

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·XEI

Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	CANbus, I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, DMA, I ² S, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 18x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj128gp706t-i-pt

Email: info@E-XFL.COM

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

Pin Diagrams (Continued)



Pin Name	Pin Type	Buffer Type	Description
AN0-AN31	I	Analog	Analog input channels.
AVDD	Р	Р	Positive supply for analog modules. This pin must be connected at all times.
AVss	Р	Р	Ground reference for analog modules.
CLKI CLKO	I O	ST/CMOS —	External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN23	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
COFS CSCK CSDI CSDO	I/O I/O I O	ST ST ST —	Data Converter Interface frame synchronization pin. Data Converter Interface serial clock input/output pin. Data Converter Interface serial data input pin. Data Converter Interface serial data output pin.
C1RX C1TX C2RX C2TX	 0 0	ST — ST —	ECAN1 bus receive pin. ECAN1 bus transmit pin. ECAN2 bus receive pin. ECAN2 bus transmit pin.
PGED1 PGEC1 PGED2 PGEC2 PGED3 PGEC3	I/O I I/O I I/O I	ST ST ST ST ST ST	Data I/O pin for programming/debugging communication channel 1. Clock input pin for programming/debugging communication channel 1. Data I/O pin for programming/debugging communication channel 2. Clock input pin for programming/debugging communication channel 2. Data I/O pin for programming/debugging communication channel 3. Clock input pin for programming/debugging communication channel 3.
IC1-IC8	I	ST	Capture inputs 1 through 8.
INT0 INT1 INT2 INT3 INT4		ST ST ST ST ST	External interrupt 0. External interrupt 1. External interrupt 2. External interrupt 3. External interrupt 4.
MCLR	I/P	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OCFA OCFB OC1-OC8	 0	ST ST —	Compare Fault A input (for Compare Channels 1, 2, 3 and 4). Compare Fault B input (for Compare Channels 5, 6, 7 and 8). Compare outputs 1 through 8.
OSC1 OSC2	I I/O	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.
RA0-RA7 RA9-RA10 RA12-RA15	I/O I/O I/O	ST ST ST	PORTA is a bidirectional I/O port.
RB0-RB15	I/O	ST	PORTB is a bidirectional I/O port.
RC1-RC4 RC12-RC15	I/O I/O	ST ST	PORTC is a bidirectional I/O port.
RD0-RD15	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE7	I/O	ST	PORTE is a bidirectional I/O port.
RF0-RF8 RF12-RF13	I/O I/O	ST ST	PORTF is a bidirectional I/O port.
Legend: CMC	S = CMO	S compatible	e input or output; Analog = Analog input; P = Power

ST = Schmitt Trigger input with CMOS levels;

O = Output; I = Input

TABLE 4-2: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJXXXGPX10 DEVICES

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
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0062	—	_	_	_	_	_	_	_	CN23IE	CN22IE	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006A	—	_	_	-			_	-	CN23PUE	CN22PUE	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000

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

TABLE 4-3: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJXXXGPX08 DEVICES

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
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0062	_	_	_	_	_	_	_	_	_	_	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006A	—	_	—	—	_	_	_	_	_	_	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000

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

TABLE 4-4: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJXXXGPX06 DEVICES

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
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0062	_	_	_	_	_	_	_	_	_	_	CN21IE	CN20IE	_	CN18IE	CN17IE	CN16IE	0000
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006A	_	_	_	_	_	_	_	_	_	_	CN21PUE	CN20PUE	_	CN18PUE	CN17PUE	CN16PUE	0000

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

TABLE 4-15: ADC1 REGISTER MAP

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	FOR	M<1:0>		SSRC<2:0>	>	_	SIMSAM	ASAM	SAMP	DONE	0000
AD1CON2	0322	,	VCFG<2:0	>	—	—	CSCNA	CHP	S<1:0>	BUFS	—		SMPI	<3:0>		BUFM	ALTS	0000
AD1CON3	0324	ADRC	—	—		S	AMC<4:0>						ADCS	6<7:0>				0000
AD1CHS123	0326	—	—	_	-	—	CH123N	VB<1:0>	CH123SB	—	-	_	—	—	CH123N	VA<1:0>	CH123SA	0000
AD1CHS0	0328	CH0NB	_	_		С	H0SB<4:0	>		CH0NA	_	_	CH0SA<4:0>				0000	
AD1PCFGH ⁽¹⁾	032A	PCFG31	PCFG30	PCFG29	PCFG28	PCFG27	PCFG26	PCFG25	PCFG24	PCFG23	PCFG22	PCFG21	PCFG20	PCFG19	PCFG18	PCFG17	PCFG16	0000
AD1PCFGL	032C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSH ⁽¹⁾	032E	CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24	CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16	0000
AD1CSSL	0330	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD1CON4	0332	_	_	_	_	_	_	_	_	_	_	_	_	_	I	DMABL<2:)>	0000

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

Note 1: Not all ANx inputs are available on all devices. See the device pin diagrams for available ANx inputs.

TABLE 4-16: ADC2 REGISTER MAP

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
ADC2BUF0	0340					ADC Data Buffer 0											xxxx	
AD2CON1	0360	ADON	_	ADSIDL	ADDMABM	_	AD12B	FOR	M<1:0>		SSRC<2:0	>	_	SIMSAM	ASAM	SAMP	DONE	0000
AD2CON2	0362		VCFG<2:0	>	_	_	CSCNA	CHP	S<1:0>	BUFS	_		SMPI	<3:0>		BUFM	ALTS	0000
AD2CON3	0364	ADRC	_	_		S	AMC<4:0>						ADC	S<7:0>				0000
AD2CHS123	0366	_	_	_	_	_	CH123N	VB<1:0>	CH123SB	_	_	_	_	_	CH1231	NA<1:0>	CH123SA	0000
AD2CHS0	0368	CH0NB	_	_	_		CH0S	B<3:0>		CH0NA	_	_	_	CH0SA<3:0>			0000	
Reserved	036A	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	0000
AD2PCFGL	036C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
Reserved	036E	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	0000
AD2CSSL	0370	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD2CON4	0372	_	_	_	_	_	_	_	_	_	_	_	_	_		DMABL<2	:0>	0000

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

4.2.7 SOFTWARE STACK

In addition to its use as a working register, the W15 register in the dsPIC33FJXXXGPX06/X08/X10 devices is also used as a software Stack Pointer. The Stack Pointer always points to the first available free word and grows from lower to higher addresses. It pre-decrements for stack pops and post-increments for stack pushes, as shown in Figure 4-6. For a PC push during any CALL instruction, the MSb of the PC is zero-extended before the push, ensuring that the MSb is always clear.

Note:	A PC push during exception processing
	concatenates the SRL register to the MSb
	of the PC prior to the push.

The Stack Pointer Limit register (SPLIM) associated with the Stack Pointer sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' because all stack operations must be word-aligned. Whenever an EA is generated using W15 as a source or destination pointer, the resulting address is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 0x2000 in RAM, initialize the SPLIM with the value 0x1FFE.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0x0800. This prevents the stack from interfering with the Special Function Register (SFR) space.

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

FIGURE 4-6: CALL STACK FRAME



4.2.8 DATA RAM PROTECTION FEATURE

The dsPIC33F product family supports Data RAM protection features which enable segments of RAM to be protected when used in conjunction with Boot and Secure Code Segment Security. BSRAM (Secure RAM segment for BS) is accessible only from the Boot Segment Flash code when enabled. SSRAM (Secure RAM segment for RAM) is accessible only from the Secure Segment Flash code when enabled. See Table 4-1 for an overview of the BSRAM and SSRAM SFRs.

4.3 Instruction Addressing Modes

The addressing modes in Table 4-35 form the basis of the addressing modes optimized to support the specific features of individual instructions. The addressing modes provided in the MAC class of instructions are somewhat different from those in the other instruction types.

4.3.1 FILE REGISTER INSTRUCTIONS

Most file register instructions use a 13-bit address field (f) to directly address data present in the first 8192 bytes of data memory (Near Data Space). Most file register instructions employ a working register, W0, which is denoted as WREG in these instructions. The destination is typically either the same file register or WREG (with the exception of the MUL instruction), which writes the result to a register or register pair. The MOV instruction allows additional flexibility and can access the entire data space.

4.3.2 MCU INSTRUCTIONS

The 3-operand MCU instructions are of the form:

Operand 3 = Operand 1 < function> Operand 2

where Operand 1 is always a working register (i.e., the addressing mode can only be register direct) which is referred to as Wb. Operand 2 can be a W register, fetched from data memory, or a 5-bit literal. The result location can be either a W register or a data memory location. The following addressing modes are supported by MCU instructions:

- Register Direct
- · Register Indirect
- · Register Indirect Post-Modified
- · Register Indirect Pre-Modified
- 5-bit or 10-bit Literal

Note: Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

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, which 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



REGISTER 7	-1: SR: C	PU STATUS F	REGISTER	")							
R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R -0	R/W-0				
OA	OB	SA	SB	OAB	SAB	DA	DC				
bit 15							bit 8				
R/W-0 ⁽³⁾	R/W-0 ⁽³⁾	R/W-0 ⁽³⁾	R-0	R/W-0	R/W-0	R/W-0	R/W-0				
IPL2 ⁽²⁾	IPL1 ⁽²⁾	IPL0 ⁽²⁾	RA	N	OV	Z	С				
bit 7							bit 0				
Legend:											
C = Clear only	bit	R = Readable	bit	U = Unimplemented bit, read as '0'							

141

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

bit 7-5

IPL<2:0>: CPU Interrupt Priority Level Status bits⁽²⁾

- 111 = CPU Interrupt Priority Level is 7 (15), user interrupts disabled
- 110 = CPU Interrupt Priority Level is 6 (14)
- 101 = CPU Interrupt Priority Level is 5 (13)
- 100 = CPU Interrupt Priority Level is 4 (12)
- 011 = CPU Interrupt Priority Level is 3 (11)
- 010 = CPU Interrupt Priority Level is 2 (10)
- 001 = CPU Interrupt Priority Level is 1 (9)
- 000 = CPU Interrupt Priority Level is 0 (8)

Note 1: For complete register details, see Register 3-1: "SR: CPU STATUS REGISTER".

- 2: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.
- 3: The IPL<2:0> Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

REGISTER 7-2: CORCON: CORE CONTROL REGISTER⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT		DL<2:0>	
bit 15							bit 8
R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3 ⁽²⁾	PSV	RND	IF
bit 7					•		bit 0
Legend:		C = Clear onl	y bit				
R = Readable	bit	W = Writable	bit	-n = Value at	POR	'1' = Bit is set	

bit 3

0' = Bit is cleared

IPL3: CPU Interrupt Priority Level Status bit 3(2)

'x = Bit is unknown

1 = CPU interrupt priority level is greater than 7

0 = CPU interrupt priority level is 7 or less

Note 1: For complete register details, see Register 3-2: "CORCON: CORE CONTROL REGISTER".

2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

U = Unimplemented bit, read as '0'

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2 (CONTINUED)

bit 2	C1RXIF: ECAN1 Receive Data Ready Interrupt Flag Status bit
	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

REGISTER 7-9:	IFS4: INTERRUPT FLAG STATUS REGISTER 4
---------------	--

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	_	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
C2TXIF	C1TXIF	DMA7IF	DMA6IF		U2EIF	U1EIF	—
bit 7							bit 0

Legend:				
R = Readable	bit	W = Writable bit	U = Unimplemented bit,	read as '0'
-n = Value at I	POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
bit 15-8	Unimpleme	nted: Read as '0'		
bit 7	C2TXIF: EC	AN2 Transmit Data Requ	est Interrupt Flag Status bit	
	1 = Interrupt 0 = Interrupt	request has occurred request has not occurred	ł	
bit 6	C1TXIF: EC	AN1 Transmit Data Requ	est Interrupt Flag Status bit	
	1 = Interrupt 0 = Interrupt	request has occurred request has not occurred	t	
bit 5	DMA7IF: DM	IA Channel 7 Data Trans	fer Complete Interrupt Flag St	atus bit
	1 = Interrupt 0 = Interrupt	request has occurred request has not occurred	t	
bit 4	DMA6IF: DM	IA Channel 6 Data Trans	fer Complete Interrupt Flag St	atus bit
	1 = Interrupt	request has occurred		
	0 = Interrupt	request has not occurred	ł	
bit 3	Unimpleme	nted: Read as '0'		
bit 2	U2EIF: UAR	T2 Error Interrupt Flag St	atus bit	
	1 = Interrupt 0 = Interrupt	request has occurred request has not occurred	ł	
bit 1	U1EIF: UAR	T1 Error Interrupt Flag St	atus bit	
	1 = Interrupt	request has occurred		
	0 = Interrupt	request has not occurred	1	
bit 0	Unimpleme	nted: Read as '0'		

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0					
		T6IP<2:0>		_		DMA4IP<2:0>						
bit 15	·						bit 8					
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0					
_	—	—		—		OC8IP<2:0>						
oit 7							bit (
Legend:												
R = Readab	le bit	W = Writable	bit	U = Unimpler	mented bit, rea	id as '0'						
-n = Value a	t POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkn	own					
=												
dit 15	Unimpleme	nted: Read as 'o)'									
oit 14-12	T6IP<2:0>:	Timer6 Interrupt	Priority bits									
	111 = Interrupt is priority 7 (highest priority interrupt)											
	•											
	•											
	001 = Interru 000 = Interru	upt is priority 1 upt source is dis	abled									
bit 11	Unimpleme	nted: Read as 'o)'									
bit 10-8	DMA4IP<2:(111 = Intern • •	0>: DMA Channe upt is priority 7 (I	el 4 Data Trai nighest priorit	nsfer Complete y interrupt)	Interrupt Prior	rity bits						
		upt is priority 1	ablad									
hit 7 2		upt source is dis										
		. Output Compa) 		:4. / h-:4-							
dit 2-0	0C8IP<2:0> 111 = Interru	upt is priority 7 (I	righest priorit	y interrupt Prior	ity dits							
	• 001 = Interru 000 = Interru	upt is priority 1 upt source is dis	abled									

REGISTER	7-28: IPC1	3: INTERRUPT	PRIORITY	CONTROL	REGISTER 1	3	
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_		C2RXIP<2:0>				INT4IP<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
_		INT3IP<2:0>	1011 0			T9IP<2:0>	1010 0
bit 7							bit 0
Legend:	o hit	\// = \//ritable k	oit	II – I Inimple	mented hit rea	ad as '0'	
n = Value at		41' = Rit is set	510	$0^{\circ} = \text{Bit is observed}$	nence bit, rea	v – Ritic unkn	0,4/0
	FUK	I – DILIS SEL			eareu		OWII
bit 15	Unimpleme	nted: Read as 'o	ı'				
bit 14-12	C2RXIP<2.	N: ECAN2 Rece	, ivo Data Ro	adv Interrunt P	riority bits		
51(14-12	111 = Interr	unt is priority 7 (h	niahest nriori	ty interrunt)			
	•		lightest phon	ty interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1	ablad				
1.11.44		upt source is disa					
Dit 11	Unimpleme	ented: Read as "d)'				
bit 10-8	IN14IP<2:0	>: External Interr	upt 4 Priority	bits			
	111 = Interr	upt is priority 7 (r	nignest prior	ty interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				
bit 7	Unimpleme	ented: Read as 'c)'				
bit 6-4	INT3IP<2:0	External Interr	upt 3 Priority	/ bits			
	111 = Interr	upt is priority 7 (h	nighest priori	ty interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				
bit 3	Unimpleme	nted: Read as 'c)'				
bit 2-0	T9IP<2:0>:	Timer9 Interrupt	Priority bits				
	111 = Interr	upt is priority 7 (h	nighest priori	ty interrupt)			
	•						
	•						
	• 001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				

REGISTER 9-	3: PLLF	BD: PLL FEEI	DBACK DI	ISOR REGIS	STER		
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0 ⁽¹⁾
_		_					PLLDIV<8>
bit 15		-				·	bit 8
R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
			PLLC	IV<7:0>			
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit		bit	U = Unimplemented bit, read as '0'				
-n = Value at POR (1' = Bit is set			'0' = Bit is cleared x = Bit is unknown			known	

bit 15-9 Unimplemented: Read as '0'

bit 8-0 PLLDIV<8:0>: PLL Feedback Divisor bits (also denoted as 'M', PLL multiplier)

```
000000000 = 2

000000001 = 3

000000010 = 4

.

.

000110000 = 50 (default)

.

.

111111111 = 513
```

10.0 POWER-SAVING FEATURES

Note: This data sheet summarizes the features of the dsPIC33FJXXXGPX06/X08/X10 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 9.
 "Watchdog Timer and Power-Saving Modes" (DS70196) in the "dsPIC33F Family Reference Manual", which is available from the Microchip web site (www.microchip.com).

The dsPIC33FJXXXGPX06/X08/X10 devices provide the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. dsPIC33FJXXXGPX06/X08/X10 devices can manage power consumption in four different ways:

- Clock frequency
- Instruction-based Sleep and Idle modes
- Software-controlled Doze mode
- · Selective peripheral control in software

Combinations of these methods can be used to selectively tailor an application's power consumption while still maintaining critical application features, such as timing-sensitive communications.

10.1 Clock Frequency and Clock Switching

dsPIC33FJXXXGPX06/X08/X10 devices allow a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSC bits (OSCCON<10:8>). The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 9.0 "Oscillator Configuration"**.

10.2 Instruction-Based Power-Saving Modes

dsPIC33FJXXXGPX06/X08/X10 devices have two special power-saving modes that are entered through the execution of a special PWRSAV instruction. Sleep mode stops clock operation and halts all code execution. Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. The assembly syntax of the PWRSAV instruction is shown in Example 10-1.

Note: SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

10.2.1 SLEEP MODE

Sleep mode has these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption is reduced to a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock continues to run in Sleep mode if the WDT is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items such as the input change notification on the I/O ports, or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation is disabled in Sleep mode.

The device will wake-up from Sleep mode on any of these events:

- Any interrupt source that is individually enabled.
- Any form of device Reset.
- A WDT time-out.

On wake-up from Sleep, the processor restarts with the same clock source that was active when Sleep mode was entered.

EXAMPLE 10-1: PWRSAV INSTRUCTION SYNTAX

PWRSAV #SLEEP_MODE PWRSAV #IDLE MODE ; Put the device into SLEEP mode ; Put the device into IDLE mode

11.2 Open-Drain Configuration

In addition to the PORT, LAT and TRIS registers for data control, some port pins can also be individually configured for either digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs higher than VDD (e.g., 5V) on any desired digital only pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum VIH specification.

See **"Pin Diagrams"** for the available pins and their functionality.

11.3 Configuring Analog Port Pins

The use of the ADxPCFGH, ADxPCFGL and TRIS registers control the operation of the ADC port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) is converted.

Clearing any bit in the ADxPCFGH or ADxPCFGL register configures the corresponding bit to be an analog pin. This is also the Reset state of any I/O pin that has an analog (ANx) function associated with it.

Note:	In devices with	two ADC	modules	s, if the
	corresponding	PCFG	bit in	either
	AD1PCFGH(L)	and AE	D2PCFGł	H(L) is
	cleared, the pin i	s configu	red as an	analog
	input.			

When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level).

Pins configured as digital inputs will not convert an analog input. Analog levels on any pin that is defined as a digital input (including the ANx pins) can cause the input buffer to consume current that exceeds the device specifications.

Note:	The voltage on an analog input pin can be
	between -0.3V to (VDD + 0.3 V).

EXAMPLE 11-1: PORT WRITE/READ EXAMPLE

MOV 0xFF00, W0 MOV W0, TRISBB NOP btss PORTB, #13

; Configure PORTB<15:8> as inputs ; and PORTB<7:0> as outputs ; Delay 1 cycle ; Next Instruction

11.4 I/O Port Write/Read Timing

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a NOP.

11.5 Input Change Notification

The input change notification function of the I/O ports allows the dsPIC33FJXXXGPX06/X08/X10 devices to generate interrupt requests to the processor in response to a change-of-state on selected input pins. This feature is capable of detecting input change-of-states even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 24 external signals (CN0 through CN23) that can be selected (enabled) for generating an interrupt request on a change-of-state.

There are four control registers associated with the CN module. The CNEN1 and CNEN2 registers contain the CN interrupt enable (CNxIE) control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin also has a weak pull-up connected to it. The pull-ups act as a current source that is connected to the pin and eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain the weak pull-up enable (CNxPUE) bits for each of the CN pins. Setting any of the control bits enables the weak pull-ups for the corresponding pins.

Note: Pull-ups on change notification pins should always be disabled whenever the port pin is configured as a digital output.

13.0 TIMER2/3, TIMER4/5, TIMER6/7 AND TIMER8/9

Note: This data sheet summarizes the features of the dsPIC33FJXXXGPX06/X08/X10 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 11.** "**Timers**" (DS70205) in the "dsPIC33F Family Reference Manual", which is available from the Microchip web site (www.microchip.com).

The Timer2/3, Timer4/5, Timer6/7 and Timer8/9 modules are 32-bit timers, which can also be configured as four independent 16-bit timers with selectable operating modes.

As a 32-bit timer, Timer2/3, Timer4/5, Timer6/7 and Timer8/9 operate in three modes:

- Two Independent 16-bit Timers (e.g., Timer2 and Timer3) with all 16-bit operating modes (except Asynchronous Counter mode)
- Single 32-bit Timer
- Single 32-bit Synchronous Counter
- They also support these features:
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during Idle and Sleep modes
- Interrupt on a 32-bit Period Register Match
- Time Base for Input Capture and Output Compare Modules (Timer2 and Timer3 only)
- ADC1 Event Trigger (Timer2/3 only)
- ADC2 Event Trigger (Timer4/5 only)

Individually, all eight of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the event trigger; this is implemented only with Timer2/3. The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON, T5CON, T6CON, T7CON, T8CON and T9CON registers. T2CON, T4CON, T6CON and T8CON are shown in generic form in Register 13-1. T3CON, T5CON, T7CON and T9CON are shown in Register 13-2.

For 32-bit timer/counter operation, Timer2, Timer4, Timer6 or Timer8 is the least significant word; Timer3, Timer5, Timer7 or Timer9 is the most significant word of the 32-bit timers.

Note: For 32-bit operation, T3CON, T5CON, T7CON and T9CON control bits are ignored. Only T2CON, T4CON, T6CON and T8CON control bits are used for setup and control. Timer2, Timer4, Timer6 and Timer8 clock and gate inputs are utilized for the 32-bit timer modules, but an interrupt is generated with the Timer3, Timer5, Ttimer7 and Timer9 interrupt flags.

To configure Timer2/3, Timer4/5, Timer6/7 or Timer8/9 for 32-bit operation:

- 1. Set the corresponding T32 control bit.
- 2. Select the prescaler ratio for Timer2, Timer4, Timer6 or Timer8 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, PR5, PR7 or PR9 contains the most significant word of the value, while PR2, PR4, PR6 or PR8 contains the least significant word.
- If interrupts are required, set the interrupt enable bit, T3IE, T5IE, T7IE or T9IE. Use the priority bits, T3IP<2:0>, T5IP<2:0>, T7IP<2:0> or T9IP<2:0>, to set the interrupt priority. While Timer2, Timer4, Timer6 or Timer8 control the timer, the interrupt appears as a Timer3, Timer5, Timer7 or Timer9 interrupt.
- 6. Set the corresponding TON bit.

The timer value at any point is stored in the register pair, TMR3:TMR2, TMR5:TMR4, TMR7:TMR6 or TMR9:TMR8. TMR3, TMR5, TMR7 or TMR9 always contains the most significant word of the count, while TMR2, TMR4, TMR6 or TMR8 contains the least significant word.

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.

A block diagram for a 32-bit timer pair (Timer4/5) example is shown in Figure 13-1 and a timer (Timer4) operating in 16-bit mode example is shown in Figure 13-2.

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

REGISTER	R 13-1: TxCO	N (T2CON, T4	4CON, T6C	ON OR T8CC	ON) CONTRO	L REGISTER						
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0					
TON		TSIDL			_	—	—					
bit 15							bit 8					
U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0					
	TGATE	TCKPS	6<1:0>	T32	—	TCS ⁽¹⁾	—					
bit 7							bit 0					
Legend:												
R = Readab	ole bit	W = Writable	bit	U = Unimple	mented bit, rea	ad as '0'						
-n = Value a	at POR	'1' = Bit is set		'0' = Bit is cle	eared	x = Bit is unkn	iown					
bit 15	TON: Timerx	On bit										
	When T32 =	<u>1:</u>										
	1 = Starts 32	-bit Timerx/y										
	0 = Stops 32	-bit Timerx/y										
	$\frac{\text{vmen } 132 =}{1 = \text{Starts } 16}$	<u>0:</u> -bit Timerx										
	0 = Stops 16	-bit Timerx										
bit 14	Unimplemen	ted: Read as '	0'									
bit 13	TSIDL: Stop	in Idle Mode bit	:									
	1 = Discontin 0 = Continue	ue module ope module operat	ration when o ion in Idle mo	device enters lo ode	dle mode							
bit 12-7	Unimplemer	ted: Read as '	0'									
bit 6	TGATE: Time	erx Gated Time	Accumulatio	n Enable bit								
	When TCS =	When TCS = 1:										
	This bit is ign	This bit is ignored.										
	<u>When TCS =</u> 1 = Catod tin	<u>When TCS = 0:</u> 1 = Gated time accumulation enabled										
	0 = Gated tin	ne accumulation	n disabled									
bit 5-4	TCKPS<1:0>	: Timerx Input	Clock Presca	ale Select bits								
	11 = 1:256	•										
	10 = 1:64											
	01 = 1:8											
hit 0	00 = 1.1	imor Modo Colo	ot hit									
DIL S	1 = Timery a	nd Timery form	a single 32-h	vit timer								
	0 = Timerx a	nd Timery act a	s two 16-bit t	imers								
bit 2	Unimplemer	ted: Read as '	0'									
bit 1	TCS: Timerx	Clock Source S	Select bit ⁽¹⁾									
	1 = External 0 = Internal c	clock from pin ٦ lock (Fcy)	TxCK (on the	rising edge)								
bit 0	Unimplemer	ted: Read as '	0'									

Note 1: The TxCK pin is not available on all timers. Refer to the "Pin Diagrams" section for the available pins.

NOTES:

FIGURE 19-1: ECAN™ MODULE BLOCK DIAGRAM



REGISTER 19-8: CIEC: ECAN[™] TRANSMIT/RECEIVE ERROR COUNT REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	
			TERR	CNT<7:0>				
bit 15							bit 8	
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	
			RERR	CNT<7:0>				
bit 7							bit 0	
Legend:								
R = Readable b	it	W = Writable bit		U = Unimplemented bit, read as '0'				
-n = Value at PC	OR	'1' = Bit is set		'0' = Bit is cleared x = Bit is unknown				

bit 15-8**TERRCNT<7:0>:** Transmit Error Count bitsbit 7-0**RERRCNT<7:0>:** Receive Error Count bits

REGISTER 1	9-9: CiCFG	61: ECAN™ E	BAUD RATI	E CONFIGUR	ATION REGI	STER 1	
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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
SJW<1:0>		BRP<5:0>					
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit			bit	U = Unimplemented bit, read as '0'			
-n = Value at POR		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 15-8	Unimplemented: Read as '0'						
bit 7-6	SJW<1:0>: Synchronization Jump Width bits						
	11 = Length is 4 x TQ						
	$10 = \text{Length} \text{ is } 3 \times \text{Tq}$						
	$01 = \text{Length is } 2 \times 1 Q$						
bit 5-0	BRD<5:0>: Baud Rate Prescalar hits						
	11 1111 = T	$c_0 = 2 \times 64 \times 1/2$					
	•						
	•						
	00 0010 = T	Q = 2 x 3 x 1/F	CAN				
	00 0001 = TQ = 2 x 2 x 1/FCAN						

00 0000 = TQ = 2 x 1 x 1/FCAN