



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	40 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	85
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 32x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic24hj64gp210at-i-pt">https://www.e-xfl.com/product-detail/microchip-technology/pic24hj64gp210at-i-pt</a>

# PIC24HJXXXGPX06A/X08A/X10A

## 5.4.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of program Flash memory at a time. To do this, it is necessary to erase the 8-row erase page that contains the desired row. The general process is:

1. Read eight rows of program memory (512 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase the page (see Example 5-1):
  - a) Set the NVMOP bits (NVMCON<3:0>) to '0010' to configure for block erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
  - b) Write the starting address of the page to be erased into the TBLPAG and W registers.
  - c) Perform a dummy table write operation (TBLWTL) to any address within the page that needs to be erased.
  - d) Write 0x55 to NVMKEY.
  - e) Write 0xAA to NVMKEY.
  - f) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 64 instructions from data RAM into the program memory buffers (see Example 5-2).
5. Write the program block to Flash memory:
  - a) Set the NVMOP bits to '0001' to configure for row programming. Clear the ERASE bit and set the WREN bit.
  - b) Write 0x55 to NVMKEY.
  - c) Write 0xAA to NVMKEY.
  - d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.
6. Repeat steps 4 and 5, using the next available 64 instructions from the block in data RAM by incrementing the value in TBLPAG, until all 512 instructions are written back to Flash memory.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as shown in Example 5-3.

### EXAMPLE 5-1: ERASING A PROGRAM MEMORY PAGE

```
; Set up NVMCON for block erase operation
MOV    #0x4042, W0          ;
MOV    W0, NVMCON           ; Initialize NVMCON
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR), W0 ;
MOV    W0, TBLPAG           ; Initialize PM Page Boundary SFR
MOV    #tbloffset(PROG_ADDR), W0 ; Initialize in-page EA<15:0> pointer
TBLWTL W0, [W0]             ; Set base address of erase block
DISI    #5                  ; Block all interrupts with priority <7
                                ; for next 5 instructions

MOV    #0x55, W0
MOV    W0, NVMKEY           ; Write the 55 key
MOV    #0xAA, W1
MOV    W1, NVMKEY           ; Write the AA key
BSET   NVMCON, #WR          ; Start the erase sequence
NOP                                ; Insert two NOPs after the erase
NOP                                ; command is asserted
```

**Note:** A program memory page erase operation is set up by performing a dummy table write (TBLWTL) operation to any address within the page. This methodology is different from the page erase operation on dsPIC30F/33F devices in which the erase page was selected using a dedicated pair of registers (NVMADRU and NVMADR).

# PIC24HJXXXGPX06A/X08A/X10A

## REGISTER 7-12: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T6IE	DMA4IE	—	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
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
IC5IE	IC4IE	IC3IE	DMA3IE	C1IE	C1RXIE	SPI2IE	SPI2EIE
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	<b>T6IE:</b> Timer6 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 14	<b>DMA4IE:</b> DMA Channel 4 Data Transfer Complete Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 13	<b>Unimplemented:</b> Read as '0'
bit 12	<b>OC8IE:</b> Output Compare Channel 8 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 11	<b>OC7IE:</b> Output Compare Channel 7 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 10	<b>OC6IE:</b> Output Compare Channel 6 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 9	<b>OC5IE:</b> Output Compare Channel 5 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 8	<b>IC6IE:</b> Input Capture Channel 6 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 7	<b>IC5IE:</b> Input Capture Channel 5 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 6	<b>IC4IE:</b> Input Capture Channel 4 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 5	<b>IC3IE:</b> Input Capture Channel 3 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 4	<b>DMA3IE:</b> DMA Channel 3 Data Transfer Complete Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 3	<b>C1IE:</b> ECAN1 Event Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled

# PIC24HJXXXGPX06A/X08A/X10A

## REGISTER 7-20: IPC5: INTERRUPT PRIORITY CONTROL REGISTER 5

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC8IP<2:0>			—	IC7IP<2:0>		
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
—	AD2IP<2:0>			—	INT1IP<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 **Unimplemented:** Read as '0'

bit 14-12 **IC8IP<2:0>:** Input Capture Channel 8 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•  
•  
•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **IC7IP<2:0>:** Input Capture Channel 7 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 **AD2IP<2:0>:** ADC2 Conversion Complete 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 **INT1IP<2:0>:** External Interrupt 1 Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•  
•  
•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

# PIC24HJXXXGPX06A/X08A/X10A

## REGISTER 7-25: IPC10: INTERRUPT PRIORITY CONTROL REGISTER 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP<2:0>			—	OC6IP<2:0>		
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
—	OC5IP<2:0>			—	IC6IP<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 **Unimplemented:** Read as '0'

bit 14-12 **OC7IP<2:0>:** Output Compare Channel 7 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•

•

•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC6IP<2:0>:** Output Compare Channel 6 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 **OC5IP<2:0>:** Output Compare Channel 5 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 **IC6IP<2:0>:** Input Capture Channel 6 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•

•

•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

# PIC24HJXXXGPX06A/X08A/X10A

---

NOTES:

## 14.0 INPUT CAPTURE

**Note 1:** This data sheet summarizes the features of the PIC24HJXXXGPX06A/X08A/X10A family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F/PIC24H Family Reference Manual”, **Section 12. “Input Capture”** (DS70198), which is available from the Microchip web site (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 input capture module is useful in applications requiring frequency (period) and pulse measurement. The PIC24HJXXXGPX06A/X08A/X10A devices support up to eight input capture channels.

The input capture module captures the 16-bit value of the selected Time Base register when an event occurs at the ICx pin. The events that cause a capture event are listed below in three categories:

- Simple Capture Event modes:
  - Capture timer value on every falling edge of input at ICx pin
  - Capture timer value on every rising edge of input at ICx pin

- Capture timer value on every edge (rising and falling)
- Prescaler Capture Event modes:
  - Capture timer value on every 4th rising edge of input at ICx pin
  - Capture timer value on every 16th rising edge of input at ICx pin

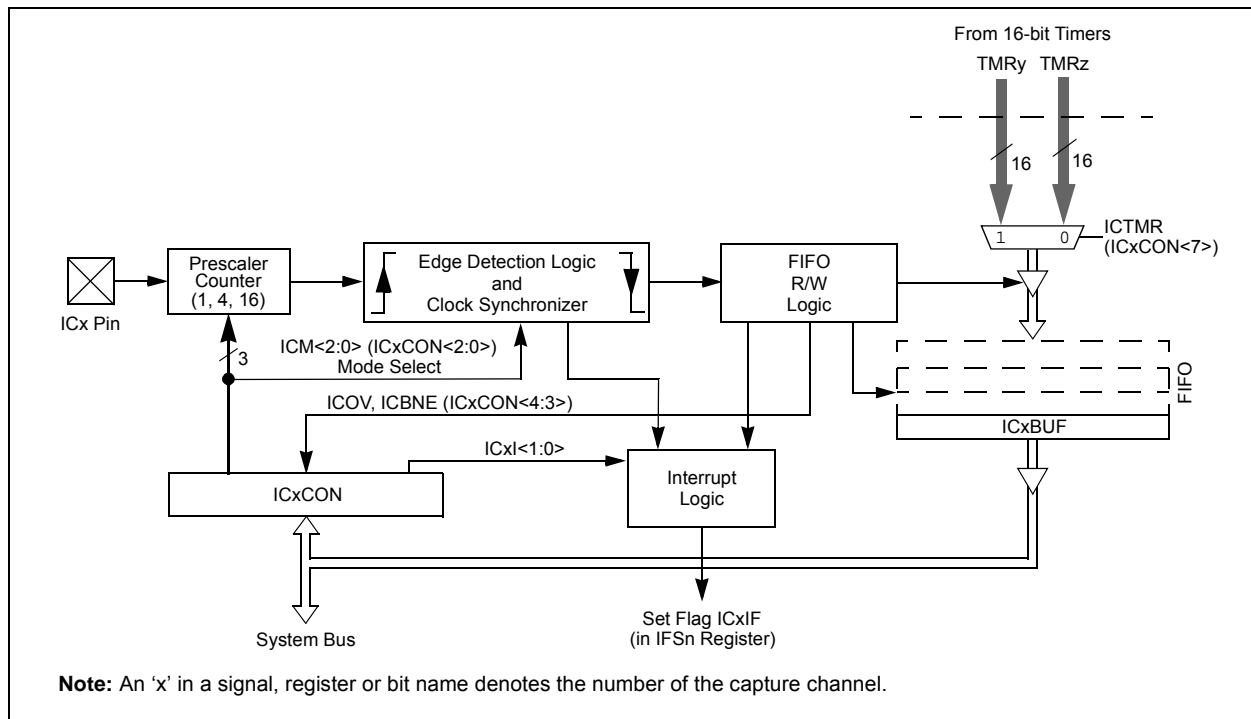
Each input capture channel can select between one of two 16-bit timers (Timer2 or Timer3) for the time base. The selected timer can use either an internal or external clock.

Other operational features include:

- Device wake-up from capture pin during CPU Sleep and Idle modes
- Interrupt on input capture event
- 4-word FIFO buffer for capture values
  - Interrupt optionally generated after 1, 2, 3 or 4 buffer locations are filled
- Input capture can also be used to provide additional sources of external interrupts.

**Note:** Only IC1 and IC2 can trigger a DMA data transfer. If DMA data transfers are required, the FIFO buffer size must be set to 1 (IC1<1:0> = 00).

**FIGURE 14-1: INPUT CAPTURE BLOCK DIAGRAM**



## 15.0 OUTPUT COMPARE

**Note 1:** This data sheet summarizes the features of the PIC24HJXXXGPX06A/X08A/X10A families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “*dsPIC33F/PIC24H Family Reference Manual*”, **Section 13. “Output Compare”** (DS70209), which is available on 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.

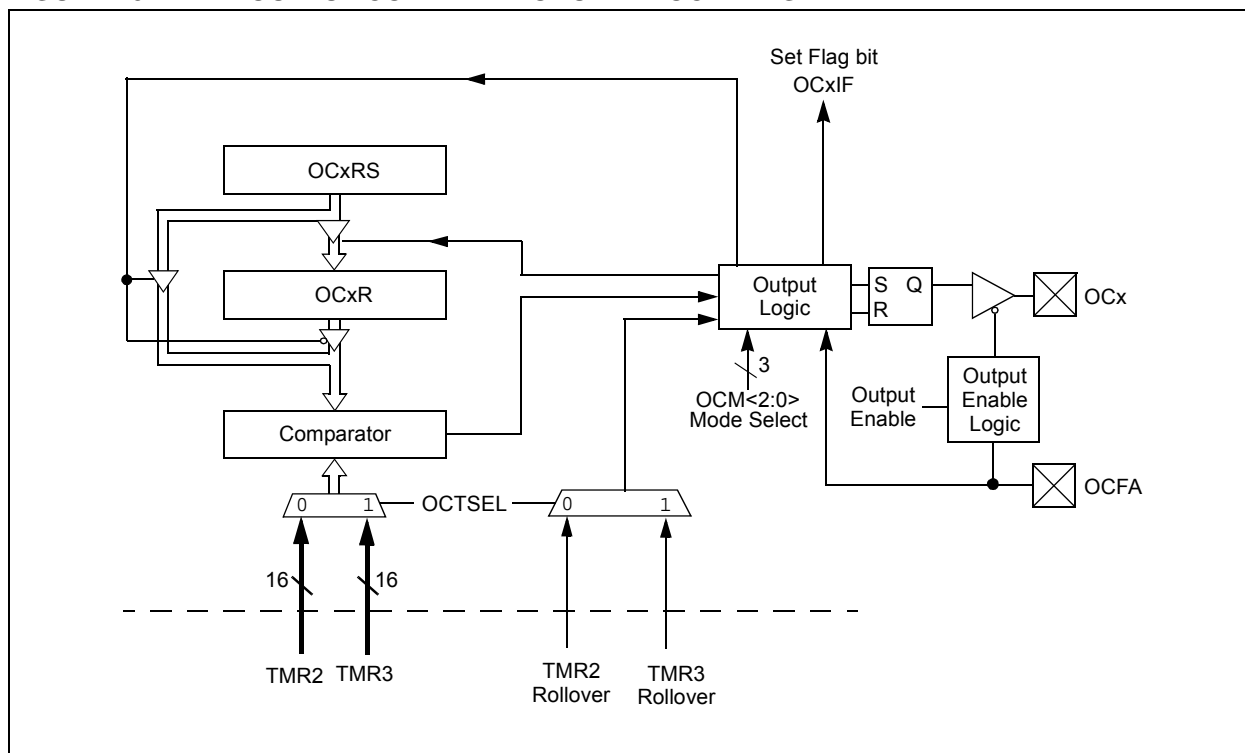
The state of the output pin changes when the timer value matches the Compare register value. The output compare module generates either a single output pulse, or a sequence of output pulses, by changing the state of the output pin on the compare match events. The output compare module can also generate interrupts on compare match events.

The output compare module has multiple operating modes:

- Active-Low One-Shot mode
- Active-High One-Shot mode
- Toggle mode
- Delayed One-Shot mode
- Continuous Pulse mode
- PWM mode without Fault Protection
- PWM mode with Fault Protection

The output compare module can select either Timer2 or Timer3 for its time base. The module compares the value of the timer with the value of one or two Compare registers depending on the operating mode selected.

**FIGURE 15-1: OUTPUT COMPARE MODULE BLOCK DIAGRAM**





# PIC24HJXXXGPX06A/X08A/X10A

## 16.3 SPI Control Registers

**REGISTER 16-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER**

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
SPIEN	—	SPISIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/C-0	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7							bit 0

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

- bit 15 **SPIEN:** SPIx Enable bit  
1 = Enables module and configures SCKx, SDOx, SDIx and  $\overline{SSx}$  as serial port pins  
0 = Disables module
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **SPISIDL:** Stop in Idle Mode bit  
1 = Discontinue module operation when device enters Idle mode  
0 = Continue module operation in Idle mode
- bit 12-7 **Unimplemented:** Read as '0'
- bit 6 **SPIROV:** Receive Overflow Flag bit  
1 = A new byte/word is completely received and discarded. The user software has not read the previous data in the SPIxBUF register  
0 = No overflow has occurred
- bit 5-2 **Unimplemented:** Read as '0'
- bit 1 **SPITBF:** SPIx Transmit Buffer Full Status bit  
1 = Transmit not yet started, SPIxTXB is full  
0 = Transmit started, SPIxTXB is empty  
Automatically set in hardware when CPU writes SPIxBUF location, loading SPIxTXB.  
Automatically cleared in hardware when SPIx module transfers data from SPIxTXB to SPIxSR.
- bit 0 **SPIRBF:** SPIx Receive Buffer Full Status bit  
1 = Receive complete, SPIxRXB is full  
0 = Receive is not complete, SPIxRXB is empty  
Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB.  
Automatically cleared in hardware when core reads SPIxBUF location, reading SPIxRXB.

# PIC24HJXXXGPX06A/X08A/X10A

---

## REGISTER 17-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

---

- bit 5      **ACKDT:** Acknowledge Data bit (when operating as I<sup>2</sup>C master, applicable during master receive)  
Value that will be transmitted when the software initiates an Acknowledge sequence.  
1 = Send NACK during Acknowledge  
0 = Send ACK during Acknowledge
- bit 4      **ACKEN:** Acknowledge Sequence Enable bit  
(when operating as I<sup>2</sup>C master, applicable during master receive)  
1 = Initiate Acknowledge sequence on SDAx and SCLx pins and transmit ACKDT data bit.  
Hardware clear at end of master Acknowledge sequence.  
0 = Acknowledge sequence not in progress
- bit 3      **RCEN:** Receive Enable bit (when operating as I<sup>2</sup>C master)  
1 = Enables Receive mode for I<sup>2</sup>C. Hardware clear at end of eighth bit of master receive data byte.  
0 = Receive sequence not in progress
- bit 2      **PEN:** Stop Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiate Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence.  
0 = Stop condition not in progress
- bit 1      **RSEN:** Repeated Start Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiate Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of  
master Repeated Start sequence.  
0 = Repeated Start condition not in progress
- bit 0      **SEN:** Start Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiate Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence.  
0 = Start condition not in progress

# PIC24HJXXXGPX06A/X08A/X10A

## REGISTER 19-1: CiCTRL1: ECAN™ MODULE CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	r-0	R/W-1	R/W-0	R/W-0
—	—	CSIDL	ABAT	—	REQOP<2:0>		
bit 15						bit 8	

R-1	R-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
OPMODE<2:0>			—	CANCAP	—	—	WIN
bit 7							bit 0

<b>Legend:</b>	r = Bit is Reserved		
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      **CSIDL:** Stop in Idle Mode bit  
1 = Discontinue module operation when device enters Idle mode  
0 = Continue module operation in Idle mode
- bit 12      **ABAT:** Abort All Pending Transmissions bit  
1 = Signal all transmit buffers to abort transmission  
0 = Module will clear this bit when all transmissions are aborted
- bit 11      **Reserved:** Do not use
- bit 10-8      **REQOP<2:0>:** Request Operation Mode bits  
111 = Set Listen All Messages mode  
110 = Reserved – do not use  
101 = Reserved – do not use  
100 = Set Configuration mode  
011 = Set Listen Only Mode  
010 = Set Loopback mode  
001 = Set Disable mode  
000 = Set Normal Operation mode
- bit 7-5      **OPMODE<2:0>:** Operation Mode bits  
111 = Module is in Listen All Messages mode  
110 = Reserved  
101 = Reserved  
100 = Module is in Configuration mode  
011 = Module is in Listen Only mode  
010 = Module is in Loopback mode  
001 = Module is in Disable mode  
000 = Module is in Normal Operation mode
- bit 4      **Unimplemented:** Read as '0'
- bit 3      **CANCAP:** CAN Message Receive Timer Capture Event Enable bit  
1 = Enable input capture based on CAN message receive  
0 = Disable CAN capture
- bit 2-1      **Unimplemented:** Read as '0'
- bit 0      **WIN:** SFR Map Window Select bit  
1 = Use filter window  
0 = Use buffer window

# PIC24HJXXXGPX06A/X08A/X10A

## REGISTER 19-11: CiFEN1: ECAN™ MODULE ACCEPTANCE FILTER ENABLE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8
bit 15						bit 8	

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
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

**FLTENn:** Enable Filter n (0-15) to Accept Messages bits

1 = Enable Filter n

0 = Disable Filter n

# PIC24HJXXXGPX06A/X08A/X10A

## REGISTER 20-7: ADxCSSH: ADCx INPUT SCAN SELECT REGISTER HIGH<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
CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24
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
CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16
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                      **CSS<31:16>**: ADC Input Scan Selection bits

- 1 = Select ANx for input scan
- 0 = Skip ANx for input scan

**Note 1:** On devices without 32 analog inputs, all ADxCSSH bits may be selected by user. However, inputs selected for scan without a corresponding input on device will convert VREFL.

**2:** CSSx = ANx, where x = 16 through 31.

## REGISTER 20-8: ADxCSSL: ADCx INPUT SCAN SELECT REGISTER LOW<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
CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
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
CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
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                      **CSS<15:0>**: ADC Input Scan Selection bits

- 1 = Select ANx for input scan
- 0 = Skip ANx for input scan

**Note 1:** On devices without 16 analog inputs, all ADxCSSL bits may be selected by user. However, inputs selected for scan without a corresponding input on device will convert VREF-.

**2:** CSSx = ANx, where x = 0 through 15.

# PIC24HJXXXGPX06A/X08A/X10A

## 21.4 Watchdog Timer (WDT)

For PIC24HJXXXGPX06A/X08A/X10A devices, the WDT is driven by the LPRC oscillator. When the WDT is enabled, the clock source is also enabled.

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

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

The WDT, prescaler and postscaler are reset:

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

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

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

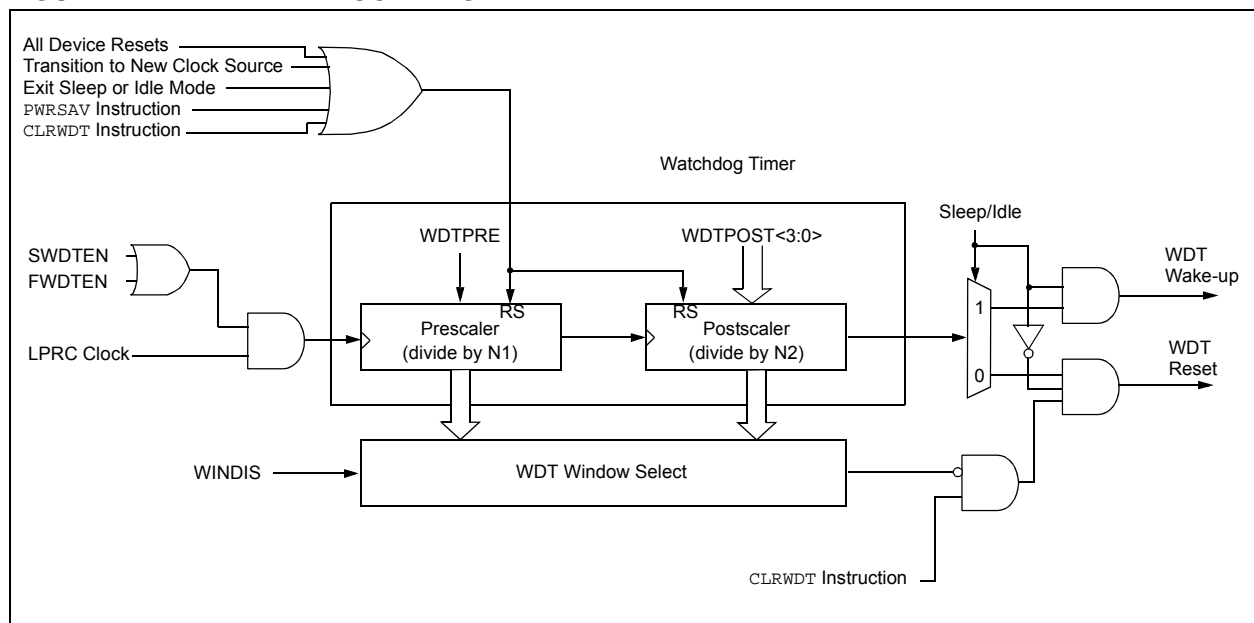
**Note:** The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

The WDT is enabled or disabled by the FWDTEN Configuration bit in the FWDT Configuration register. When the FWDTEN Configuration bit is set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN Configuration bit has been programmed to '0'. The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user to enable the WDT for critical code segments and disable the WDT during non-critical segments for maximum power savings.

**Note:** If the WINDIS bit (FWDT<6>) is cleared, the CLRWDT instruction should be executed by the application software only during the last 1/4 of the WDT period. This CLRWDT window can be determined by using a timer. If a CLRWDT instruction is executed before this window, a WDT Reset occurs.

**FIGURE 21-2: WDT BLOCK DIAGRAM**



## 22.0 INSTRUCTION SET SUMMARY

**Note:** This data sheet summarizes the features of the PIC24HJXXXGPX06A/X08A/X10A families of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section in the “*dsPIC33F/PIC24H Family Reference Manual*”, which is available from the Microchip web site ([www.microchip.com](http://www.microchip.com)).

The PIC24H instruction set is identical to that of the PIC24F, and is a subset of the dsPIC30F/33F instruction set.

Most instructions are a single program memory word (24 bits). Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word, divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into five basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- DSP operations
- Control operations

Table 22-1 shows the general symbols used in describing the instructions.

The PIC24H instruction set summary in Table 22-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand which is typically a register ‘Wb’ without any address modifier
- The second source operand which is typically a register ‘Ws’ with or without an address modifier
- The destination of the result which is typically a register ‘Wd’ with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value ‘f’
- The destination, which could either be the file register ‘f’ or the W0 register, which is denoted as ‘WREG’

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of ‘Ws’ or ‘f’)
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register ‘Wb’)

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of ‘k’)
- The W register or file register where the literal value is to be loaded (specified by ‘Wb’ or ‘f’)

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand which is a register ‘Wb’ without any address modifier
- The second source operand which is a literal value
- The destination of the result (only if not the same as the first source operand) which is typically a register ‘Wd’ with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double word instructions, which were made double word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSBs are ‘0’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes and RETURN/RETFIE instructions, which are single-word instructions but take two or three cycles. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or double word instruction. Moreover, double word moves require two cycles. The double word instructions execute in two instruction cycles.

**Note:** For more details on the instruction set, refer to the “*16-bit MCU and DSC Programmer’s Reference Manual*” (DS70157).

# PIC24HJXXXGPX06A/X08A/X10A

**TABLE 24-32: SPIx SLAVE MODE (FULL-DUPLEX, CKE = 1, CKP = 0, SMP = 0) TIMING REQUIREMENTS**

AC CHARACTERISTICS			Standard Operating Conditions: 2.4V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Typ <sup>(2)</sup>	Max	Units	Conditions
SP70	TscP	Maximum SCK Input Frequency	—	—	15	MHz	See <b>Note 3</b>
SP72	TscF	SCKx Input Fall Time	—	—	—	ns	See parameter DO32 and <b>Note 4</b>
SP73	TscR	SCKx Input Rise Time	—	—	—	ns	See parameter DO31 and <b>Note 4</b>
SP30	TdoF	SDOx Data Output Fall Time	—	—	—	ns	See parameter DO32 and <b>Note 4</b>
SP31	TdoR	SDOx Data Output Rise Time	—	—	—	ns	See parameter DO31 and <b>Note 4</b>
SP35	Tsch2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	6	20	ns	—
SP36	TdoV2scH, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	—	—	ns	—
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	30	—	—	ns	—
SP41	Tsch2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	30	—	—	ns	—
SP50	TssL2scH, TssL2scL	$\overline{SSx} \downarrow$ to SCKx $\uparrow$ or SCKx Input	120	—	—	ns	—
SP51	TssH2doZ	$\overline{SSx} \uparrow$ to SDOx Output High-Impedance <sup>(4)</sup>	10	—	50	ns	—
SP52	Tsch2ssH TscL2ssH	$\overline{SSx}$ after SCKx Edge	1.5 TCY + 40	—	—	ns	See <b>Note 4</b>
SP60	TssL2doV	SDOx Data Output Valid after $\overline{SSx}$ Edge	—	—	50	ns	—

**Note 1:** These parameters are characterized, but are not tested in manufacturing.

**2:** Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

**3:** The minimum clock period for SCKx is 66.7 ns. Therefore, the SCK clock generated by the Master must not violate this specification.

**4:** Assumes 50 pF load on all SPIx pins.



# PIC24HJXXXGPX06A/X08A/X10A

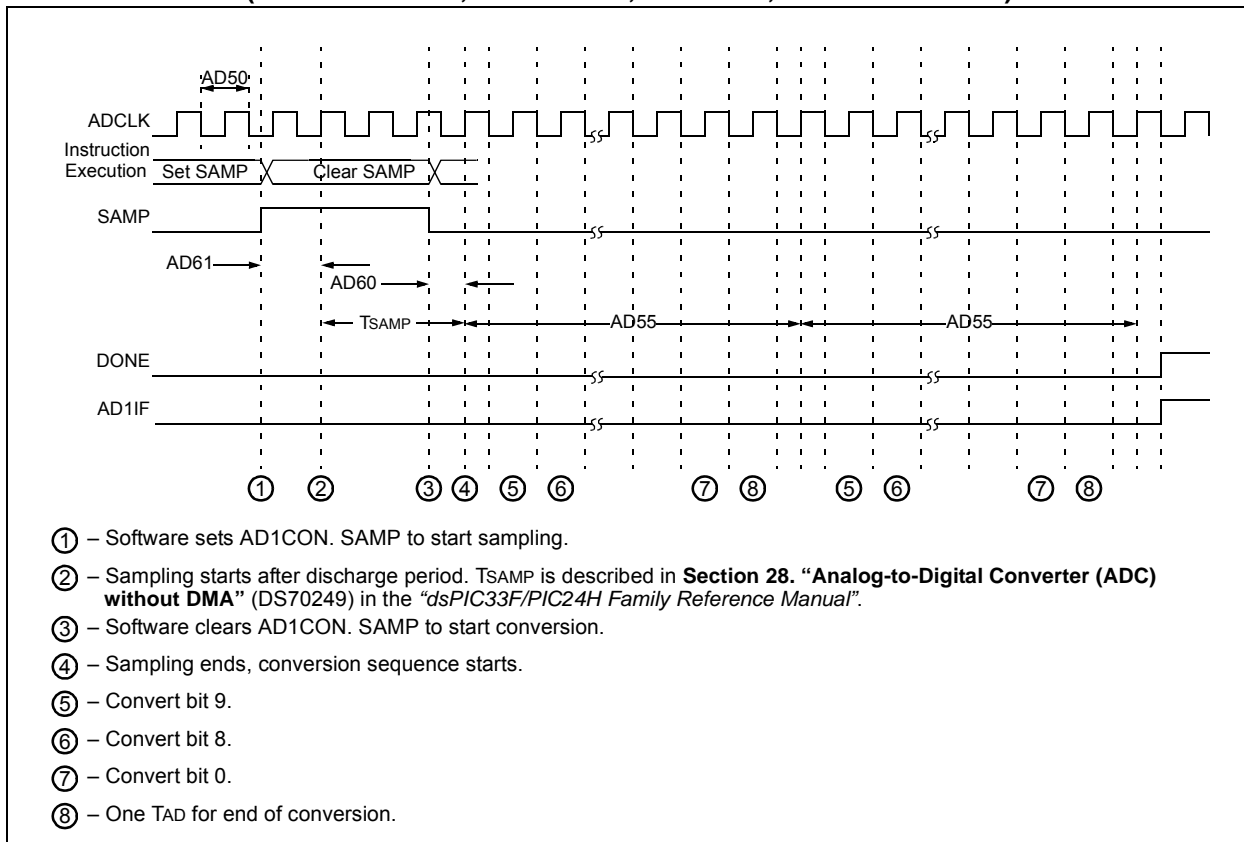
**TABLE 24-39: ADC MODULE SPECIFICATIONS**

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended				
Param No.	Symbo l	Characteristic	Min.	Typ	Max.	Units	Conditions
<b>Device Supply</b>							
AD01	AVDD	Module VDD Supply	Greater of $V_{DD} - 0.3$ or 3.0	—	Lesser of $V_{DD} + 0.3$ or 3.6	V	—
AD02	AVSS	Module VSS Supply	$V_{SS} - 0.3$	—	$V_{SS} + 0.3$	V	—
<b>Reference Inputs</b>							
AD05	VREFH	Reference Voltage High	$AV_{SS} + 2.5$	—	$AV_{DD}$	V	$V_{REFH} = AV_{DD}$ $V_{REFL} = AV_{SS} = 0$
AD05a			3.0	—	3.6	V	
AD06	VREFL	Reference Voltage Low	$AV_{SS}$	—	$AV_{DD} - 2.5$	V	$V_{REFH} = AV_{DD}$ $V_{REFL} = AV_{SS} = 0$
AD06a			0	—	0	V	
AD07	VREF	Absolute Reference Voltage	2.5	—	3.6	V	$V_{REF} = V_{REFH} - V_{REFL}$
AD08	IREF	Current Drain	—	—	10	$\mu\text{A}$	ADC off
AD08a	IAD	Operating Current	—	7.0	9.0	mA	10-bit ADC mode, See <b>Note 1</b>
			—	2.7	3.2	mA	12-bit ADC mode, See <b>Note 1</b>
<b>Analog Input</b>							
AD12	VINH	Input Voltage Range VINH	VINL	—	VREFH	V	This voltage reflects Sample and Hold Channels 0, 1, 2, and 3 (CH0-CH3), positive input
AD13	VINL	Input Voltage Range VINL	VREFL	—	$AV_{SS} + 1V$	V	This voltage reflects Sample and Hold Channels 0, 1, 2, and 3 (CH0-CH3), negative input
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	200	$\Omega$	10-bit ADC
			—	—	200	$\Omega$	12-bit ADC

**Note 1:** These parameters are not characterized or tested in manufacturing.

# PIC24HJXXXGPX06A/X08A/X10A

**FIGURE 24-23: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS**  
(CHPS<1:0> = 01, SIMSAM = 0, ASAM = 0, SSRC<2:0> = 000)



**FIGURE 24-24: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS** (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 1, SSRC<2:0> = 111, SAMC<4:0> = 00001)

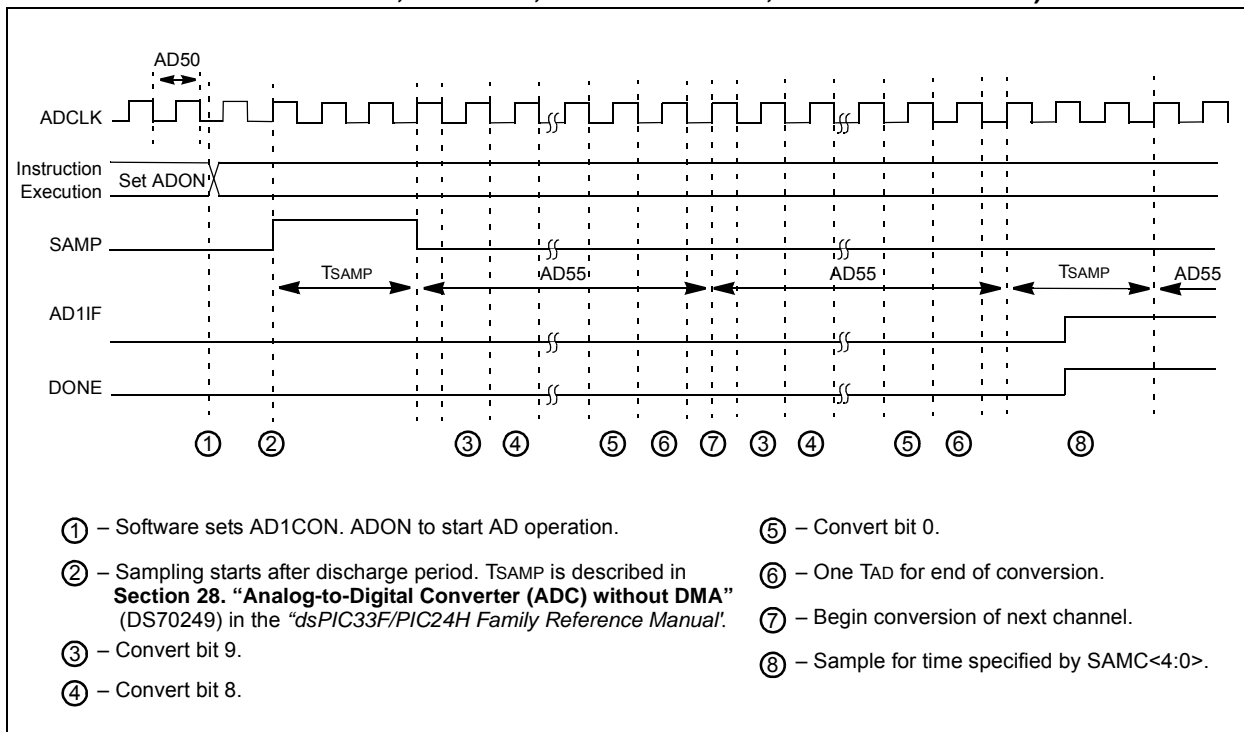


FIGURE 26-9: TYPICAL FRC FREQUENCY @ VDD = 3.3V

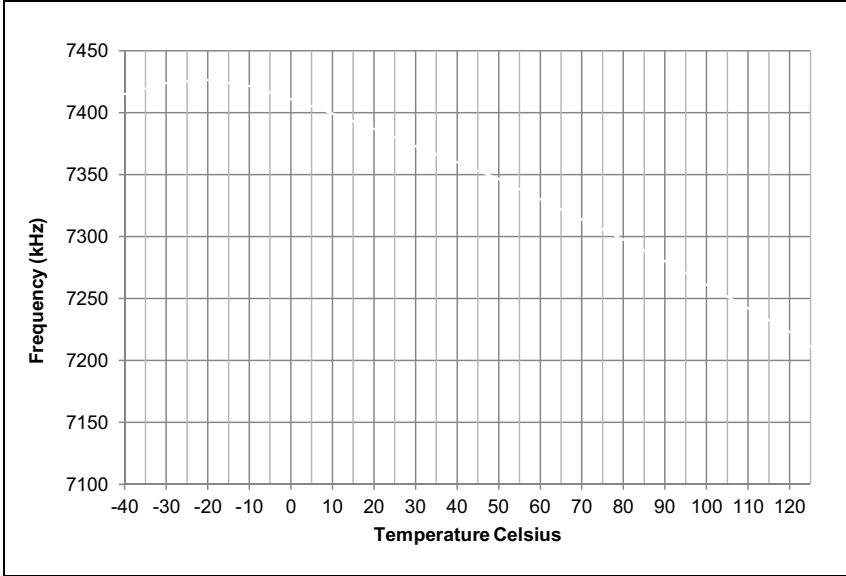
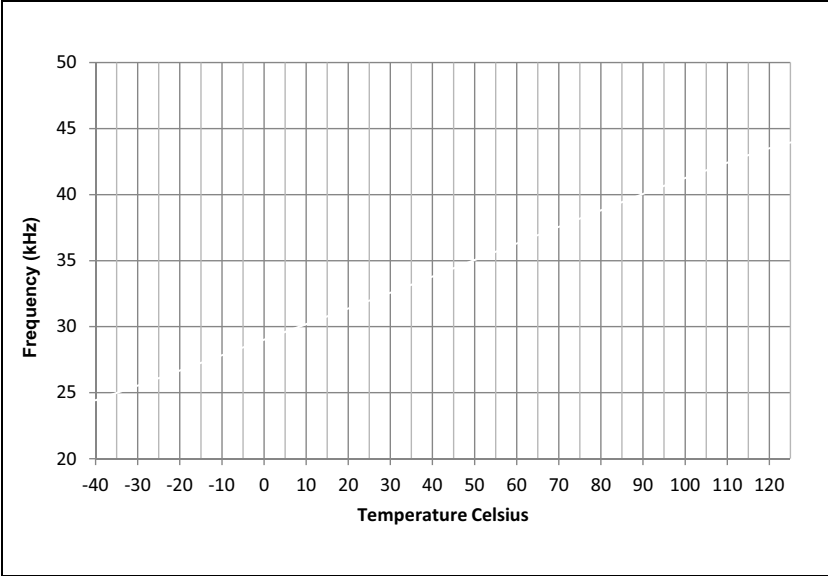


FIGURE 26-10: TYPICAL LPRC FREQUENCY @ VDD = 3.3V



# PIC24HJXXXGPX06A/X08A/X10A

## INDEX

### A

AC Characteristics .....	252, 291
ADC Module .....	294
ADC Module (10-bit Mode) .....	294
ADC Module (12-bit Mode) .....	294
Internal RC Accuracy .....	254
Load Conditions .....	252, 291
ADC Module .....	
ADC1 Register Map .....	42
ADC2 Register Map .....	42
Alternate Interrupt Vector Table (AIVT) .....	69
Analog-to-Digital Converter .....	207
DMA .....	207
Initialization .....	207
Key Features .....	207
Arithmetic Logic Unit (ALU) .....	28
Assembler .....	
MPASM Assembler .....	238
Automatic Clock Stretch .....	166

### B

Block Diagrams .....	
16-bit Timer1 Module .....	145
ADC1 Module .....	208
Connections for On-Chip Voltage Regulator .....	226
ECAN Module .....	180
Input Capture .....	153
Output Compare .....	155
PIC24H .....	16
PIC24H CPU Core .....	24
PIC24H Oscillator System Diagram .....	123
PIC24H PLL .....	125
Reset System .....	65
Shared Port Structure .....	141
SPI .....	159
Timer2 (16-bit) .....	149
Timer2/3 (32-bit) .....	148
UART .....	173
Watchdog Timer (WDT) .....	227

### C

C Compilers .....	
MPLAB C18 .....	238
Clock Switching .....	131
Enabling .....	131
Sequence .....	131
Code Examples .....	
Erasing a Program Memory Page .....	62
Initiating a Programming Sequence .....	63
Loading Write Buffers .....	63
Port Write/Read .....	142
PWRSAV Instruction Syntax .....	133
Code Protection .....	221, 228
Configuration Bits .....	221
Description (Table) .....	222
Configuration Register Map .....	221
Configuring Analog Port Pins .....	142
CPU .....	
Control Register .....	26
CPU Clocking System .....	124
PLL Configuration .....	124
Selection .....	124
Sources .....	124
Customer Change Notification Service .....	321

Customer Notification Service .....	321
Customer Support .....	321

### D

Data Address Space .....	31
Alignment .....	31
Memory Map for PIC24HJXXXGPX06A/X08A/X10A Devices with 16 KB RAM .....	33
Memory Map for PIC24HJXXXGPX06A/X08A/X10A Devices with 8 KB RAM .....	32
Near Data Space .....	31
Software Stack .....	53
Width .....	31
DC and AC Characteristics .....	
Graphs and Tables .....	297
DC Characteristics .....	242
Doze Current (IDOE) .....	289
High Temperature .....	288
I/O Pin Input Specifications .....	248
I/O Pin Output Specifications .....	250, 290
Idle Current (IDOE) .....	247
Idle Current (IDLE) .....	245
Operating Current (IDD) .....	244
Operating MIPS vs. Voltage .....	288
Power-Down Current (IPD) .....	246
Power-down Current (IPD) .....	288
Program Memory .....	251
Temperature and Voltage .....	288
Temperature and Voltage Specifications .....	243
Thermal Operating Conditions .....	288
Development Support .....	237
DMA Module .....	
DMA Register Map .....	43
DMAC Registers .....	114
DMAxCNT .....	114
DMAxCON .....	114
DMAxPAD .....	114
DMAxREQ .....	114
DMAxSTA .....	114
DMAxSTB .....	114

### E

ECAN Module .....	
CiFMSKSEL2 register .....	199
ECAN1 Register Map (C1CTRL1.WIN = 0 or 1) .....	44
ECAN1 Register Map (C1CTRL1.WIN = 0) .....	45
ECAN1 Register Map (C1CTRL1.WIN = 1) .....	45
ECAN2 Register Map (C2CTRL1.WIN = 0 or 1) .....	47
ECAN2 Register Map (C2CTRL1.WIN = 0) .....	47
ECAN2 Register Map (C2CTRL1.WIN = 1) .....	48
Frame Types .....	179
Modes of Operation .....	181
Overview .....	179
ECAN Registers .....	
Filter 15-8 Mask Selection Register (CiFMSKSEL2) .....	199
Electrical Characteristics .....	241
AC .....	252, 291
Enhanced CAN Module .....	179
Equations .....	
Device Operating Frequency .....	124
FOSC Calculation .....	124
XT with PLL Mode Example .....	125
Errata .....	13

# PIC24HJXXXGPX06A/X08A/X10A

## F

Flash Program Memory .....	59
Control Registers .....	60
Operations .....	60
Programming Algorithm .....	62
RTSP Operation .....	60
Table Instructions .....	59
Flexible Configuration .....	221
FSCM .....	
Delay for Crystal and PLL Clock Sources .....	68
Device Resets .....	68

## H

High Temperature Electrical Characteristics .....	287
---	-----

## I

I/O Ports .....	141
Parallel I/O (PIO) .....	141
Write/Read Timing .....	142
I <sup>2</sup> C .....	
Operating Modes .....	165
Registers .....	167
I <sup>2</sup> C Module .....	
I2C1 Register Map .....	40
I2C2 Register Map .....	40
In-Circuit Debugger .....	228
In-Circuit Emulation .....	221
In-Circuit Serial Programming (ICSP) .....	221, 228
Input Capture .....	
Registers .....	154
Input Change Notification Module .....	142
Instruction Addressing Modes .....	53
File Register Instructions .....	53
Fundamental Modes Supported .....	54
MCU Instructions .....	53
Move and Accumulator Instructions .....	54
Other Instructions .....	54
Instruction Set .....	
Overview .....	231
Summary .....	229
Instruction-Based Power-Saving Modes .....	133
Idle .....	134
Sleep .....	133
Internal RC Oscillator .....	
Use with WDT .....	227
Internet Address .....	321
Interrupt Control and Status Registers .....	73
IECx .....	73
IFSx .....	73
INTCON1 .....	73
INTCON2 .....	73
INTTREG .....	73
IPCx .....	73
Interrupt Setup Procedures .....	111
Initialization .....	111
Interrupt Disable .....	111
Interrupt Service Routine .....	111
Trap Service Routine .....	111
Interrupt Vector Table (IVT) .....	69
Interrupts Coincident with Power Save Instructions .....	134

## J

JTAG Boundary Scan Interface .....	221
------------------------------------	-----

## M

Memory Organization .....	29
Microchip Internet Web Site .....	321
Modes of Operation .....	
Disable .....	181
Initialization .....	181
Listen All Messages .....	181
Listen Only .....	181
Loopback .....	181
Normal Operation .....	181
MPLAB ASM30 Assembler, Linker, Librarian .....	238
MPLAB Integrated Development .....	
Environment Software .....	237
MPLAB PM3 Device Programmer .....	240
MPLAB REAL ICE In-Circuit Emulator System .....	239
MPLINK Object Linker/MPLIB Object Librarian .....	238
Multi-Bit Data Shifter .....	28

## N

NVM Module .....	
Register Map .....	52

## O

Open-Drain Configuration .....	142
Output Compare .....	155

## P

Packaging .....	301
Details .....	304
Marking .....	301
Peripheral Module Disable (PMD) .....	134
Pinout I/O Descriptions (table) .....	17
PMD Module .....	
Register Map .....	52
POR and Long Oscillator Start-up Times .....	68
PORTA .....	
Register Map .....	50
PORTB .....	
Register Map .....	50
PORTC .....	
Register Map .....	50
PORTD .....	
Register Map .....	50
PORTE .....	
Register Map .....	51
PORTF .....	
Register Map .....	51
PORTG .....	
Register Map .....	51
Power-Saving Features .....	133
Clock Frequency and Switching .....	133
Program Address Space .....	29
Construction .....	55
Data Access from Program Memory Using	
Program Space Visibility .....	58
Data Access from Program Memory	
Using Table Instructions .....	57
Data Access from, Address Generation .....	56
Memory Map .....	29
Table Read Instructions .....	
TBLRDH .....	57
TBLRDL .....	57
Visibility Operation .....	58
Program Memory .....	
Interrupt Vector .....	30
Organization .....	30
Reset Vector .....	30