



Welcome to [E-XFL.COM](https://www.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	70 MIPS
Connectivity	I <sup>2</sup> C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	35
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 16
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 9x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UQFN (6x6)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic24ep128gp204t-i-mv">https://www.e-xfl.com/product-detail/microchip-technology/pic24ep128gp204t-i-mv</a>

FIGURE 2-7: INTERLEAVED PFC

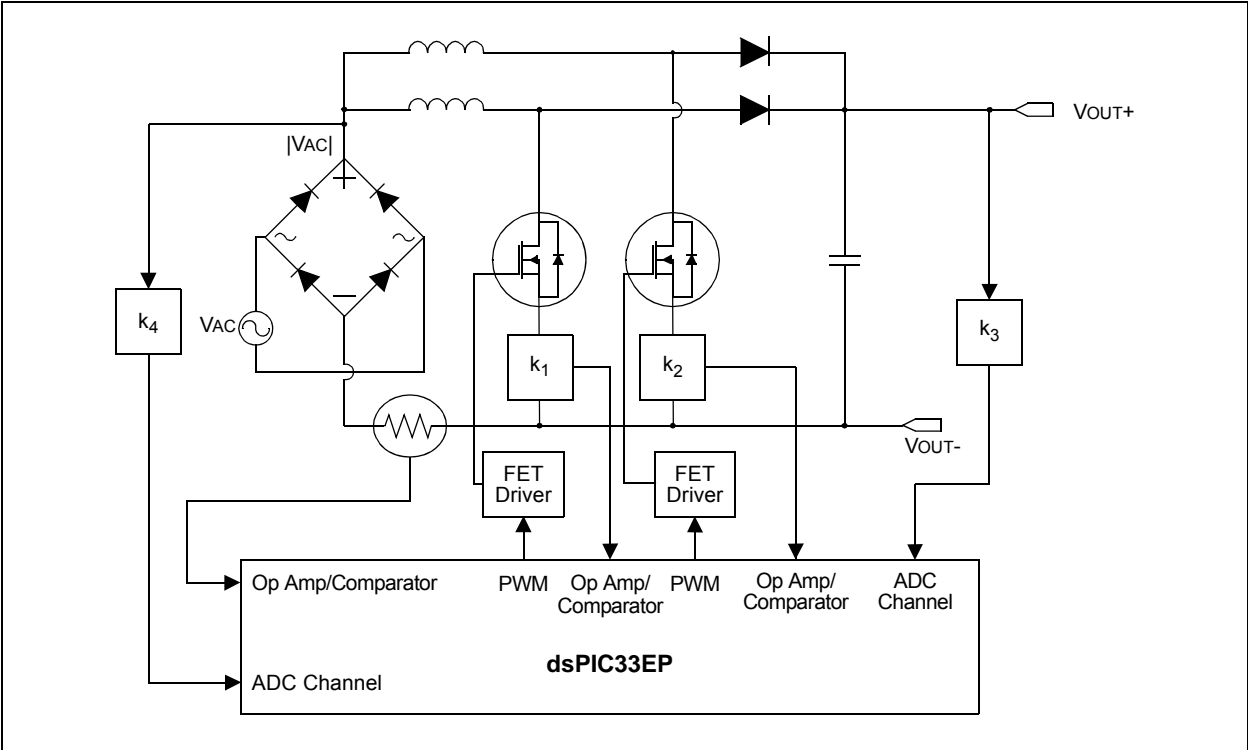
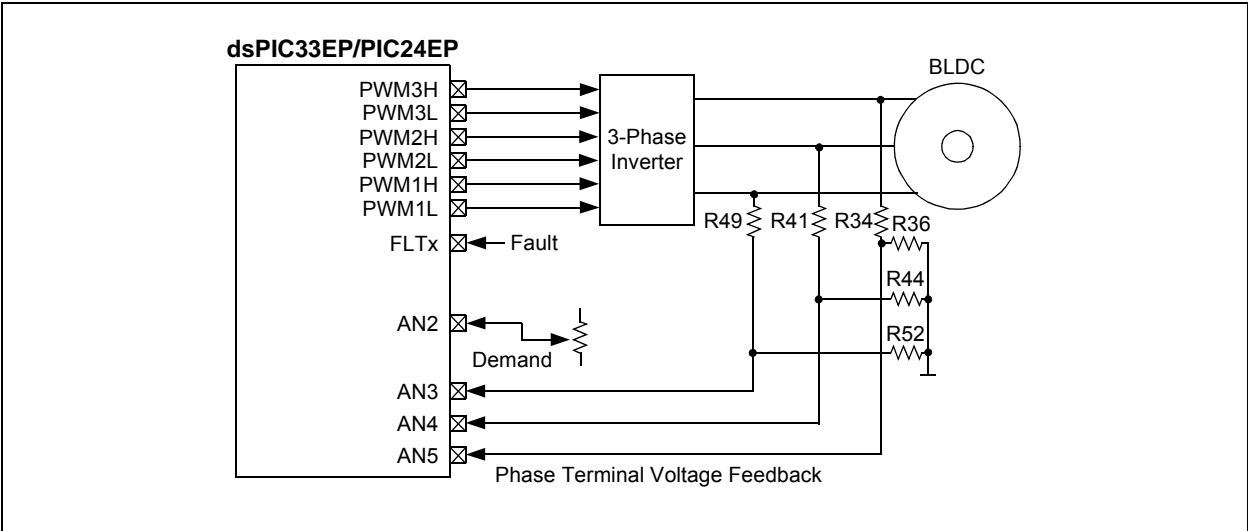


FIGURE 2-8: BEMF VOLTAGE MEASURED USING THE ADC MODULE



**FIGURE 7-1: dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X AND PIC24EPXXXGP/MC20X INTERRUPT VECTOR TABLE**

<div style="display: flex; align-items: center;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); margin-right: 5px;">Decreasing Natural Order Priority</div> <div style="flex-grow: 1; border-left: 1px solid black; border-right: 1px solid black; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; height: 10px; background: linear-gradient(to right, transparent 49%, black 49% 51%, black 51% 52%, transparent 52%);"></div> <div style="position: absolute; bottom: 0; left: 0; right: 0; height: 10px; background: linear-gradient(to right, transparent 49%, black 49% 51%, black 51% 52%, transparent 52%);"></div> <div style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); font-weight: bold;">IVT</div> </div> </div>	Reset – GOTO Instruction	0x000000	<div style="display: flex; align-items: center;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); margin-right: 5px;">See Table 7-1 for Interrupt Vector Details</div> <div style="flex-grow: 1; border-left: 1px solid black; border-right: 1px solid black; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; height: 10px; background: linear-gradient(to right, transparent 49%, black 49% 51%, black 51% 52%, transparent 52%);"></div> <div style="position: absolute; bottom: 0; left: 0; right: 0; height: 10px; background: linear-gradient(to right, transparent 49%, black 49% 51%, black 51% 52%, transparent 52%);"></div> </div> </div>
	Reset – GOTO Address	0x000002	
	Oscillator Fail Trap Vector	0x000004	
	Address Error Trap Vector	0x000006	
	Generic Hard Trap Vector	0x000008	
	Stack Error Trap Vector	0x00000A	
	Math Error Trap Vector	0x00000C	
	DMAC Error Trap Vector	0x00000E	
	Generic Soft Trap Vector	0x000010	
	Reserved	0x000012	
	Interrupt Vector 0	0x000014	
	Interrupt Vector 1	0x000016	
	:	:	
	:	:	
	:	:	
	Interrupt Vector 52	0x00007C	
	Interrupt Vector 53	0x00007E	
	Interrupt Vector 54	0x000080	
	:	:	
	:	:	
	:	:	
	Interrupt Vector 116	0x0000FC	
	Interrupt Vector 117	0x0000FE	
	Interrupt Vector 118	0x000100	
	Interrupt Vector 119	0x000102	
	Interrupt Vector 120	0x000104	
	:	:	
	:	:	
	:	:	
	Interrupt Vector 244	0x0001FC	
	Interrupt Vector 245	0x0001FE	
	START OF CODE	0x000200	

## 8.0 DIRECT MEMORY ACCESS (DMA)

**Note 1:** This data sheet summarizes the features of the dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X and PIC24EPXXXGP/MC20X families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to “**Direct Memory Access (DMA)**” (DS70348) in the “*dsPIC33/PIC24 Family Reference Manual*”, which is available from the Microchip web site ([www.microchip.com](http://www.microchip.com)).

**2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 “Memory Organization”** in this data sheet for device-specific register and bit information.

The DMA Controller transfers data between Peripheral Data registers and Data Space SRAM

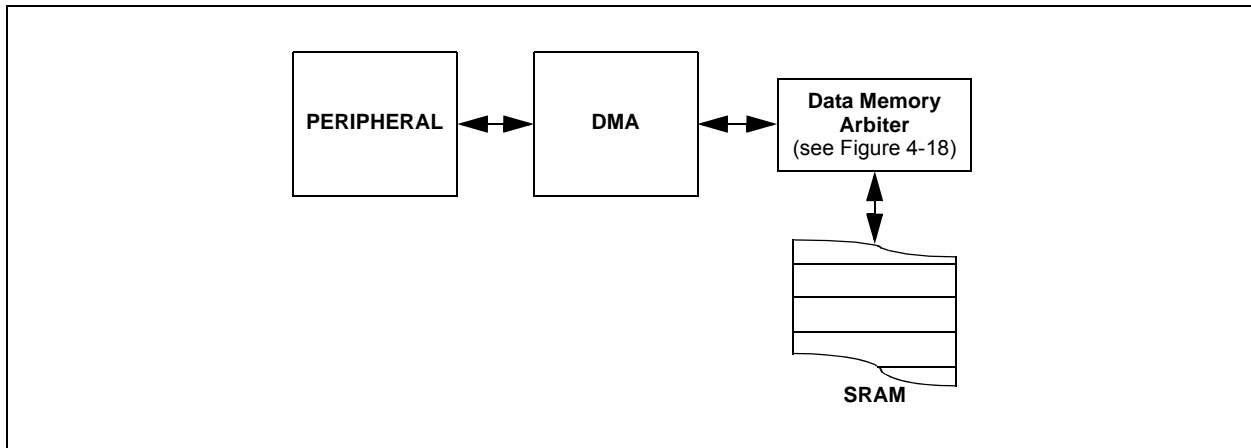
In addition, DMA can access the entire data memory space. The Data Memory Bus Arbiter is utilized when either the CPU or DMA attempts to access SRAM, resulting in potential DMA or CPU stalls.

The DMA Controller supports 4 independent channels. Each channel can be configured for transfers to or from selected peripherals. Some of the peripherals supported by the DMA Controller include:

- ECAN™
- Analog-to-Digital Converter (ADC)
- Serial Peripheral Interface (SPI)
- UART
- Input Capture
- Output Compare

Refer to Table 8-1 for a complete list of supported peripherals.

**FIGURE 8-1: DMA CONTROLLER MODULE**



**REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER<sup>(1)</sup> (CONTINUED)**

bit 4	<b>Unimplemented:</b> Read as '0'
bit 3	<b>CF:</b> Clock Fail Detect bit <sup>(3)</sup> 1 = FSCM has detected clock failure 0 = FSCM has not detected clock failure
bit 2-1	<b>Unimplemented:</b> Read as '0'
bit 0	<b>OSWEN:</b> Oscillator Switch Enable bit 1 = Requests oscillator switch to selection specified by the NOSC<2:0> bits 0 = Oscillator switch is complete

- Note 1:** Writes to this register require an unlock sequence. Refer to “**Oscillator**” (DS70580) in the “*dsPIC33/PIC24 Family Reference Manual*” (available from the Microchip web site) for details.
- 2:** Direct clock switches between any primary oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.
- 3:** This bit should only be cleared in software. Setting the bit in software (= 1) will have the same effect as an actual oscillator failure and trigger an oscillator failure trap.

**REGISTER 10-3: PMD3: PERIPHERAL MODULE DISABLE CONTROL REGISTER 3**

U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0
—	—	—	—	—	CMPMD	—	—
bit 15						bit 8	
R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
CRCMD	—	—	—	—	—	I2C2MD	—
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      **CMPMD:** Comparator Module Disable bit  
                  1 = Comparator module is disabled  
                  0 = Comparator module is enabled
- bit 9-8      **Unimplemented:** Read as '0'
- bit 7      **CRCMD:** CRC Module Disable bit  
                  1 = CRC module is disabled  
                  0 = CRC module is enabled
- bit 6-2      **Unimplemented:** Read as '0'
- bit 1      **I2C2MD:** I2C2 Module Disable bit  
                  1 = I2C2 module is disabled  
                  0 = I2C2 module is enabled
- bit 0      **Unimplemented:** Read as '0'

**REGISTER 10-4: PMD4: PERIPHERAL MODULE DISABLE CONTROL REGISTER 4**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
—	—	—	—	REFOMD	CTMUMD	—	—
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 15-4      **Unimplemented:** Read as '0'
- bit 3      **REFOMD:** Reference Clock Module Disable bit  
                  1 = Reference clock module is disabled  
                  0 = Reference clock module is enabled
- bit 2      **CTMUMD:** CTMU Module Disable bit  
                  1 = CTMU module is disabled  
                  0 = CTMU module is enabled
- bit 1-0      **Unimplemented:** Read as '0'

**NOTES:**

**REGISTER 11-15: RPINR37: PERIPHERAL PIN SELECT INPUT REGISTER 37**  
**(dsPIC33EPXXXMC20X/50X AND PIC24EPXXXMC20X DEVICES ONLY)**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	SYNC1R<6:0>							
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      **Unimplemented:** Read as '0'

bit 14-8      **SYNCl1R<6:0>:** Assign PWM Synchronization Input 1 to the Corresponding RPn Pin bits  
 (see Table 11-2 for input pin selection numbers)

1111001 = Input tied to RPI121

.

.

.

0000001 = Input tied to CMP1

0000000 = Input tied to Vss

bit 7-0      **Unimplemented:** Read as '0'



**REGISTER 16-8: PDCx: PWMx GENERATOR DUTY CYCLE REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDCx<15:8>							
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
PDCx<7:0>							
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 15-0                      **PDCx<15:0>**: PWMx Generator # Duty Cycle Value bits

**REGISTER 16-9: PHASEx: PWMx PRIMARY PHASE-SHIFT REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHASEx<15:8>							
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
PHASEx<7:0>							
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 15-0                      **PHASEx<15:0>**: PWMx Phase-Shift Value or Independent Time Base Period for the PWM Generator bits

- Note 1:** If ITB (PWMCONx<9>) = 0, the following applies based on the mode of operation:  
Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCON<11:10>) = 00, 01 or 10),  
PHASEx<15:0> = Phase-shift value for PWMxH and PWMxL outputs
- 2:** If ITB (PWMCONx<9>) = 1, the following applies based on the mode of operation:  
Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01 or 10),  
PHASEx<15:0> = Independent time base period value for PWMxH and PWMxL

**REGISTER 16-17: LEBDLYx: PWMx LEADING-EDGE BLANKING DELAY REGISTER**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	LEB<11:8>			
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
LEB<7:0>							
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 15-12 **Unimplemented:** Read as '0'

bit 11-0 **LEB<11:0>:** Leading-Edge Blanking Delay for Current-Limit and Fault Inputs bits

## 18.1 SPI Helpful Tips

1. In Frame mode, if there is a possibility that the master may not be initialized before the slave:
  - a) If FRMPOL (SPIxCON2<13>) = 1, use a pull-down resistor on  $\overline{SSx}$ .
  - b) If FRMPOL = 0, use a pull-up resistor on  $\overline{SSx}$ .

**Note:** This insures that the first frame transmission after initialization is not shifted or corrupted.

2. In Non-Framed 3-Wire mode, (i.e., not using  $\overline{SSx}$  from a master):
  - a) If CKP (SPIxCON1<6>) = 1, always place a pull-up resistor on  $\overline{SSx}$ .
  - b) If CKP = 0, always place a pull-down resistor on  $\overline{SSx}$ .

**Note:** This will insure that during power-up and initialization the master/slave will not lose Sync due to an errant SCKx transition that would cause the slave to accumulate data shift errors for both transmit and receive appearing as corrupted data.

3. FRMEN (SPIxCON2<15>) = 1 and SSEN (SPIxCON1<7>) = 1 are exclusive and invalid. In Frame mode, SCKx is continuous and the Frame Sync pulse is active on the  $\overline{SSx}$  pin, which indicates the start of a data frame.

**Note:** Not all third-party devices support Frame mode timing. Refer to the SPIx specifications in **Section 30.0 “Electrical Characteristics”** for details.

4. In Master mode only, set the SMP bit (SPIxCON1<9>) to a ‘1’ for the fastest SPIx data rate possible. The SMP bit can only be set at the same time or after the MSTEN bit (SPIxCON1<5>) is set.

To avoid invalid slave read data to the master, the user's master software must ensure enough time for slave software to fill its write buffer before the user application initiates a master write/read cycle. It is always advisable to preload the SPIxBUF Transmit register in advance of the next master transaction cycle. SPIxBUF is transferred to the SPIx Shift register and is empty once the data transmission begins.

## 18.2 SPI Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page, which can be accessed using this link, contains the latest updates and additional information.

**Note:** In the event you are not able to access the product page using the link above, enter this URL in your browser:  
<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en555464>

### 18.2.1 KEY RESOURCES

- “**Serial Peripheral Interface (SPI)**” (DS70569) in the “*dsPIC33/PIC24 Family Reference Manual*”
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related “*dsPIC33/PIC24 Family Reference Manual*” Sections
- Development Tools

**REGISTER 19-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)**

- bit 6      **STREN:** SCLx Clock Stretch Enable bit (when operating as I<sup>2</sup>C slave)  
Used in conjunction with the SCLREL bit.  
1 = Enables software or receives clock stretching  
0 = Disables software or receives clock stretching
- bit 5      **ACKDT:** Acknowledge Data bit (when operating as I<sup>2</sup>C master, applicable during master receive)  
Value that is 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<sup>2</sup>C master, applicable during master receive)  
1 = Initiates Acknowledge sequence on SDAx and SCLx pins and transmits ACKDT data bit. Hardware is clear at the end of the master Acknowledge sequence.  
0 = Acknowledge sequence is not in progress
- bit 3      **RCEN:** Receive Enable bit (when operating as I<sup>2</sup>C master)  
1 = Enables Receive mode for I<sup>2</sup>C. Hardware is clear at the end of the eighth bit of the master receive data byte.  
0 = Receive sequence is not in progress
- bit 2      **PEN:** Stop Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiates Stop condition on SDAx and SCLx pins. Hardware is clear at the end of the master Stop sequence.  
0 = Stop condition is not in progress
- bit 1      **RSEN:** Repeated Start Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiates Repeated Start condition on SDAx and SCLx pins. Hardware is clear at the end of the master Repeated Start sequence.  
0 = Repeated Start condition is not in progress
- bit 0      **SEN:** Start Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiates Start condition on SDAx and SCLx pins. Hardware is clear at the end of the master Start sequence.  
0 = Start condition is not in progress

**Note 1:** When performing master operations, ensure that the IPMIEN bit is set to '0'.

## 24.0 PERIPHERAL TRIGGER GENERATOR (PTG) MODULE

**Note 1:** This data sheet summarizes the features of the dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X and PIC24EPXXXGP/MC20X families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to “**Peripheral Trigger Generator (PTG)**” (DS70669) in the “*dsPIC33/PIC24 Family Reference Manual*”, which is available from the Microchip web site ([www.microchip.com](http://www.microchip.com)).

**2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 “Memory Organization”** in this data sheet for device-specific register and bit information.

### 24.1 Module Introduction

The Peripheral Trigger Generator (PTG) provides a means to schedule complex high-speed peripheral operations that would be difficult to achieve using software. The PTG module uses 8-bit commands, called “Steps”, that the user writes to the PTG Queue registers (PTGQUE0-PTGQUE7), which perform operations, such as wait for input signal, generate output trigger and wait for timer.

The PTG module has the following major features:

- Multiple clock sources
- Two 16-bit general purpose timers
- Two 16-bit general limit counters
- Configurable for rising or falling edge triggering
- Generates processor interrupts to include:
  - Four configurable processor interrupts
  - Interrupt on a Step event in Single-Step mode
  - Interrupt on a PTG Watchdog Timer time-out
- Able to receive trigger signals from these peripherals:
  - ADC
  - PWM
  - Output Compare
  - Input Capture
  - Op Amp/Comparator
  - INT2
- Able to trigger or synchronize to these peripherals:
  - Watchdog Timer
  - Output Compare
  - Input Capture
  - ADC
  - PWM
  - Op Amp/Comparator

**REGISTER 24-6: PTGSDLIM: PTG STEP DELAY LIMIT REGISTER<sup>(1,2)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGSDLIM<15:8>							
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
PTGSDLIM<7:0>							
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 15-0                      **PTGSDLIM<15:0>**: PTG Step Delay Limit Register bits  
 Holds a PTG Step delay value representing the number of additional PTG clocks between the start of a Step command and the completion of a Step command.

- Note 1:** A base Step delay of one PTG clock is added to any value written to the PTGSDLIM register (Step Delay = (PTGSDLIM) + 1).  
**2:** This register is read-only when the PTG module is executing Step commands (PTGEN = 1 and PTGSTRT = 1).

**REGISTER 24-7: PTGC0LIM: PTG COUNTER 0 LIMIT REGISTER<sup>(1)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC0LIM<15:8>							
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
PTGC0LIM<7:0>							
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 15-0                      **PTGC0LIM<15:0>**: PTG Counter 0 Limit Register bits  
 May be used to specify the loop count for the PTGJMPC0 Step command or as a limit register for the General Purpose Counter 0.

- Note 1:** This register is read-only when the PTG module is executing Step commands (PTGEN = 1 and PTGSTRT = 1).

**NOTES:**

**REGISTER 25-5: CMxMSKCON: COMPARATOR x MASK GATING  
CONTROL REGISTER (CONTINUED)**

bit 3	<b>ABEN:</b> AND Gate B Input Enable bit 1 = MBI is connected to AND gate 0 = MBI is not connected to AND gate
bit 2	<b>ABNEN:</b> AND Gate B Input Inverted Enable bit 1 = Inverted MBI is connected to AND gate 0 = Inverted MBI is not connected to AND gate
bit 1	<b>AAEN:</b> AND Gate A Input Enable bit 1 = MAI is connected to AND gate 0 = MAI is not connected to AND gate
bit 0	<b>AANEN:</b> AND Gate A Input Inverted Enable bit 1 = Inverted MAI is connected to AND gate 0 = Inverted MAI is not connected to AND gate



**NOTES:**

## **29.2 MPLAB XC Compilers**

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

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

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

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

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

## **29.3 MPASM Assembler**

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

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

The MPASM Assembler features include:

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

## **29.4 MPLINK Object Linker/ MPLIB Object Librarian**

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

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

The object linker/library features include:

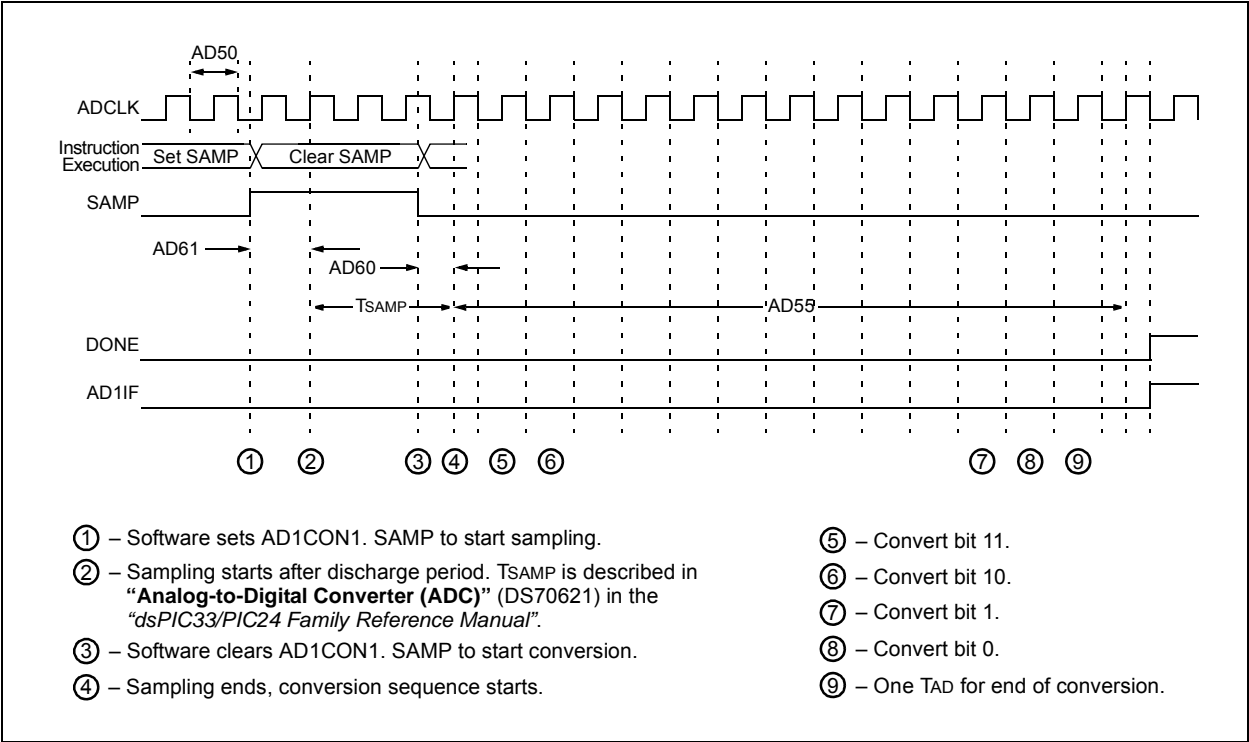
- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## **29.5 MPLAB Assembler, Linker and Librarian for Various Device Families**

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

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

FIGURE 30-36: ADC CONVERSION (12-BIT MODE) TIMING CHARACTERISTICS  
(ASAM = 0, SSRC<2:0> = 000, SSRCG = 0)



**Revision C (December 2011)**

This revision includes typographical and formatting changes throughout the data sheet text.

In addition, where applicable, new sections were added to each peripheral chapter that provide information and links to related resources, as well as helpful tips. For examples, see **Section 20.1 “UART Helpful Tips”** and **Section 3.6 “CPU Resources”**.

All occurrences of TLA were updated to VTLA throughout the document, with the exception of the pin diagrams (updated diagrams were not available at time of publication).

A new chapter, **Section 31.0 “DC and AC Device Characteristics Graphs”**, was added.

All other major changes are referenced by their respective section in Table A-2.

**TABLE A-2: MAJOR SECTION UPDATES**

Section Name	Update Description
<b>“16-bit Microcontrollers and Digital Signal Controllers (up to 256-Kbyte Flash and 32-Kbyte SRAM) with High-Speed PWM, Op amps, and Advanced Analog”</b>	The content on the first page of this section was extensively reworked to provide the reader with the key features and functionality of this device family in an “at-a-glance” format.
<b>Section 1.0 “Device Overview”</b>	Updated the dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X, and PIC24EPXXXGP/MC20X Block Diagram (see Figure 1-1), which now contains a CPU block and a reference to the CPU diagram. Updated the description and Note references in the Pinout I/O Descriptions for these pins: C1IN2-, C2IN2-, C3IN2-, OA1OUT, OA2OUT, and OA3OUT (see Table 1-1).
<b>Section 2.0 “Guidelines for Getting Started with 16-bit Digital Signal Controllers and Microcontrollers”</b>	Updated the Recommended Minimum Connection diagram (see Figure 2-1).
<b>Section 3.0 “CPU”</b>	Updated the dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X, and PIC24EPXXXGP/MC20X CPU Block Diagram (see Figure 3-1). Updated the Status register definition in the Programmer’s Model (see Figure 3-2).
<b>Section 4.0 “Memory Organization”</b>	Updated the Data Memory Maps (see Figure 4-6 and Figure 4-11). Removed the DCB<1:0> bits from the OC1CON2, OC2CON2, OC3CON2, and OC4CON2 registers in the Output Compare 1 Through Output Compare 4 Register Map (see Table 4-10). Added the TRIG1 and TRGCON1 registers to the PWM Generator 1 Register Map (see Table 4-13). Added the TRIG2 and TRGCON2 registers to the PWM Generator 2 Register Map (see Table 4-14). Added the TRIG3 and TRGCON3 registers to the PWM Generator 3 Register Map (see Table 4-15). Updated the second note in <b>Section 4.7.1 “Bit-Reversed Addressing Implementation”</b> .
<b>Section 8.0 “Direct Memory Access (DMA)”</b>	Updated the DMA Controller diagram (see Figure 8-1).
<b>Section 14.0 “Input Capture”</b>	Updated the bit values for the ICx clock source of the ICTSEL<12:10> bits in the ICxCON1 register (see Register 14-1).
<b>Section 15.0 “Output Compare”</b>	Updated the bit values for the OCx clock source of the OCTSEL<2:0> bits in the OCxCON1 register (see Register 15-1). Removed the DCB<1:0> bits from the Output Compare x Control Register 2 (see Register 15-2).

**Revision H (August 2013)**

This revision includes minor typographical and formatting changes throughout the text.

Other major changes are referenced by their respective section in Table A-6.

**TABLE A-6: MAJOR SECTION UPDATES**

Section Name	Update Description
<b>Cover Section</b>	<ul style="list-style-type: none"> <li>Adds Peripheral Pin Select (PPS) to allow Digital Function Remapping and Change Notification Interrupts to Input/Output section</li> <li>Adds heading information to 64-Pin TQFP</li> </ul>
<b>Section 4.0 “Memory Organization”</b>	<ul style="list-style-type: none"> <li>Corrects Reset values for ANSELE, TRISF, TRISC, ANSELC and TRISA</li> <li>Corrects address range from 0x2FFF to 0x7FFF</li> <li>Corrects DSRPAG and DSWPAG (now 3 hex digits)</li> <li>Changes Call Stack Frame from &lt;15:1&gt; to PC&lt;15:0&gt;</li> <li>Word length in Figure 4-20 is changed to 50 words for clarity</li> </ul>
<b>Section 5.0 “Flash Program Memory”</b>	<ul style="list-style-type: none"> <li>Corrects descriptions of NVM registers</li> </ul>
<b>Section 9.0 “Oscillator Configuration”</b>	<ul style="list-style-type: none"> <li>Removes resistor from Figure 9-1</li> <li>Adds Fast RC Oscillator with Divide-by-16 (FRCDIV16) row to Table 9-1</li> <li>Removes incorrect information from ROI bit in Register 9-2</li> </ul>
<b>Section 14.0 “Input Capture”</b>	<ul style="list-style-type: none"> <li>Changes 31 user-selectable Trigger/Sync interrupts to 19 user-selectable Trigger/Sync interrupts</li> <li>Corrects ICTSEL&lt;12:10&gt; bits (now ICTSEL&lt;2:0&gt;)</li> </ul>
<b>Section 17.0 “Quadrature Encoder Interface (QEI) Module (dsPIC33EPXXXMC20X/50X and PIC24EPXXXMC20X Devices Only)”</b>	<ul style="list-style-type: none"> <li>Corrects QCAPEN bit description</li> </ul>
<b>Section 19.0 “Inter-Integrated Circuit™ (I<sup>2</sup>C™)”</b>	<ul style="list-style-type: none"> <li>Adds note to clarify that 100kbit/sec operation of I<sup>2</sup>C is not possible at high processor speeds</li> </ul>
<b>Section 22.0 “Charge Time Measurement Unit (CTMU)”</b>	<ul style="list-style-type: none"> <li>Clarifies Figure 22-1 to accurately reflect peripheral behavior</li> </ul>
<b>Section 23.0 “10-Bit/12-Bit Analog-to-Digital Converter (ADC)”</b>	<ul style="list-style-type: none"> <li>Correct Figure 23-1 (changes CH123x to CH123Sx)</li> </ul>
<b>Section 24.0 “Peripheral Trigger Generator (PTG) Module”</b>	<ul style="list-style-type: none"> <li>Adds footnote to Register 24-1 (In order to operate with CVRSS=1, at least one of the comparator modules must be enabled).</li> </ul>
<b>Section 25.0 “Op Amp/Comparator Module”</b>	<ul style="list-style-type: none"> <li>Adds note to Figure 25-3 (In order to operate with CVRSS=1, at least one of the comparator modules must be enabled)</li> <li>Adds footnote to Register 25-2 (COE is not available when OPMODE (CMxCON&lt;10&gt;) = 1)</li> </ul>
<b>Section 27.0 “Special Features”</b>	<ul style="list-style-type: none"> <li>Corrects the bit description for FNOSC&lt;2:0&gt;</li> </ul>
<b>Section 30.0 “Electrical Characteristics”</b>	<ul style="list-style-type: none"> <li>Corrects 512K part power-down currents based on test data</li> <li>Corrects WDT timing limits based on LPRC oscillator tolerance</li> </ul>
<b>Section 31.0 “High-Temperature Electrical Characteristics”</b>	<ul style="list-style-type: none"> <li>Adds Table 31-5 (DC Characteristics: Idle Current (I<sub>IDLE</sub>))</li> </ul>