

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 ² C, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	21
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 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN-S (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24hj64gp202t-i-mm

4.0 MEMORY ORGANIZATION

Note: This data sheet summarizes the features of the PIC24HJ32GP302/304, PIC24HJ64GPX02/X04 and PIC24HJ128GPX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 4. "Program Memory"** (DS70203) of the "*dsPIC33F/PIC24H Family Reference Manual*", which is available from the Microchip web site (www.microchip.com).

The PIC24HJ32GP302/304, PIC24HJ64GPX02/X04 and PIC24HJ128GPX02/X04 architecture features separate program and data memory spaces and buses. This architecture also allows the direct access of program memory from the data space during code execution.

4.1 Program Address Space

The program address memory space of the PIC24HJ32GP302/304, PIC24HJ64GPX02/X04 and PIC24HJ128GPX02/X04 devices is 4M instructions. The space is addressable by a 24-bit value derived either from the 23-bit Program Counter (PC) during program execution, or from table operation or data space remapping as described in **Section 4.6 "Interfacing Program and Data Memory Spaces"**.

User application access to the program memory space is restricted to the lower half of the address range (0x000000 to 0x7FFFFFFF). The exception is the use of TBLRD/TBLWT operations, which use TBLPAG<7> to permit access to the Configuration bits and Device ID sections of the configuration memory space.

The memory map for the PIC24HJ32GP302/304, PIC24HJ64GPX02/X04 and PIC24HJ128GPX02/X04 devices is shown in **Figure 4-1**.

FIGURE 4-1: PROGRAM MEMORY MAP FOR PIC24HJ32GP302/304, PIC24HJ64GPX02/X04 AND PIC24HJ128GPX02/X04 DEVICES

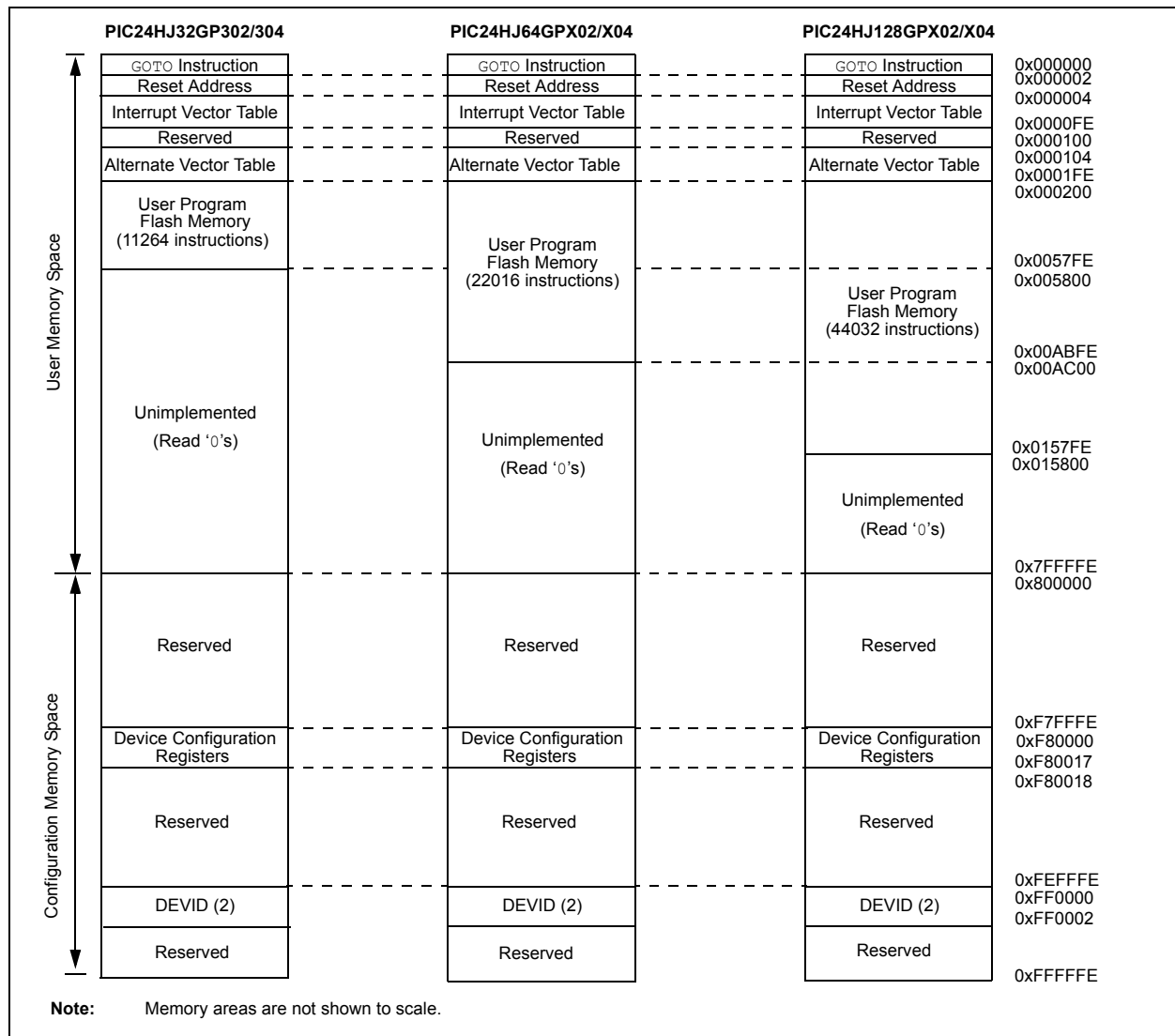


TABLE 4-4: INTERRUPT CONTROLLER REGISTER MAP

SFR Name	SFR Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets	
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000	
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000	
IFS0	0084	—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF	0000	
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF	IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000	
IFS2	0088	—	DMA4IF	PMPIF	—	—	—	—	—	—	—	—	DMA3IF	C1IF ⁽¹⁾	C1RXIF ⁽¹⁾	SPI2IF	SPI2EIF	0000	
IFS3	008A	—	RTCIF	DMA5IF	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
IFS4	008C	—	—	—	—	—	—	—	—	—	C1TXIF ⁽¹⁾	DMA7IF	DMA6IF	CRCIF	U2EIF	U1EIF	—	0000	
IEC0	0094	—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE	0000	
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE	IC8IE	IC7IE	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000	
IEC2	0098	—	DMA4IE	PMPIE	—	—	—	—	—	—	—	—	DMA3IE	C1IE ⁽¹⁾	C1RXIE ⁽¹⁾	SPI2IE	SPI2EIE	0000	
IEC3	009A	—	RTCIE	DMA5IE	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
IEC4	009C	—	—	—	—	—	—	—	—	—	C1TXIE ⁽¹⁾	DMA7IE	DMA6IE	CRCIE	U2EIE	U1EIE	—	0000	
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			4444	
IPC1	00A6	—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			—	DMA0IP<2:0>			4444	
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	T3IP<2:0>			4444	
IPC3	00AA	—	—	—	—	—	DMA1IP<2:0>			—	AD1IP<2:0>			—	U1TXIP<2:0>			0444	
IPC4	00AC	—	CNIP<2:0>			—	CMIP<2:0>			—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444	
IPC5	00AE	—	IC8IP<2:0>			—	IC7IP<2:0>			—	—	—	—	—	INT1IP<2:0>			4404	
IPC6	00B0	—	T4IP<2:0>			—	OC4IP<2:0>			—	OC3IP<2:0>			—	DMA2IP<2:0>			4444	
IPC7	00B2	—	U2TXIP<2:0>			—	U2RXIP<2:0>			—	INT2IP<2:0>			—	T5IP<2:0>			4444	
IPC8	00B4	—	C1IP<2:0> ⁽¹⁾			—	C1RXIP<2:0> ⁽¹⁾			—	SPI2IP<2:0>			—	SPI2EIP<2:0>			4444	
IPC9	00B6	—	—	—	—	—	—	—	—	—	—	—	—	—	DMA3IP<2:0>			0004	
IPC11	00BA	—	—	—	—	—	DMA4IP<2:0>			—	PMPIP<2:0>			—	—	—	—	0440	
IPC15	00C2	—	—	—	—	—	RTCIP<2:0>			—	DMA5IP<2:0>			—	—	—	—	0440	
IPC16	00C4	—	CRCIP<2:0>			—	U2EIP<2:0>			—	U1EIP<2:0>			—	—	—	—	4440	
IPC17	00C6	—	—	—	—	—	C1TXIP<2:0> ⁽¹⁾			—	DMA7IP<2:0>			—	DMA6IP<2:0>			0444	
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>								4444

Legend: × = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Interrupts disabled on devices without ECAN™ modules.

TABLE 4-7: OUTPUT COMPARE REGISTER MAP

SFR Name	SFR Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OC1RS	0180	Output Compare 1 Secondary Register																xxxx
OC1R	0182	Output Compare 1 Register																xxxx
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC2RS	0186	Output Compare 2 Secondary Register																xxxx
OC2R	0188	Output Compare 2 Register																xxxx
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC3RS	018C	Output Compare 3 Secondary Register																xxxx
OC3R	018E	Output Compare 3 Register																xxxx
OC3CON	0190	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC4RS	0192	Output Compare 4 Secondary Register																xxxx
OC4R	0194	Output Compare 4 Register																xxxx
OC4CON	0196	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-8: I2C1 REGISTER MAP

SFR Name	SFR Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
I2C1RCV	0200	—	—	—	—	—	—	—	—	Receive Register								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	Transmit Register								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	Baud Rate Generator Register									0000
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	Address Register										0000
I2C1MSK	020C	—	—	—	—	—	—	Address Mask Register										0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-9: UART1 REGISTER MAP

SFR Name	SFR Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMO	—	UEN1	UEN0	WAKE	LPBACK	ABAU	URXINV	BRGH	PDSEL<1:0>		STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	UTX8	UART Transmit Register								xxxx
U1RXREG	0226	—	—	—	—	—	—	—	URX8	UART Received Register								0000
U1BRG	0228	Baud Rate Generator Prescaler																0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-13: ADC1 REGISTER MAP FOR PIC24HJ64GP202/502, PIC24HJ128GP202/502 AND PIC24HJ32GP302

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	ADC Data Buffer 0																xxxx
AD1CON1	0320	ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000
AD1CON3	0324	ADRC	—	—	SAMC<4:0>				ADCS<7:0>									0000
AD1CHS123	0326	—	—	—	—	—	CH123NB<1:0>		CH123SB	—	—	—	—	—	CH123NA<1:0>		CH123SA	0000
AD1CHS0	0328	CH0NB	—	—	CH0SB<4:0>					CH0NA	—	—	CH0SA<4:0>					0000
AD1PCFGL	032C	—	—	—	PCFG12	PCFG11	PCFG10	PCFG9	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	—	—	—	CSS12	CSS11	CSS10	CSS9	—	—	—	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD1CON4	0332	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>			0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-14: ADC1 REGISTER MAP FOR PIC24HJ64GP204/504, PIC24HJ128GP204/504 AND PIC24HJ32GP304

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	ADC Data Buffer 0																xxxx
AD1CON1	0320	ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000
AD1CON3	0324	ADRC	—	—	SAMC<4:0>					ADCS<7:0>							0000	
AD1CHS123	0326	—	—	—	—	—	CH123NB<1:0>		CH123SB	—	—	—	—	—	CH123NA<1:0>		CH123SA	0000
AD1CHS0	0328	CH0NB	—	—	CH0SB<4:0>					CH0NA	—	—	CH0SA<4:0>					0000
AD1PCFGL	032C	—	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	—	—	—	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD1CON4	0332	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>			0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

5.6.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

Programmers 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 block (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) Write 0x55 to NVMKEY.
 - d) Write 0xAA to NVMKEY.
 - e) 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 application 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
```

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2

U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	DMA4IF	PMPIF	—	—	—	—	—
bit 15						bit 8	

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DMA3IF	C1IF ⁽¹⁾	C1RXIF ⁽¹⁾	SPI2IF	SPI2EIF
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 **DMA4IF:** DMA Channel 4 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 12-5 **Unimplemented:** Read as '0'
- bit 4 **DMA3IF:** DMA Channel 3 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 3 **C1IF:** ECAN1 Event Interrupt Flag Status bit⁽¹⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 2 **C1RXIF:** ECAN1 Receive Data Ready Interrupt Flag Status bit⁽¹⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 0 **SPI2EIF:** SPI2 Error Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred

Note 1: Interrupts disabled on devices without ECAN™ modules.

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

U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	DMA4IE	PMPIE	—	—	—	—	—
bit 15						bit 8	

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DMA3IE	C1IE ⁽¹⁾	C1RXIE ⁽¹⁾	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 **Unimplemented:** Read as '0'
- bit 14 **DMA4IE:** DMA Channel 4 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 13 **PMPIE:** Parallel Master Port Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 12-5 **Unimplemented:** Read as '0'
- bit 4 **DMA3IE:** DMA Channel 3 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request has enabled
- bit 3 **C1IE:** ECAN1 Event Interrupt Enable bit⁽¹⁾
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 2 **C1RXIE:** ECAN1 Receive Data Ready Interrupt Enable bit⁽¹⁾
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 1 **SPI2IE:** SPI2 Event Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 0 **SPI2EIE:** SPI2 Error Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled

Note 1: Interrupts disabled on devices without ECAN™ modules.

REGISTER 7-28: IPC17: INTERRUPT PRIORITY CONTROL REGISTER 17

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	C1TXIP<2:0> ⁽¹⁾		
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
—	DMA7IP<2:0>			—	DMA6IP<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 15-11 **Unimplemented:** Read as '0'

bit 10-8 **C1TXIP<2:0>:** ECAN1 Transmit Data Request Interrupt Priority bits⁽¹⁾

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 **DMA7IP<2:0>:** DMA Channel 7 Data Transfer 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 **DMA6IP<2:0>:** DMA Channel 6 Data Transfer Complete Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•
•
•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

Note 1: Interrupts disabled on devices without ECAN™ modules.

REGISTER 11-9: RPINR19: PERIPHERAL PIN SELECT INPUT REGISTER 19

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2CTSR<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2RXR<4:0>				
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 15-13 **Unimplemented:** Read as '0'

bit 12-8 **U2CTSR<4:0>:** Assign UART2 Clear to Send (U2CTS) to the corresponding RPn pin

11111 = Input tied to Vss

11001 = Input tied to RP25

•

•

•

00001 = Input tied to RP1

00000 = Input tied to RP0

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **U2RXR<4:0>:** Assign UART2 Receive (U2RX) to the corresponding RPn pin

11111 = Input tied to Vss

11001 = Input tied to RP25

•

•

•

00001 = Input tied to RP1

00000 = Input tied to RP0

The Timer2/3 and Timer4/5 modules can operate in one of the following modes:

- Timer mode
- Gated Timer mode
- Synchronous Counter mode

In Timer and Gated Timer modes, the input clock is derived from the internal instruction cycle clock (F_{cy}). In Synchronous Counter mode, the input clock is derived from the external clock input at TxCK pin.

The timer modes are determined by the following bits:

- TCS (TxCON<1>): Timer Clock Source Control bit
- TGATE (TxCON<6>): Timer Gate Control bit

Timer control bit settings for different operating modes are given in the [Table 13-1](#).

TABLE 13-1: TIMER MODE SETTINGS

Mode	TCS	TGATE
Timer	0	0
Gated timer	0	1
Synchronous counter	1	x

13.1 16-Bit Operation

To configure any of the timers for individual 16-bit operation:

1. Clear the T32 bit corresponding to that timer.
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Load the timer period value into the PRx register.
5. If interrupts are required, set the interrupt enable bit, TxIE. Use the priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON bit.

Note: Only Timer2 and Timer3 can trigger a DMA data transfer.

13.2 32-Bit Operation

A 32-bit timer module can be formed by combining a Type B and a Type C 16-bit timer module. For 32-bit timer operation, the T32 control bit in the Type B Timer Control register (TxCON<3>) must be set. The Type C timer holds the most significant word (msw) and the Type B timer holds the least significant word (lsw) for 32-bit operation.

When configured for 32-bit operation, only the Type B Timer Control register (TxCON) bits are required for setup and control. Type C timer control register bits are ignored (except TSIDL bit).

For interrupt control, the combined 32-bit timer uses the interrupt enable, interrupt flag and interrupt priority control bits of the Type C timer. The interrupt control and status bits for the Type B timer are ignored during 32-bit timer operation.

The Type B and Type C timers that can be combined to form a 32-bit timer are listed in [Table 13-2](#).

TABLE 13-2: 32-BIT TIMER

TYPE B Timer (lsw)	TYPE C Timer (msw)
Timer2	Timer3
Timer4	Timer5

A block diagram representation of the 32-bit timer module is shown in [Figure 13-3](#). The 32-timer module can operate in one of the following modes:

- Timer mode
- Gated Timer mode
- Synchronous Counter mode

To configure the features of Timer2/3 or Timer4/5 for 32-bit operation:

1. Set the T32 control bit.
2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the corresponding TCS and TGATE bits.
4. Load the timer period value. PR3 or PR5 contains the most significant word of the value, while PR2 or PR4 contains the least significant word.
5. If interrupts are required, set the interrupt enable bits, T3IE or T5IE. Use the priority bits, T3IP<2:0> or T5IP<2:0> to set the interrupt priority. While Timer2 or Timer4 controls the timer, the interrupt appears as a Timer3 or Timer5 interrupt.
6. Set the corresponding TON bit.

The timer value at any point is stored in the register pair, TMR3:TMR2 or TMR5:TMR4, which always contains the most significant word of the count, while TMR2 or TMR4 contains the least significant word.

18.1 UART Helpful Tips

1. In multi-node direct-connect UART networks, UART receive inputs react to the complementary logic level defined by the URXINV bit (UxMODE<4>), which defines the idle state, the default of which is logic high, (i.e., URXINV = 0). Because remote devices do not initialize at the same time, it is likely that one of the devices, because the RX line is floating, will trigger a start bit detection and will cause the first byte received after the device has been initialized to be invalid. To avoid this situation, the user should use a pull-up or pull-down resistor on the RX pin depending on the value of the URXINV bit.
 - a) If URXINV = 0, use a pull-up resistor on the RX pin.
 - b) If URXINV = 1, use a pull-down resistor on the RX pin.
2. The first character received on a wake-up from Sleep mode caused by activity on the UxRX pin of the UART module will be invalid. In Sleep mode, peripheral clocks are disabled. By the time the oscillator system has restarted and stabilized from Sleep mode, the baud rate bit sampling clock relative to the incoming UxRX bit timing is no longer synchronized, resulting in the first character being invalid. This is to be expected.

18.2 UART Resources

Many useful resources related to UART 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](http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en532315), 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=en532315
--

18.2.1 KEY RESOURCES

- **Section 17. “UART”** (DS70188)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All related dsPIC33F/PIC24H Family Reference Manuals Sections
- Development Tools

REGISTER 19-2: CICTRL2: ECAN™ CONTROL REGISTER 2

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	R-0	R-0	R-0	R-0	R-0
—	—	—	DNCNT<4:0>				
bit 7				bit 0			

Legend:	C = Writeable bit, but only '0' can be written to clear the 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 15-5 **Unimplemented:** Read as '0'

bit 4-0 **DNCNT<4:0>:** DeviceNet™ Filter Bit Number bits

10010-11111 = Invalid selection

10001 = Compare up to data byte 3, bit 6 with EID<17>

•

•

•

00001 = Compare up to data byte 1, bit 7 with EID<0>

00000 = Do not compare data bytes

REGISTER 19-4: CifCTRL: ECAN™ FIFO CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
DMABS<2:0>			—	—	—	—	—
bit 15							
			bit 8				

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	FSA<4:0>				
bit 7							
			bit 0				

Legend:	C = Writeable bit, but only '0' can be written to clear the 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 15-13 **DMABS<2:0>**: DMA Buffer Size bits

111 = Reserved
 110 = 32 buffers in DMA RAM
 101 = 24 buffers in DMA RAM
 100 = 16 buffers in DMA RAM
 011 = 12 buffers in DMA RAM
 010 = 8 buffers in DMA RAM
 001 = 6 buffers in DMA RAM
 000 = 4 buffers in DMA RAM

bit 12-5 **Unimplemented**: Read as '0'

bit 4-0 **FSA<4:0>**: FIFO Area Starts with Buffer bits

11111 = Read buffer RB31
 11110 = Read buffer RB30
 •
 •
 •
 00001 = TX/RX buffer TRB1
 00000 = TX/RX buffer TRB0

REGISTER 19-24: CIRXOVF1: ECAN™ RECEIVE BUFFER OVERFLOW REGISTER 1

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF15	RXOVF14	RXOVF13	RXOVF12	RXOVF11	RXOVF10	RXOVF9	RXOVF8
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF7	RXOVF6	RXOVF5	RXOVF4	RXOVF3	RXOVF2	RXOVF1	RXOVF0
bit 7							bit 0

Legend: C = Writeable bit, but only '0' can be written to clear the 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 15-0 **RXOVF<15:0>:** Receive Buffer n Overflow bits
1 = Module attempted to write to a full buffer (set by module)
0 = No overflow condition

REGISTER 19-25: CIRXOVF2: ECAN™ RECEIVE BUFFER OVERFLOW REGISTER 2

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF31	RXOVF30	RXOVF29	RXOVF28	RXOVF27	RXOVF26	RXOVF25	RXOVF24
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF23	RXOVF22	RXOVF21	RXOVF20	RXOVF19	RXOVF18	RXOVF17	RXOVF16
bit 7							bit 0

Legend: C = Writeable bit, but only '0' can be written to clear the 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 15-0 **RXOVF<31:16>:** Receive Buffer n Overflow bits
1 = Module attempted to write to a full buffer (set by module)
0 = No overflow condition

22.2 RTCC Resources

Many useful resources related to RTCC 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](http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en534555), contains the latest updates and additional information.

<p>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=en534555</p>
--

22.2.1 KEY RESOURCES

- **Section 37. “Real-Time Clock and Calendar (RTCC)” (DS70301)**
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All related dsPIC33F/PIC24H Family Reference Manuals Sections
- Development Tools

22.3 RTCC Registers

REGISTER 22-1: RCFGAL: RTCC CALIBRATION AND CONFIGURATION REGISTER⁽¹⁾

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN ⁽²⁾	—	RTCWREN	RTCSYNC	HALFSEC ⁽³⁾	RTCOE	RTCPTR<1:0>	
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
CAL<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 **RTCEN:** RTCC Enable bit⁽²⁾
 1 = RTCC module is enabled
 0 = RTCC module is disabled
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **RTCWREN:** RTCC Value Registers Write Enable bit
 1 = RTCVALH and RTCVALL registers can be written to by the user
 0 = RTCVALH and RTCVALL registers are locked out from being written to by the user
- bit 12 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
 1 = RTCVALH, RTCVALL and ALCFGRPT registers can change while reading due to a rollover ripple resulting in an invalid data read. If the register is read twice and results in the same data, the data can be assumed to be valid
 0 = RTCVALH, RTCVALL or ALCFGRPT registers can be read without concern over a rollover ripple
- bit 11 **HALFSEC:** Half-Second Status bit⁽³⁾
 1 = Second half period of a second
 0 = First half period of a second
- bit 10 **RTCOE:** RTCC Output Enable bit
 1 = RTCC output enabled
 0 = RTCC output disabled
- bit 9-8 **RTCPTR<1:0>:** RTCC Value Register Window Pointer bits
 Points to the corresponding RTCC Value registers when reading RTCVALH and RTCVALL registers; the RTCPTR<1:0> value decrements on every read or write of RTCVALH until it reaches '00'.
RTCVAL<15:8>:
 11 = Reserved
 10 = MONTH
 01 = WEEKDAY
 00 = MINUTES
RTCVAL<7:0>:
 11 = YEAR
 10 = DAY
 01 = HOURS
 00 = SECONDS

Note 1: The RCFGAL register is only affected by a POR.

Note 2: A write to the RTCEN bit is only allowed when RTCWREN = 1.

Note 3: This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

TABLE 25-3: CODE FLASH SECURITY SEGMENT SIZES FOR 32 KB DEVICES

CONFIG BITS	BSS<2:0> = x11 0K		BSS<2:0> = x10 1K		BSS<2:0> = x01 4K		BSS<2:0> = x00 8K	
SSS<2:0> = x11 0K	VS = 256 IW	0x000000h 0x0001FEh 0x000200h 0x0007FEh 0x000800h 0x001FFEh 0x002000h 0x003FFEh 0x004000h 0x0057FEh	VS = 256 IW	0x000000h 0x0001FEh 0x000200h 0x0007FEh 0x000800h 0x001FFEh 0x002000h 0x003FFEh 0x004000h 0x0057FEh	VS = 256 IW	0x000000h 0x0001FEh 0x000200h 0x0007FEh 0x000800h 0x001FFEh 0x002000h 0x003FFEh 0x004000h 0x0057FEh	VS = 256 IW	0x000000h 0x0001FEh 0x000200h 0x0007FEh 0x000800h 0x001FFEh 0x002000h 0x003FFEh 0x004000h 0x0057FEh
			BS = 768 IW		BS = 3840 IW		BS = 7936 IW	
	GS = 11008 IW		GS = 10240 IW		GS = 7168 IW		GS = 3072 IW	
		0x0157FEh		0x0157FEh		0x0157FEh		0x0157FEh

TABLE 26-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
12	BTST	BTST <i>f</i> , #bit4	Bit Test <i>f</i>	1	1	Z
		BTST.C <i>Ws</i> , #bit4	Bit Test <i>Ws</i> to C	1	1	C
		BTST.Z <i>Ws</i> , #bit4	Bit Test <i>Ws</i> to Z	1	1	Z
		BTST.C <i>Ws</i> , <i>Wb</i>	Bit Test <i>Ws</i> < <i>Wb</i> > to C	1	1	C
		BTST.Z <i>Ws</i> , <i>Wb</i>	Bit Test <i>Ws</i> < <i>Wb</i> > to Z	1	1	Z
13	BTSTS	BTSTS <i>f</i> , #bit4	Bit Test then Set <i>f</i>	1	1	Z
		BTSTS.C <i>Ws</i> , #bit4	Bit Test <i>Ws</i> to C, then Set	1	1	C
		BTSTS.Z <i>Ws</i> , #bit4	Bit Test <i>Ws</i> to Z, then Set	1	1	Z
14	CALL	CALL <i>lit</i> 23	Call subroutine	2	2	None
		CALL <i>Wn</i>	Call indirect subroutine	1	2	None
15	CLR	CLR <i>f</i>	<i>f</i> = 0x0000	1	1	None
		CLR WREG	WREG = 0x0000	1	1	None
		CLR <i>Ws</i>	<i>Ws</i> = 0x0000	1	1	None
16	CLRWDI	CLRWDI	Clear Watchdog Timer	1	1	WDTO, Sleep
17	COM	COM <i>f</i>	<i>f</i> = \bar{f}	1	1	N, Z
		COM <i>f</i> , WREG	WREG = \bar{f}	1	1	N, Z
		COM <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = \overline{Ws}	1	1	N, Z
18	CP	CP <i>f</i>	Compare <i>f</i> with WREG	1	1	C, DC, N, OV, Z
		CP <i>Wb</i> , #lit5	Compare <i>Wb</i> with lit5	1	1	C, DC, N, OV, Z
		CP <i>Wb</i> , <i>Ws</i>	Compare <i>Wb</i> with <i>Ws</i> (<i>Wb</i> – <i>Ws</i>)	1	1	C, DC, N, OV, Z
19	CP0	CP0 <i>f</i>	Compare <i>f</i> with 0x0000	1	1	C, DC, N, OV, Z
		CP0 <i>Ws</i>	Compare <i>Ws</i> with 0x0000	1	1	C, DC, N, OV, Z
20	CPB	CPB <i>f</i>	Compare <i>f</i> with WREG, with Borrow	1	1	C, DC, N, OV, Z
		CPB <i>Wb</i> , #lit5	Compare <i>Wb</i> with lit5, with Borrow	1	1	C, DC, N, OV, Z
		CPB <i>Wb</i> , <i>Ws</i>	Compare <i>Wb</i> with <i>Ws</i> , with Borrow (<i>Wb</i> – <i>Ws</i> – C)	1	1	C, DC, N, OV, Z
21	CPSEQ	CPSEQ <i>Wb</i> , <i>Wn</i>	Compare <i>Wb</i> with <i>Wn</i> , skip if =	1	1 (2 or 3)	None
22	CPSGT	CPSGT <i>Wb</i> , <i>Wn</i>	Compare <i>Wb</i> with <i>Wn</i> , skip if >	1	1 (2 or 3)	None
23	CPSLT	CPSLT <i>Wb</i> , <i>Wn</i>	Compare <i>Wb</i> with <i>Wn</i> , skip if <	1	1 (2 or 3)	None
24	CPSNE	CPSNE <i>Wb</i> , <i>Wn</i>	Compare <i>Wb</i> with <i>Wn</i> , skip if ≠	1	1 (2 or 3)	None
25	DAW	DAW <i>Wn</i>	<i>Wn</i> = decimal adjust <i>Wn</i>	1	1	C
26	DEC	DEC <i>f</i>	<i>f</i> = <i>f</i> – 1	1	1	C, DC, N, OV, Z
		DEC <i>f</i> , WREG	WREG = <i>f</i> – 1	1	1	C, DC, N, OV, Z
		DEC <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Ws</i> – 1	1	1	C, DC, N, OV, Z
27	DEC2	DEC2 <i>f</i>	<i>f</i> = <i>f</i> – 2	1	1	C, DC, N, OV, Z
		DEC2 <i>f</i> , WREG	WREG = <i>f</i> – 2	1	1	C, DC, N, OV, Z
		DEC2 <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Ws</i> – 2	1	1	C, DC, N, OV, Z
28	DISI	DISI #lit14	Disable Interrupts for <i>k</i> instruction cycles	1	1	None
29	DIV	DIV.S <i>Wm</i> , <i>Wn</i>	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
		DIV.SD <i>Wm</i> , <i>Wn</i>	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
		DIV.U <i>Wm</i> , <i>Wn</i>	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
		DIV.UD <i>Wm</i> , <i>Wn</i>	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
30	EXCH	EXCH <i>Wns</i> , <i>Wnd</i>	Swap <i>Wns</i> with <i>Wnd</i>	1	1	None
31	FBCL	FBCL <i>Ws</i> , <i>Wnd</i>	Find Bit Change from Left (MSb) Side	1	1	C
32	FF1L	FF1L <i>Ws</i> , <i>Wnd</i>	Find First One from Left (MSb) Side	1	1	C
33	FF1R	FF1R <i>Ws</i> , <i>Wnd</i>	Find First One from Right (LSb) Side	1	1	C
34	GOTO	GOTO <i>Expr</i>	Go to address	2	2	None
		GOTO <i>Wn</i>	Go to indirect	1	2	None

FIGURE 28-17: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)

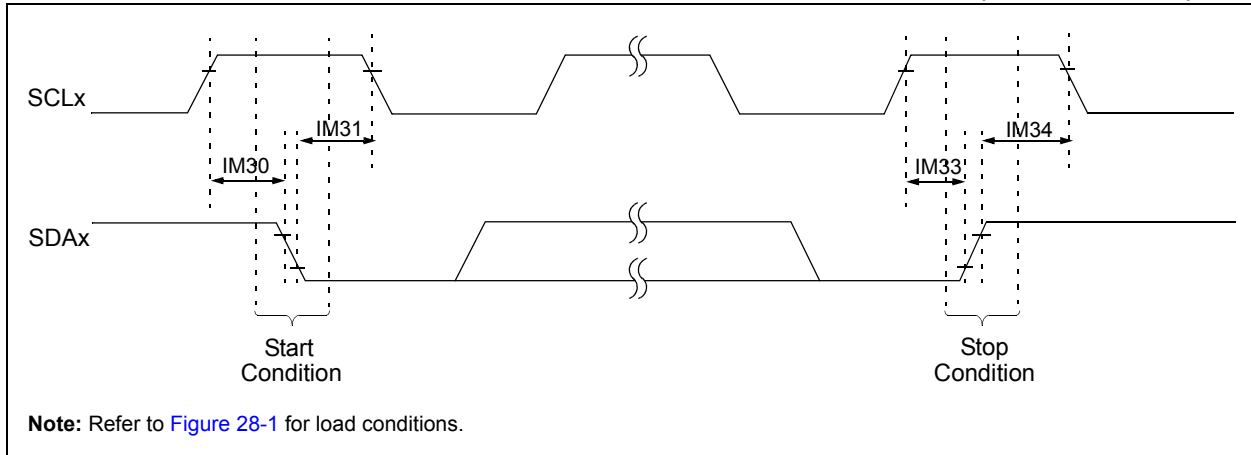
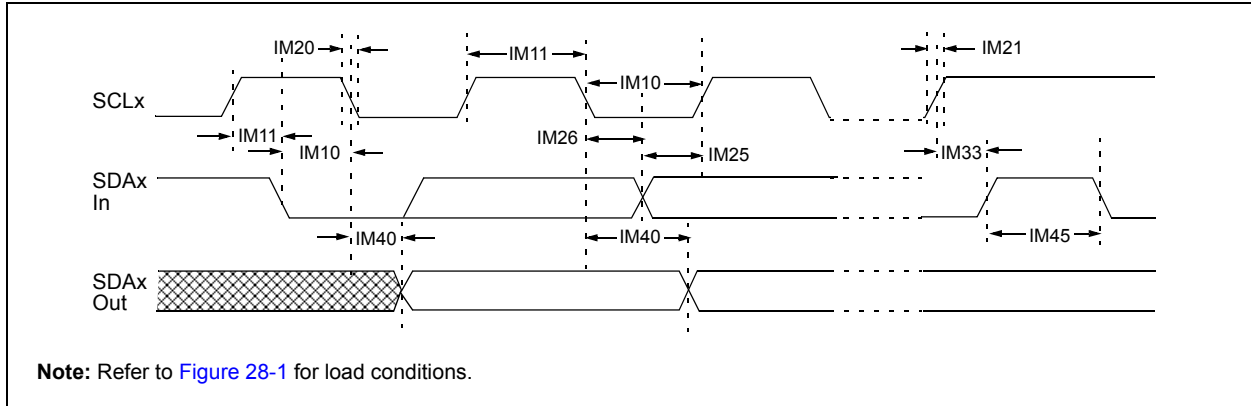


FIGURE 28-18: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)



32.0 DC AND AC DEVICE CHARACTERISTICS GRAPHS

Note: The graphs provided following this note are a statistical summary based on a limited number of samples and are provided for design guidance purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

FIGURE 32-1: V_{OH} – 2x DRIVER PINS

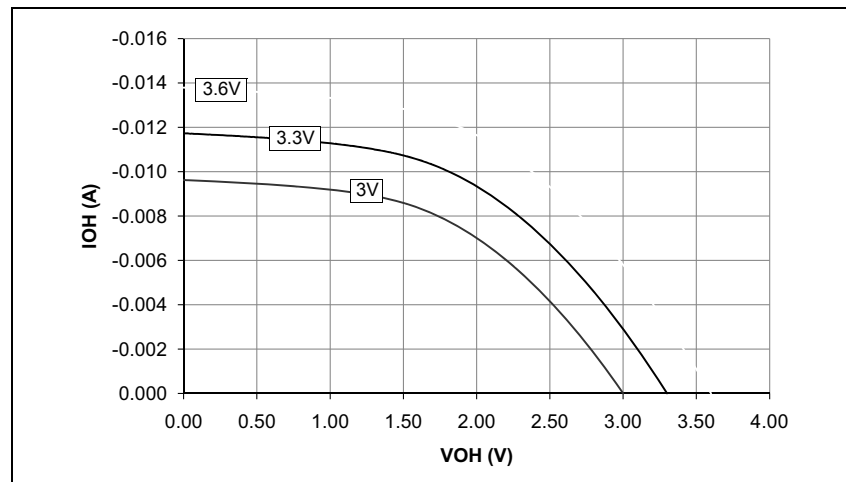


FIGURE 32-3: V_{OH} – 8x DRIVER PINS

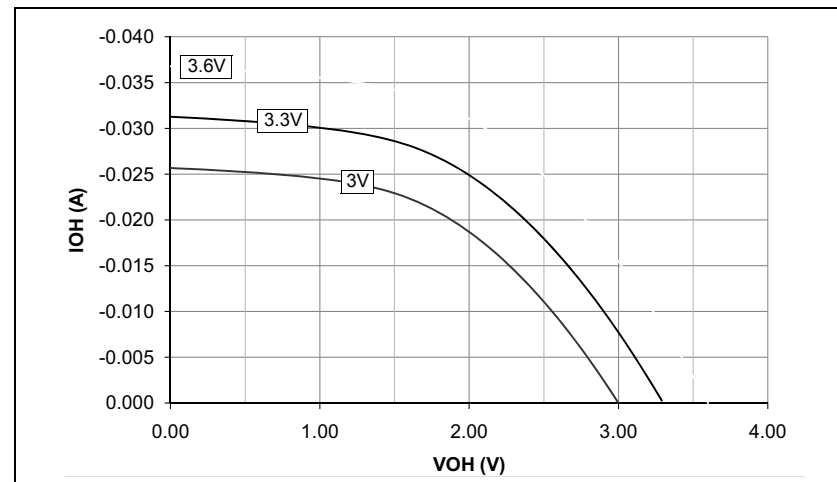


FIGURE 32-2: V_{OH} – 4x DRIVER PINS

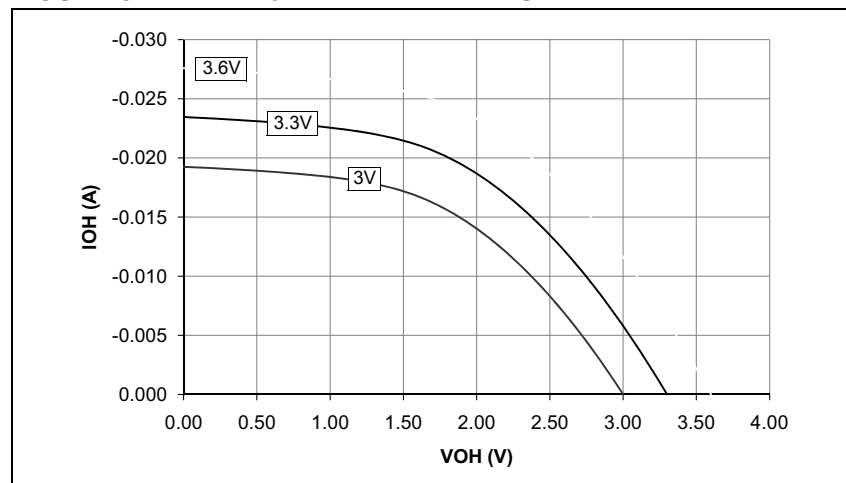


FIGURE 32-4: V_{OH} – 16x DRIVER PINS

