0> = 111 or 110.

3: The Comparator 1 output controls the OC1-OC3 channels, Comparator 2 output controls the OC4-OC6 channels, Comparator 3 output controls the OC7-OC9 channels.

4: The OCFA/OCFB Fault inputs must also be configured to an available RPn/RPIn pin. For more information, see **Section 11.4 "Peripheral Pin Select (PPS)"**.

FIGURE 16-4: 32-BIT TIMER MODE



16.3 Output Compare Mode

Output Compare mode compares the Timer register value with the value of one or two Compare registers, depending on its mode of operation. The Output Compare x module, on compare match events, has the ability to generate a single output transition or a train of output pulses. Like most PIC® MCU peripherals, the Output Compare x module can also generate interrupts on a compare match event.

Table 16-2 shows the various modes available in Output Compare modes.

TABLE 16-2: OUTPUT COMPARE/PWM MODES

MOD<3:0> (CCPxCON1L<3:0>)	T32 (CCPxCON1L<5>)	Operating Mode	
0001	0	Output High on Compare (16-bit)	Single Edge Mode
0001	1	Output High on Compare (32-bit)	
0010	0	Output Low on Compare (16-bit)	
0010	1	Output Low on Compare (32-bit)	
0011	0	Output Toggle on Compare (16-bit)	
0011	1	Output Toggle on Compare (32-bit)	
0100	0	Dual Edge Compare (16-bit)	Dual Edge Mode
0101	0	Dual Edge Compare (16-bit buffered)	PWM Mode
0110	0	Center-Aligned Pulse (16-bit buffered)	Center PWM Mode
0111	0	Variable Frequency Pulse (16-bit)	
1111	0	External Input Source Mode (16-bit)	

NOTES:

REGISTER 17-10: SPIxIMSKH: SPIx INTERRUPT MASK REGISTER HIGH

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RXWIEN	—	RXMSK5(1)	RXMSK4(1,4)	RXMSK3(1,3)	RXMSK2(1,2)	RXMSK1(1)	RXMSK0(1)
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXWIEN	—	TXMSK5(1)	TXMSK4(1,4)	TXMSK3(1,3)	TXMSK2(1,2)	TXMSK1(1)	TXMSK0(1)
bit 7							bit 0

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

bit 15 **RXWIEN:** Receive Watermark Interrupt Enable bit
1 = Triggers receive buffer element watermark interrupt when RXMSK<5:0> ≤ RXELM<5:0>
0 = Disables receive buffer element watermark interrupt

bit 14 **Unimplemented:** Read as '0'

bit 13-8 **RXMSK<5:0>:** RX Buffer Mask bits(1,2,3,4)
RX mask bits; used in conjunction with the RXWIEN bit.

bit 7 **TXWIEN:** Transmit Watermark Interrupt Enable bit
1 = Triggers transmit buffer element watermark interrupt when TXMSK<5:0> = TXELM<5:0>
0 = Disables transmit buffer element watermark interrupt

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

bit 5-0 **TXMSK<5:0>:** TX Buffer Mask bits(1,2,3,4)
TX mask bits; used in conjunction with the TXWIEN bit.

Note 1: Mask values higher than FIFODEPTH are not valid. The module will not trigger a match for any value in this case.
2: RXMSK2 and TXMSK2 bits are only present when FIFODEPTH = 8 or higher.
3: RXMSK3 and TXMSK3 bits are only present when FIFODEPTH = 16 or higher.
4: RXMSK4 and TXMSK4 bits are only present when FIFODEPTH = 32.

19.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

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"*, **"UART"** (DS39708), which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM.

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the PIC24F device family. The UART is a full-duplex, asynchronous system that can communicate with peripheral devices, such as personal computers, LIN/J2602, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins. The UART module includes an IrDA® encoder/decoder unit.

The PIC24FJ1024GA610/GB610 family devices are equipped with six UART modules, referred to as UART1, UART2, UART3, UART4, UART5 and UART6.

The primary features of the UARTx modules are:

- Full-Duplex, 8 or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with the $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Pins
- Fully Integrated Baud Rate Generator with 16-Bit Prescaler
- Baud Rates Range from up to 1 Mbps and Down to 15 Hz at 16 MIPS in 16x mode
- Baud Rates Range from up to 4 Mbps and Down to 61 Hz at 16 MIPS in 4x mode
- 4-Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-bit mode with Address Detect (9^{th} bit = 1)
- Separate Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support
- Polarity Control for Transmit and Receive Lines
- Support for Sync and Break Characters
- Supports Automatic Baud Rate Detection
- IrDA® Encoder and Decoder Logic
- Includes DMA Support
- 16x Baud Clock Output for IrDA Support

A simplified block diagram of the UARTx module is shown in Figure 19-1. The UARTx module consists of these key important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

Note: Throughout this section, references to register and bit names that may be associated with a specific UART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "UxSTA" might refer to the Status register for either UART1, UART2, UART3, UART4, UART5 or UART6.

20.2 USB Buffer Descriptors and the BDT

Endpoint buffer control is handled through a structure called the Buffer Descriptor Table (BDT). This provides a flexible method for users to construct and control endpoint buffers of various lengths and configurations.

The BDT can be located in any available 512-byte, aligned block of data RAM. The BDT Pointer (U1BDTP1) contains the upper address byte of the BDT and sets the location of the BDT in RAM. The user must set this pointer to indicate the table's location.

The BDT is composed of Buffer Descriptors (BDs) which are used to define and control the actual buffers in the USB RAM space. Each BD consists of two 16-bit, "soft" (non-fixed-address) registers, BDnSTAT and BDnADR, where n represents one of the 64 possible BDs (range of 0 to 63). BDnSTAT is the status register for BDn, while BDnADR specifies the starting address for the buffer associated with BDn.

Note: Since BDnADR is a 16-bit register, only the first 64 Kbytes of RAM can be accessed by the USB module.

Depending on the endpoint buffering configuration used, there are up to 64 sets of Buffer Descriptors, for a total of 256 bytes. At a minimum, the BDT must be at least 8 bytes long. This is because the *"USB 2.0 Specification"* mandates that every device must have Endpoint 0 with both input and output for initial setup.

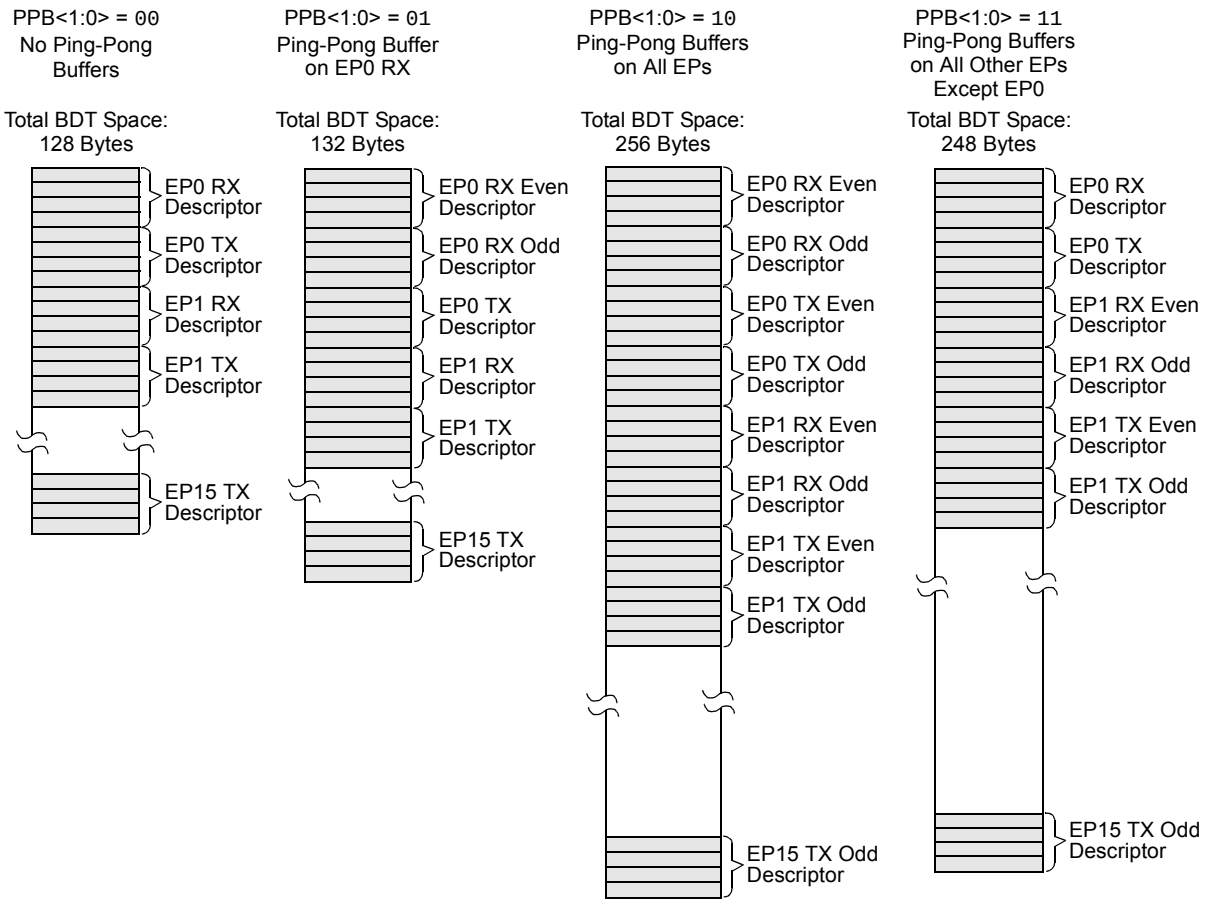
Endpoint mapping in the BDT is dependent on three variables:

- Endpoint number (0 to 15)
- Endpoint direction (RX or TX)
- Ping-pong settings (U1CNFG1<1:0>)

Figure 20-7 illustrates how these variables are used to map endpoints in the BDT.

In Host mode, only Endpoint 0 Buffer Descriptors are used. All transfers utilize the Endpoint 0 Buffer Descriptor and Endpoint Control register (U1EP0). For received packets, the attached device's source endpoint is indicated by the value of ENDPT<3:0> in the USB Status register (U1STAT<7:4>). For transmitted packets, the attached device's destination endpoint is indicated by the value written to the USB Token register (U1TOK).

FIGURE 20-7: BDT MAPPING FOR ENDPOINT BUFFERING MODES



Note: Memory area is not shown to scale.

20.3 USB Interrupts

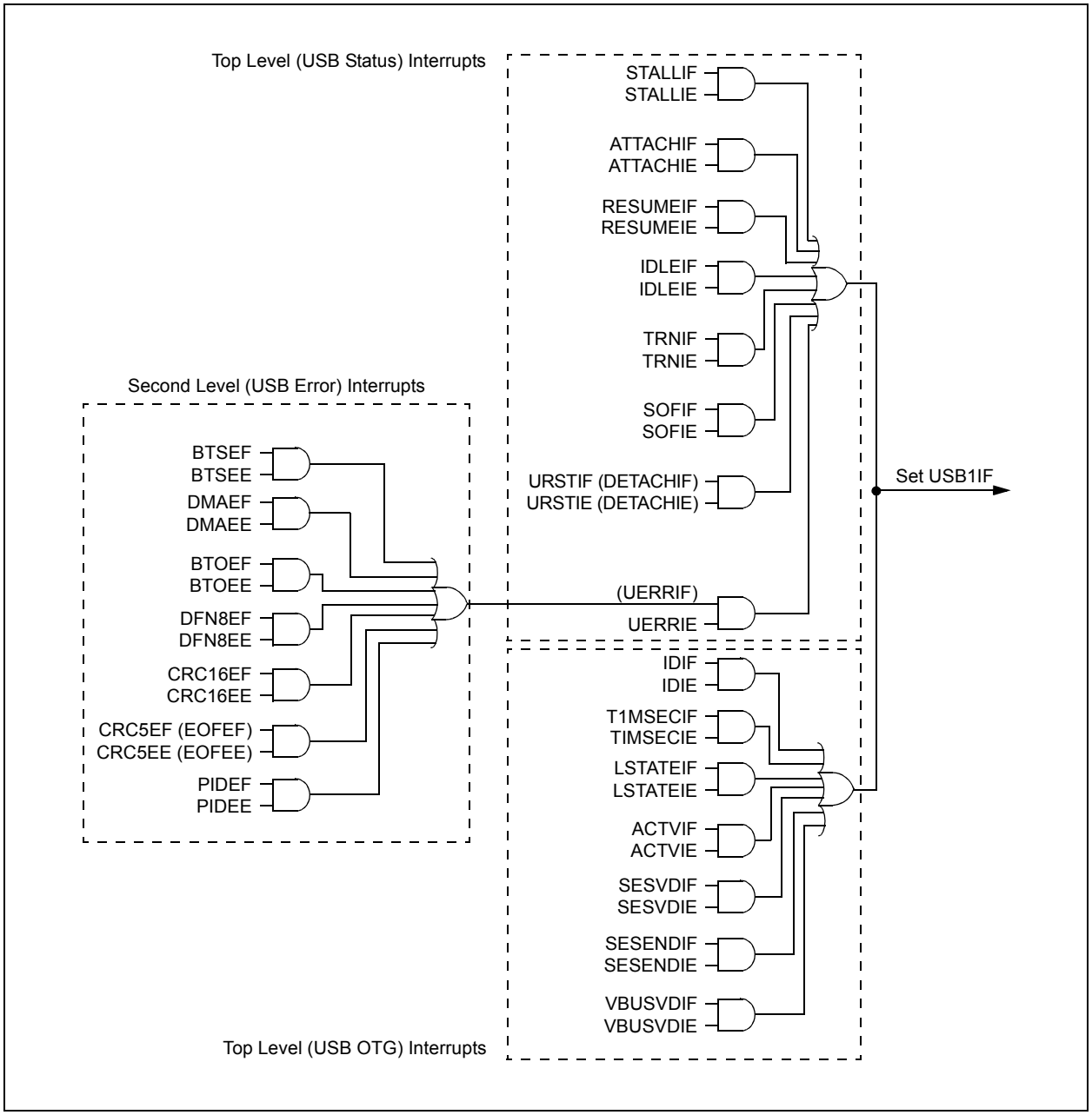
The USB OTG module has many conditions that can be configured to cause an interrupt. All interrupt sources use the same interrupt vector.

Figure 20-8 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 20-9 provides some common events within a USB frame and their corresponding interrupts.

FIGURE 20-8: USB OTG INTERRUPT FUNNEL



REGISTER 20-16: U1IR: USB INTERRUPT STATUS REGISTER (DEVICE 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	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
STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	K = Write '1' to Clear bit	HS = Hardware Settable bit	
-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 **STALLIF:** STALL Handshake Interrupt bit
- 1 = A STALL handshake was sent by the peripheral during the handshake phase of the transaction in Device mode
- 0 = A STALL handshake has not been sent

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

bit 5 **RESUMEIF:** Resume Interrupt bit
- 1 = A K-state is observed on the D+ or D- pin for 2.5 μs (differential '1' for low speed, differential '0' for full speed)
- 0 = No K-state is observed

bit 4 **IDLEIF:** Idle Detect Interrupt bit
- 1 = Idle condition is detected (constant Idle state of 3 ms or more)
- 0 = No Idle condition is detected

bit 3 **TRNIF:** Token Processing Complete Interrupt bit
- 1 = Processing of the current token is complete; read the U1STAT register for endpoint information
- 0 = Processing of the current token is not complete; clear the U1STAT register or load the next token from STAT (clearing this bit causes the STAT FIFO to advance)

bit 2 **SOFIF:** Start-of-Frame Token Interrupt bit
- 1 = A Start-of-Frame token is received by the peripheral or the Start-of-Frame threshold is reached by the host
- 0 = No Start-of-Frame token is received or threshold reached

bit 1 **UERRIF**: USB Error Condition Interrupt bit
- 1 = An unmasked error condition has occurred; only error states enabled in the U1EIE register can set this bit
- 0 = No unmasked error condition has occurred

bit 0 **URSTIF:** USB Reset Interrupt bit
- 1 = Valid USB Reset has occurred for at least 2.5 μs; Reset state must be cleared before this bit can be reasserted
- 0 = No USB Reset has occurred; 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

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 22-5: RTCCON3L: RTCC CONTROL REGISTER 3 (LOW)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWCSAMP7	PWCSAMP6	PWCSAMP5	PWCSAMP4	PWCSAMP3	PWCSAMP2	PWCSAMP1	PWCSAMP0
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
PWCSTAB7	PWCSTAB6	PWCSTAB5	PWCSTAB4	PWCSTAB3	PWCSTAB2	PWCSTAB1	PWCSTAB0
bit 7							bit 0

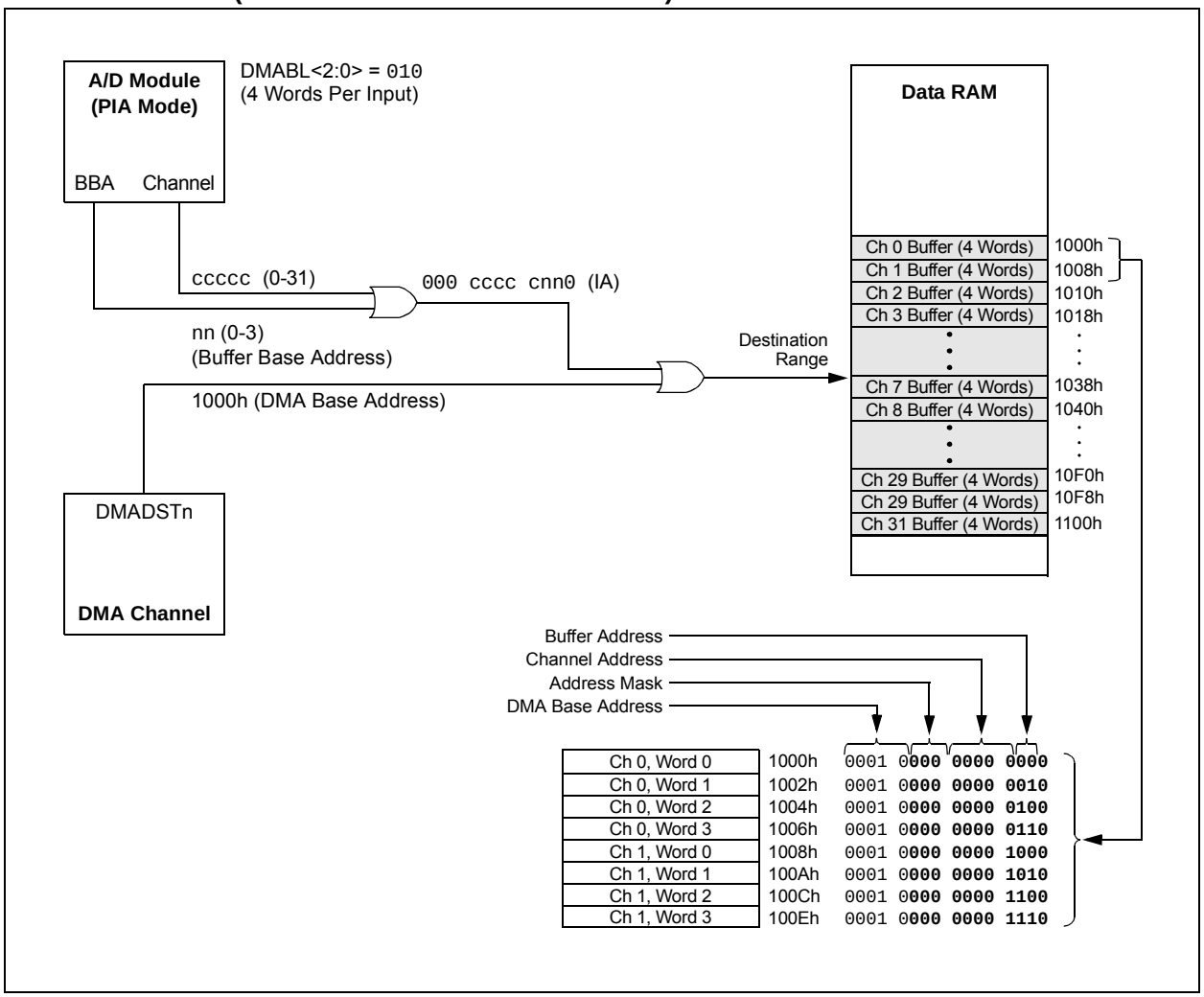
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **PWCSAMP<7:0>:** Power Control Sample Window Timer bits
11111111 = Sample window is always enabled, even when PWCEN = 0
11111110 = Sample window is 254 TPWCCLK clock periods
•
•
•
00000001 = Sample window is 1 TPWCCLK clock period
00000000 = No sample window

bit 7-0 **PWCSTAB<7:0>:** Power Control Stability Window Timer bits(1)
11111111 = Stability window is 255 TPWCCLK clock periods
11111110 = Stability window is 254 TPWCCLK clock periods
•
•
•
00000001 = Stability window is 1 TPWCCLK clock period
00000000 = No stability window; sample window starts when the alarm event triggers

Note 1: The sample window always starts when the stability window timer expires, except when its initial value is 00h.

FIGURE 25-2: EXAMPLE OF BUFFER ADDRESS GENERATION IN PIA MODE (4-WORD BUFFERS PER CHANNEL)



29.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

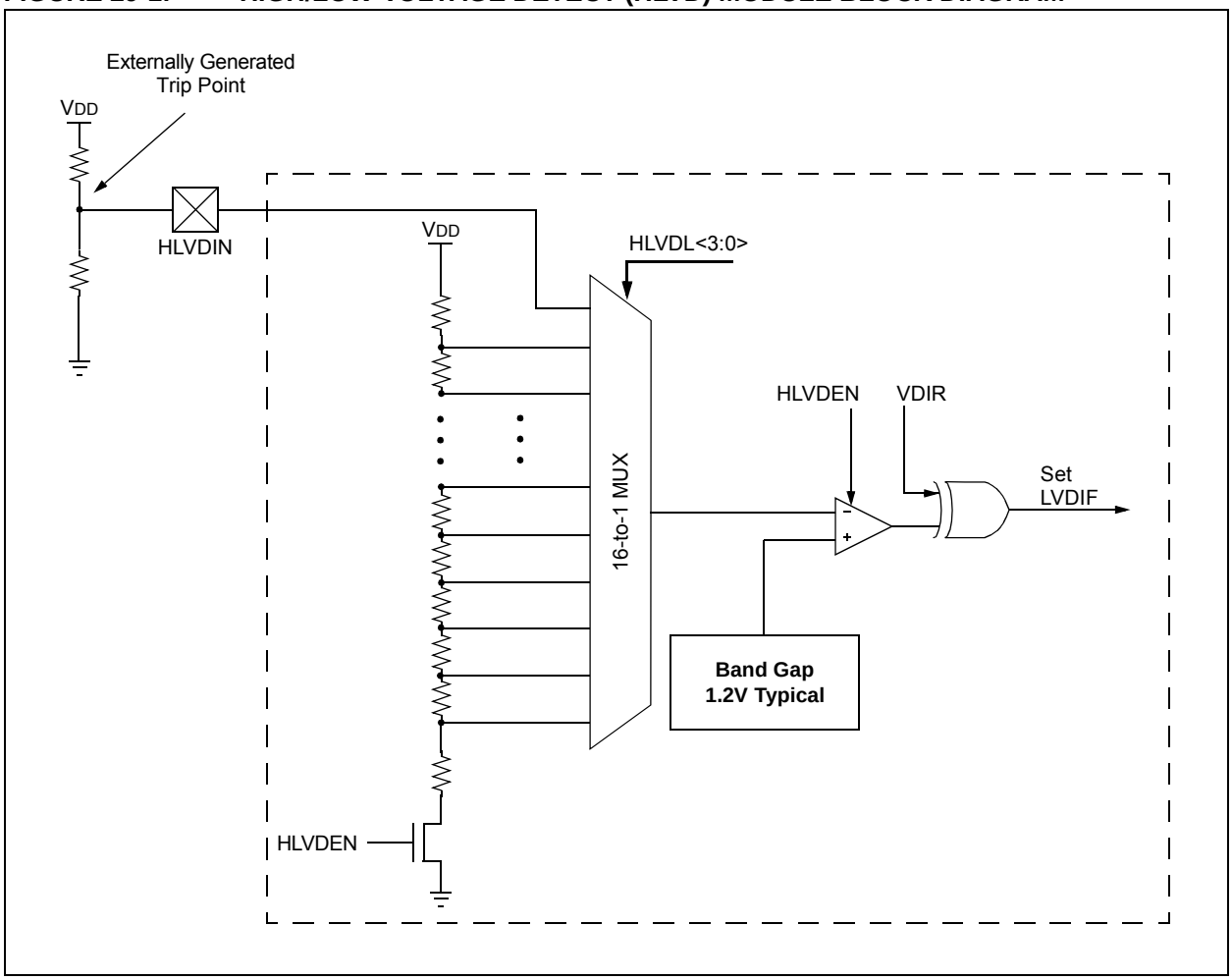
Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the High/Low-Voltage Detect, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)"** (DS39725), which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM.

The High/Low-Voltage Detect (HLVD) module is a programmable circuit that allows the user to specify both the device voltage trip point and the direction of change.

An interrupt flag is set if the device experiences an excursion past the trip point in the direction of change. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to the interrupt. The LVDIF flag may be set during a POR or BOR event. The firmware should clear the flag before the application uses it for the first time, even if the interrupt was disabled.

The HLVD Control register (see Register 29-1) completely controls the operation of the HLVD module. This allows the circuitry to be "turned off" by the user under software control, which minimizes the current consumption for the device. The HLVDEN bit (HLVDCON<15>) should be cleared when writing data to the HLVDCON register. Once the register is configured, the module is enabled from power-down by setting HLVDEN. The application must wait a minimum of 5 µS before clearing the HLVDIF flag and using the module after HLVDEN has been set.

FIGURE 29-1: HIGH/LOW-VOLTAGE DETECT (HLVD) MODULE BLOCK DIAGRAM



REGISTER 30-6: FOSCSEL CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
IESO	PLLMODE3	PLLMODE2	PLLMODE1	PLLMODE0	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

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

bit 23-10 **Unimplemented:** Read as ‘1’

bit 9-8 **Reserved:** Maintain as ‘0’

bit 7 **IESO:** Two-Speed Oscillator Start-up Enable bit
1 = Starts up the device with FRC, then automatically switches to the user-selected oscillator when ready
0 = Starts up the device with the user-selected oscillator source

bit 6-3 **PLLMODE<3:0>:** Frequency Multiplier Select bits
1111 = No PLL is used (PLLEN bit is unavailable)
1110 = 8x PLL is selected
1101 = 6x PLL is selected
1100 = 4x PLL is selected
0111 = 96 MHz USB PLL is selected (Input Frequency = 48 MHz)
0110 = 96 MHz USB PLL is selected (Input Frequency = 32 MHz)
0101 = 96 MHz USB PLL is selected (Input Frequency = 24 MHz)
0100 = 96 MHz USB PLL is selected (Input Frequency = 20 MHz)
0011 = 96 MHz USB PLL is selected (Input Frequency = 16 MHz)
0010 = 96 MHz USB PLL is selected (Input Frequency = 12 MHz)
0001 = 96 MHz USB PLL is selected (Input Frequency = 8 MHz)
0000 = 96 MHz USB PLL is selected (Input Frequency = 4 MHz)

bit 2-0 **FNOSC<2:0>:** Oscillator Selection bits
111 = Oscillator with Frequency Divider (OSCFDIV)
110 = Digitally Controlled Oscillator (DCO)
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (SOSC)
011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

REGISTER 30-10: FICD CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
BTSWP	—	—	—	—	—	—	—
bit 15							bit 8

r-1	U-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1
—	—	JTAGEN	—	—	—	ICS1	ICS0
bit 7							bit 0

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

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15 **BTSWP:** BOOTSWP Instruction Enable bit
1 = BOOTSWP instruction is disabled
0 = BOOTSWP instruction is enabled

bit 14-8 **Unimplemented:** Read as ‘1’

bit 7 **Reserved:** Maintain as ‘1’

bit 6 **Unimplemented:** Read as ‘1’

bit 5 **JTAGEN:** JTAG Port Enable bit
1 = JTAG port is enabled
0 = JTAG port is disabled

bit 4-2 **Unimplemented:** Read as ‘1’

bit 1-0 **ICS<1:0>:** ICD Communication Channel Select bits
11 = Communicates on PGEC1/PGED1
10 = Communicates on PGEC2/PGED2
01 = Communicates on PGEC3/PGED3
00 = Reserved; do not use