

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XE

Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	CANbus, I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	85
Program Memory Size	256КВ (256К х 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	30K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x10/12b
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	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj256mc710at-i-pt

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

4.6 Interfacing Program and Data Memory Spaces

The dsPIC33FJXXXMCX06A/X08A/X10A architecture uses a 24-bit wide program space and a 16-bit wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the dsPIC33FJXXXMCX06A/X08A/X10A architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space (Program Space Visibility)

Table instructions allow an application to read or write to small areas of the program memory. This capability makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look ups from a large table of static data. It can only access the least significant word of the program word.

4.6.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Page register (TBLPAG) is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full, 24-bit program space address. In this format, the Most Significant bit of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = 0) or the configuration memory (TBLPAG<7> = 1).

For remapping operations, the 8-bit Program Space Visibility register (PSVPAG) is used to define a 16K word page in the program space. When the Most Significant bit of the EA is '1', PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike table operations, this limits remapping operations strictly to the user memory area.

Table 4-38 and Figure 4-9 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

TABLE 4-38: PROGRAM SPACE ADDRESS CONSTRUCTION

	Access	Program Space Address					
Access Type	Space	<23>	<22:16>	<15>	<14:1>	<0>	
Instruction Access	User	0 PC<				0	
(Code Execution)		0xxx xxxx xxxx xxxx xxxx xxx0					
TBLRD/TBLWT (Byte/Word Read/Write)	User	TB	LPAG<7:0>	Data EA<15:0>			
		0	xxx xxxx	XXXX XX	xxxx xxxx xxxx xxxx		
	Configuration	TB	LPAG<7:0>	Data EA<15:0>			
		1	xxx xxxx	xxxx xxxx xxxx xxxx			
Program Space Visibility	User	0	0 PSVPAG<7		7:0> Data EA<14:0> ⁽¹⁾		
(Block Remap/Read)		0	XXXX XXXX		XXX XXXX XXXX XXXX		

Note 1: Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is PSVPAG<0>.

4.6.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into any 16K word page of the program space. This option provides transparent access of stored constant data from the data space without the need to use special instructions (i.e., TBLRDL/H).

Program space access through the data space occurs if the Most Significant bit of the data space EA is '1' and program space visibility is enabled by setting the PSV bit in the Core Control register (CORCON<2>). The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page register (PSVPAG). This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with the 15 bits of the EA functioning as the lower bits. Note that by incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads to this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-11), only the lower 16 bits of the 24-bit program word are used to contain the data. The upper 8 bits of any program space location used as data should be programmed with '1111 1111' or '0000 0000' to force a NOP. This prevents possible issues should the area of code ever be accidentally executed.

Note: PSV access is temporarily disabled during table reads/writes.

For operations that use PSV and are executed outside a REPEAT loop, the MOV and MOV.D instructions require one instruction cycle in addition to the specified execution time. All other instructions require two instruction cycles in addition to the specified execution time.

For operations that use PSV and are executed inside a REPEAT loop, there will be some instances that require two instruction cycles in addition to the specified execution time of the instruction:

- Execution in the first iteration
- · Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the REPEAT loop will allow the instruction accessing data using PSV to execute in a single cycle.

FIGURE 4-11: PROGRAM SPACE VISIBILITY OPERATION



© 2009-2012 Microchip Technology Inc.

NOTES:



NOTES:

16.0 MOTOR CONTROL PWM MODULE

- Note 1: This data sheet summarizes the features of the dsPIC33FJXXXMCX06A/ 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 Section 14. "Motor Control PWM" (DS70187) in the "dsPIC33F/PIC24H Family Reference Manual", which is available from the Microchip web site (www.microchip.com).
 - 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.

This module simplifies the task of generating multiple, synchronized Pulse-Width Modulated (PWM) outputs. In particular, the following power and motion control applications are supported by the PWM module:

- · 3-Phase AC Induction Motor
- Switched Reluctance (SR) Motor
- Brushless DC (BLDC) Motor
- Uninterruptible Power Supply (UPS)

The PWM module has the following features:

- Eight PWM I/O pins with four duty cycle generators
- · Up to 16-bit resolution
- 'On-the-fly' PWM frequency changes
- Edge and Center-Aligned Output modes
- Single Pulse Generation mode
- Interrupt support for asymmetrical updates in Center-Aligned mode
- Output override control for Electrically Commutative Motor (ECM) operation
- Special Event' comparator for scheduling other peripheral events
- Fault pins to optionally drive each of the PWM output pins to a defined state
- Duty cycle updates are configurable to be immediate or synchronized to the PWM time base

This module contains four duty cycle generators, numbered 1 through 4. The module has eight PWM output pins, numbered PWM1H/PWM1L through PWM4H/PWM4L. The eight I/O pins are grouped into high/low numbered pairs, denoted by the suffix H or L, respectively. For complementary loads, the low PWM pins are always the complement of the corresponding high I/O pin.

The PWM module allows several modes of operation which are beneficial for specific power control applications.

REGISTER 16-4: PxSECMP: PWMx SPECIAL EVENT COMPARE 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	
SEVTDIR ⁽¹⁾			ę	SEVTCMP<14:8	_{}>} (2)			
bit 15	<u>.</u>						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	
			SEVTC	MP<7:0>(2)				
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable	bit	U = Unimplemented bit, read as '0'				
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown	
bit 15	SEVTDIR: S	Special Event Tri	gger Time Ba	ase Direction bit	(1)			
	1 = A Specia	al Event Trigger	will occur wh	en the PWM tim	ne base is cou	nting downward	s	
	0 = A Specia	al Event Trigger	will occur wh	en the PWM tim	ne base is cou	nting upwards		
bit 14-0	SEVTCMP<	: 14:0>: Special E	Event Compa	re Value bits ⁽²⁾				

Note 1: SEVTDIR is compared with PTDIR (PTMR<15>) to generate the Special Event Trigger.

2: SEVTCMP<14:0> is compared with PTMR<14:0> to generate the Special Event Trigger.

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

bit 5	ACKDT: Acknowledge Data bit (when operating as I^2C master, applicable during master receive) Value that will be transmitted when the software initiates an Acknowledge sequence.
	0 = Send ACK during Acknowledge
bit 4	ACKEN: Acknowledge Sequence Enable bit (when operating as I ² 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 ² C master)
	 1 = Enables Receive mode for I²C. 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 ² 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 ² 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 ² 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

REGISTER 19-2: I2CxSTAT: I2Cx STATUS REGISTER

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC			
ACKSTAT	TRSTAT		_	—	BCL	GCSTAT	ADD10			
bit 15							bit 8			
R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC			
IWCOL	I2COV	D_A	Р	S	R_W	RBF	TBF			
bit 7							bit 0			
Legend:		U = Unimp	lemented bit,	, read as '0'		C = Clearable bit				
R = Readal	ole bit	W = Writat	ole bit	HS = Hardwa	are Settable bit	HSC = Hardware Se	ettable/Clearable bit			
-n = Value a	at POR	'1' = Bit is	set	'0' = Bit is cl	eared	x = Bit is unknown				
bit 15	bit 15 ACKSTAT: Acknowledge Status bit (when operating as I ² C [™] master, applicable to master transmit operation) 1 = NACK received from slave 0 = ACK received from slave Hardware set or clear at end of slave Acknowledge									
bit 14	 TRSTAT: Transmit Status bit (when operating as I²C master, applicable to master transmit operation) 1 = Master transmit is in progress (8 bits + ACK) 0 = Master transmit is not in progress 									
bit 13-11	Unimplem	ented: Rea	id as '0'				·			
bit 10	BCL: Mast	ter Bus Colli	ision Detect k	oit						
	1 = A bus 0 = No coll Hardware	collision has lision set at detec	s been detect tion of bus co	ed during a n	naster operation					
bit 9	GCSTAT: (General Cal	l Status bit							
	1 = Genera 0 = Genera Hardware	al call addre al call addre set when ac	ess was recei ess was not re Idress match	ved eceived es general ca	Ill address. Hard	ware clear at Stop de	etection.			
bit 8	ADD10: 10	D-Bit Addres	s Status bit							
	1 = 10-bit a 0 = 10-bit a Hardware	address was address was set at match	s matched s not matched n of 2nd byte	d of matched 1	0-bit address. H	ardware clear at Stop	o detection.			
bit 7	IWCOL: W	/rite Collisio	n Detect bit							
	1 = An atte 0 = No col Hardware	empt to write lision set at occur	e the I2CxTR	N register fail e to I2CxTRN	ed because the while busy (clea	I ² C module is busy ared by software).				
bit 6	I2COV: Re	ceive Overf	low Flag bit							
	1 = A byte 0 = No ove Hardware	was receive erflow set at attem	ed while the l	2CxRCV regi I2CxRSR to	ster is still holdir I2CxRCV (cleare	ng the previous byte ed by software).				
bit 5	D_A: Data	/Address bit	t (when opera	ating as I ² C s	lave)	- ,				
	1 = Indicat 0 = Indicat Hardware	es that the l es that the l clear at dev	ast byte rece ast byte rece ice address r	ived was data ived was a de natch. Hardw	a evice address are set by recep	tion of slave byte.				

REGISTER 21-2: 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:							
R = Readable bit W = Writable bit		bit	U = Unimpler	mented bit, read	as '0'		
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 15-5	Unimplement	ted: Read as 'd	כי				
bit 4-0	DNCNT<4:0>	: DeviceNet™	Filter Bit Num	ber bits			
	10010 - 1111	1 = Invalid sele	ection a byte 3 bit 6	with FID<17>			
	•						
	•						
	•						
	00001 = Com 00000 = Do n	pare up to data ot compare da	a byte 1, bit 7 ta bytes	with EID<0>			

REGISTER 21-11: CIFEN1: ECAN™ 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:	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 to Accept Messages bits

1 = Enable Filter n

0 = Disable Filter n

REGISTER 21-13: CiBUFPNT2: ECAN™ FILTER 4-7 BUFFER POINTER 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
	F7BF	P<3:0>			F6B	P<3:0>	
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
	F5BF	°<3:0>			F4B	P<3:0>	
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimpler	nented bit, rea	ad as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown
L							
bit 15-12	F7BP<3:0>: 1111 = Filte 1110 = Filte •	RX Buffer Writt r hits received ir r hits received ir	en when Filte n RX FIFO bu n RX Buffer 1	er 7 Hits bits iffer 4			
	•						
	0001 = Filte 0000 = Filte	r hits received ir r hits received ir	n RX Buffer 1 n RX Buffer 0				
bit 11-8	F6BP<3:0>: 1111 = Filte 1110 = Filte	RX Buffer Writt r hits received ir r hits received ir	en when Filte n RX FIFO bu n RX Buffer 1	er 6 Hits bits iffer 4			
	•						
	•						
	0001 = Filte 0000 = Filte	r hits received ir r hits received ir	n RX Buffer 1 n RX Buffer 0				
bit 7-4	F5BP<3:0>: 1111 = Filte 1110 = Filte	RX Buffer Writt r hits received ir r hits received ir	en when Filte n RX FIFO bu n RX Buffer 1	er 5 Hits bits iffer 4			
	•						
	0001 = Filte 0000 = Filte	r hits received ir r hits received ir	n RX Buffer 1 n RX Buffer 0				
bit 3-0	F4BP<3:0>: 1111 = Filte 1110 = Filte •	RX Buffer Writt r hits received ir r hits received ir	en when Filte n RX FIFO bu n RX Buffer 1	er 4 Hits bits Iffer 4			
	• 0001 = Filte 0000 = Filte	r hits received ir r hits received ir	n RX Buffer 1 n RX Buffer 0				

REGISTER 21-16: CIRXFnSID: ECAN™ ACCEPTANCE FILTER n STANDARD IDENTIFIER (n = 0, 1, ...,15)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
			SID<	:10:3>			
bit 15							bit 8
R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
	SID<2:0>		—	EXIDE	—	EID<1	17:16>
bit 7							bit 0
Legend:							
R = Readable bit W = Writable			bit	U = Unimpler	nented bit, read	as '0'	
-n = Value at POR '1' = Bit i		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 15-5	SID-10.0 S	tandard Identifi	er hite				
bit 15-5	1 = Message 0 = Message	address bit, SII address bit, SII	Dx, must be ' Dx, must be '	1' to match filte 0' to match filte	er er		
bit 4	Unimplemen	ted: Read as 'o)'				
bit 3	EXIDE: Exte	nded Identifier	Enable bit				
	<u>If MIDE = 1, t</u>	hen:					
	1 = Match onl 0 = Match onl	y messages wi y messages wi	th extended in the standard in	dentifier addres lentifier addres	sses ses		
	If MIDE = 0, t Ignore EXIDE	<u>hen:</u> bit.					
bit 2	Unimplemen	ted: Read as 'd)'				
bit 1-0	EID<17:16>:	Extended Iden	tifier bits				
	1 = Message 0 = Message	address bit, Ell address bit, Ell	Dx, must be ' Dx, must be '	1' to match filte 0' to match filte	er er		

REGISTER 21-17: CIRXFnEID: ECAN™ ACCEPTANCE FILTER n EXTENDED IDENTIFIER (n = 0, 1, ..., 15)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x			
	EID<15:8>									
bit 15	bit 15 bit 8									
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x			
			EID	<7:0>						
bit 7							bit 0			
Legend:										
R = Readable bit W = Writable bit			U = Unimplen	nented bit, read	l as '0'					
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown			

bit 15-0 EID<15:0>: Extended Identifier bits

1 = Message address bit, EIDx, must be '1' to match filter

0 = Message address bit, EIDx, must be '0' to match filter





REGISTER 22-2: ADxCON2: ADCx CONTROL REGISTER 2 (CONTINUED) (where x = 1 or 2)

- bit 0 ALTS: Alternate Input Sample Mode Select bit
 - 1 = Uses channel input selects for Sample A on first sample and Sample B on next sample
 - 0 = Always uses channel input selects for Sample A

TABLE 24-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
1	ADD	ADD	Acc	Add Accumulators	1	1	OA,OB,SA,SB
		ADD	f	f = f + WREG	1	1	C,DC,N,OV,Z
		ADD	f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		ADD	Wso,#Slit4,Acc	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
		BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
6	BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
		BRA	GE,Expr	Branch if greater than or equal	1	1 (2)	None
		BRA	GEU,Expr	Branch if unsigned greater than or equal	1	1 (2)	None
		BRA	GT,Expr	Branch if greater than	1	1 (2)	None
		BRA	GTU,Expr	Branch if unsigned greater than	1	1 (2)	None
		BRA	LE,Expr	Branch if less than or equal	1	1 (2)	None
		BRA	LEU,Expr	Branch if unsigned less than or equal	1	1 (2)	None
		BRA	LT,Expr	Branch if less than	1	1 (2)	None
		BRA	LTU,Expr	Branch if unsigned less than	1	1 (2)	None
		BRA	N,Expr	Branch if Negative	1	1 (2)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
		BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
		BRA	NOV, Expr	Branch if Not Overflow	1	1 (2)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
		BRA	OA,Expr	Branch if Accumulator A overflow	1	1 (2)	None
		BRA	OB,Expr	Branch if Accumulator B overflow	1	1 (2)	None
		BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
		BRA	SA,Expr	Branch if Accumulator A saturated	1	1 (2)	None
		BRA	SB,Expr	Branch if Accumulator B saturated	1	1 (2)	None
		BRA	Expr	Branch Unconditionally	1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
L		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	BSW	BSW.C	Ws,Wb	Write C bit to Ws <wb></wb>	1	1	None
ļ		BSW.Z	Ws,Wb	Write Z bit to Ws <wb></wb>	1	1	None
9	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
10	BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

25.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

25.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

25.4 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

25.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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

25.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C 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 IDE compatibility

DC CHARACTERISTICS			$\begin{tabular}{lllllllllllllllllllllllllllllllllll$					
Parameter No. ⁽³⁾	Typical ⁽²⁾	Мах	Units	Conditions				
Power-Down Current (IPD) ⁽¹⁾								
DC60d	50	200	μA	-40°C		Raso Power Down Current(3)		
DC60a	50	200	μA	+25°C	2 2\/			
DC60b	200	500	μA	+85°C	5.50	Base Fower-Down Current		
DC60c	600	1000	μA	+125°C				
DC61d	8	13	μA	-40°C		Watchdog Timer Current: ∆IwDT ⁽³⁾		
DC61a	10	15	μA	+25°C	2 21/			
DC61b	12	20	μA	+85°C	3.3V			
DC61c	13	25	μA	+125°C				

TABLE 26-7: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

Note 1: IPD (Sleep) current is measured as follows:

 CPU core is off, oscillator is configured in EC mode and external clock active, OSC1 is driven with external square wave from rail-to-rail (EC clock overshoot/undershoot < 250 mV required)

· CLKO is configured as an I/O input pin in the Configuration word

• All I/O pins are configured as inputs and pulled to Vss

• MCLR = VDD, WDT and FSCM are disabled, all peripheral modules except the ADC are disabled (PMDx bits are all '1's). The following ADC settings are enabled for each ADC module (ADCx) prior to executing the PWRSAV instruction: ADON = 1, VCFG = 1, AD12B = 1 and ADxMD = 0.

• VREGS bit (RCON<8>) = 0 (i.e., core regulator is set to stand-by while the device is in Sleep mode)

- RTCC is disabled.
- JTAG is disabled
- 2: Data in the "Typ" column is at 3.3V, +25°C unless otherwise stated.
- **3:** The Watchdog Timer Current is the additional current consumed when the WDT module is enabled. This current should be added to the base IPD current.
- 4: These currents are measured on the device containing the most memory in this family.
- 5: These parameters are characterized, but are not tested in manufacturing.

TABLE 26-38:SPIX SLAVE MODE (FULL-DUPLEX, CKE = 0, CKP = 1, SMP = 0) TIMING
REQUIREMENTS

AC CHARACTERISTICS		$\begin{array}{l} \mbox{Standard Operating Conditions: 2.4V to 3.6V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for Industrial} \\ & -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \end{array}$					
Param No.	Symbol	Characteristic ⁽¹⁾	Min	Тур ⁽²⁾	Max	Units	Conditions
SP70	TscP	Maximum SCK Input Frequency	—	_	15	MHz	See Note 3
SP72	TscF	SCKx Input Fall Time	—			ns	See parameter DO32 and Note 4
SP73	TscR	SCKx Input Rise Time	—			ns	See parameter DO31 and Note 4
SP30	TdoF	SDOx Data Output Fall Time	—			ns	See parameter DO32 and Note 4
SP31	TdoR	SDOx Data Output Rise Time	—			ns	See parameter DO31 and Note 4
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	SSx	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	SSx after SCKx Edge	1.5 TCY + 40	_	_	ns	See Note 4

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 specificiation.

4: Assumes 50 pF load on all SPIx pins.

FIGURE 26-26: CAN MODULE I/O TIMING CHARACTERISTICS

TABLE 26-42: ECAN™ TECHNOLOGY MODULE I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			$\begin{array}{l} \mbox{Standard Operating Conditions: 3.0V to 3.6V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for Industrial} \\ -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \end{array}$					
Param No.	Symbol	Characteristic ⁽¹⁾	Min	Тур	Max	Units	Conditions	
CA10	TioF	Port Output Fall Time				ns	See parameter D032	
CA11	TioR	Port Output Rise Time		—	—	ns	See parameter D031	
CA20	Tcwf	Pulse Width to Trigger CAN Wake-up Filter	120			ns	_	

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