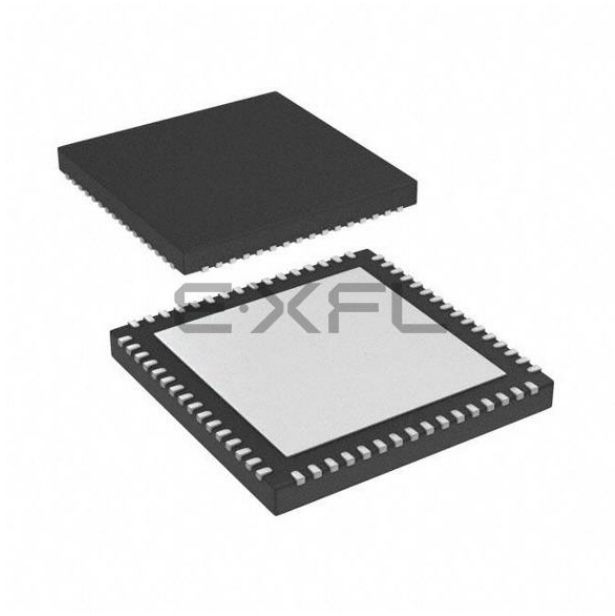
Microchip Technology - DSPIC33CH128MP506-I/MR Datasheet



Welcome to E-XFL.COM

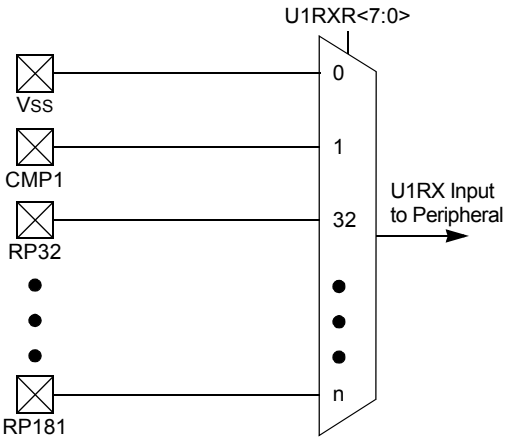
What is "Embedded - Microcontrollers"?

"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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit Dual-Core
Speed	180MHz, 200MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	53
Program Memory Size	152KB (152K x 8)
Program Memory Type	FLASH, PRAM
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 34x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch128mp506-i-mr

FIGURE 3-20: REMAPPABLE INPUT FOR U1RX



Note: For input only, Peripheral Pin Select functionality does not have priority over TRISx settings. Therefore, when configuring an RPn pin for input, the corresponding bit in the TRISx register must also be configured for input (set to '1'). Physical connection to a pin can be made through RP32 through RP71. There are internal signals and virtual pins that can be connected to an input. Table 3-30 shows the details of the input assignment.

Example 3-2 provides a configuration for bidirectional communication with flow control using UART1. The following input and output functions are used:

- Input Functions: U1RX, $\overline{\text{U1CTS}}$
- Output Functions: U1TX, $\overline{\text{U1RTS}}$

EXAMPLE 3-2: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS

```
//
*****************************************
// Unlock Registers
//*****************************************
__builtin_write_RPCON(0x0000);
//*****************************************
// Configure Input Functions (See Table 3-31)
// Assign U1Rx To Pin RP35
//***************************
_U1RXR = 35;
// Assign U1CTS To Pin RP36
//***************************
_U1CTSR = 36;
//*****************************************
// Configure Output Functions (See Table 3-33)
//*****************************************
// Assign U1Tx To Pin RP37
//***************************
_RP37 = 1;
//***************************
// Assign U1RTS To Pin RP38
//***************************
_RP38 = 2;
//*****************************************
// Lock Registers
//*****************************************
__builtin_write_RPCON(0x0800);
```

REGISTER 3-135: C1FIFOSTAx: CAN FIFO STATUS REGISTER x (x = 1 TO 7) (CONTINUED)

bit 2 **TFERFFIF:** Transmit/Receive FIFO Empty/Full Interrupt Flag bit
TXEN = 1 (FIFO configured as a transmit FIFO):
Transmit FIFO Empty Interrupt Flag
1 = FIFO is empty
0 = FIFO is not empty, at least one message is queued to be transmitted
TXEN = 0 (FIFO configured as a receive FIFO):
Receive FIFO Full Interrupt Flag
1 = FIFO is full
0 = FIFO is not full

bit 1 **TFHRFHIF:** Transmit/Receive FIFO Half Empty/Half Full Interrupt Flag bit
TXEN = 1 (FIFO configured as a transmit FIFO):
Transmit FIFO Half Empty Interrupt Flag
1 = FIFO is ≤ half full
0 = FIFO is > half full
TXEN = 0 (FIFO configured as a receive FIFO):
Receive FIFO Half Full Interrupt Flag
1 = FIFO is ≥ half full
0 = FIFO is < half full

bit 0 **TFNRFNIF:** Transmit/Receive FIFO Not Full/Not Empty Interrupt Flag bit
TXEN = 1 (FIFO configured as a transmit FIFO):
Transmit FIFO Not Full Interrupt Flag
1 = FIFO is not full
0 = FIFO is full
TXEN = 0 (FIFO configured as a receive FIFO):
Receive FIFO Not Empty Interrupt Flag
1 = FIFO is not empty, has at least one message
0 = FIFO is empty

Note 1: FIFOCI<4:0> gives a zero-indexed value to the message in the FIFO. If the FIFO is four messages deep (FSIZE<4:0> = 3), FIFOCIx will take on a value of 0 to 3, depending on the state of the FIFO.
2: These bits are updated when a message completes (or aborts) or when the FIFO is reset.
3: This bit is reset on any read of this register or when the TXQ is reset. The bits are cleared when TXREQ is set or using an SPI write.

REGISTER 3-145: C1TRECH: CAN TRANSMIT/RECEIVE ERROR COUNT REGISTER HIGH

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

U-0	U-0	R-1	R-0	R-0	R-0	R-0	R-0
—	—	TXBO	TXBP	RXBP	TXWARN	RXWARN	EWARN
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 15-6 **Unimplemented:** Read as '0'

bit 5 **TXBO:** Transmitter in Error State Bus Off bit (TERRCNT<7:0> > 255)
In Configuration mode, TXBO is set since the module is not on the bus.

bit 4 **TXBP:** Transmitter in Error State Bus Passive bit (TERRCNT<7:0> > 127)

bit 3 **RXBP:** Receiver in Error State Bus Passive bit (RERRCNT<7:0> > 127)

bit 2 **TXWARN:** Transmitter in Error State Warning bit (128 > TERRCNT<7:0> > 95)

bit 1 **RXWARN:** Receiver in Error State Warning bit (128 > RERRCNT<7:0> > 95)

bit 0 **EWARN:** Transmitter or Receiver in Error State Warning bit

REGISTER 3-146: C1TRECL: CAN TRANSMIT/RECEIVE ERROR COUNT REGISTER LOW

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TERRCNT<7:0>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RERRCNT<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 15-8 **TERRCNT<7:0>:** Transmit Error Counter bits

bit 7-0 **RERRCNT<7:0>:** Receive Error Counter bits

REGISTER 3-158: ADCON1H: ADC CONTROL REGISTER 1 HIGH

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-1	R/W-1	U-0	U-0	U-0	U-0	U-0
FORM	SHRRES1	SHRRES0	—	—	—	—	—
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 15-8 **Unimplemented:** Read as '0'

bit 7 **FORM:** Fractional Data Output Format bit
1 = Fractional
0 = Integer

bit 6-5 **SHRRES<1:0>:** Shared ADC Core Resolution Selection bits
11 = 12-bit resolution
10 = 10-bit resolution
01 = 8-bit resolution
00 = 6-bit resolution

bit 4-0 **Unimplemented:** Read as '0'

REGISTER 3-160: ADCON2H: ADC CONTROL REGISTER 2 HIGH

HSC/R-0	HSC/R-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
REFRDY	REFERR	—	—	—	—	SHRSAMC9	SHRSAMC8
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
SHRSAMC7	SHRSAMC6	SHRSAMC5	SHRSAMC4	SHRSAMC3	SHRSAMC2	SHRSAMC1	SHRSAMC0
bit 7							bit 0

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

bit 15 **REFRDY:** Band Gap and Reference Voltage Ready Flag bit
1 = Band gap is ready
0 = Band gap is not ready

bit 14 **REFERR:** Band Gap or Reference Voltage Error Flag bit
1 = Band gap was removed after the ADC module was enabled (ADON = 1)
0 = No band gap error was detected

bit 13-10 **Unimplemented:** Read as '0'

bit 9-0 **SHRSAMC<9:0>:** Shared ADC Core Sample Time Selection bits
These bits specify the number of shared ADC Core Clock Periods (TADCORE) for the shared ADC core sample time (Sample Time = (SHRSAMC<9:0> + 2) * TADCORE).
1111111111 = 1025 TADCORE
...
0000000001 = 3 TADCORE
0000000000 = 2 TADCORE

REGISTER 3-185: PTGBTE: PTG BROADCAST TRIGGER ENABLE LOW REGISTER(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE<7:0>							
bit 7							bit 0

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

bit 15-0 **PTGBTE<15:0>:** PTG Broadcast Trigger Enable bits
1 = Generates trigger when the broadcast command is executed
0 = Does not generate trigger when the broadcast command is executed

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 3-186: PTGBTEH: PTG BROADCAST TRIGGER ENABLE HIGH REGISTER(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE<31:24>							
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
PTGBTE<23:16>							
bit 7							bit 0

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

bit 15-0 **PTGBTE<31:16>:** PTG Broadcast Trigger Enable bits
1 = Generates trigger when the broadcast command is executed
0 = Does not generate trigger when the broadcast command is executed

Note 1: These bits are read-only when the module is executing Step commands.

4.2.2 DATA ADDRESS SPACE (SLAVE)

The dsPIC33CH128MP508S1 family CPU has a separate 16-bit wide data memory space. The Data Space is accessed using separate Address Generation Units (AGUs) for read and write operations. The data memory map is shown in Figure 4-5.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the Data Space. This arrangement gives a base Data Space address range of 64 Kbytes or 32K words.

The lower half of the data memory space (i.e., when EA<15> = 0) is used for implemented memory addresses, while the upper half (EA<15> = 1) is reserved for the Program Space Visibility (PSV).

The dsPIC33CH128MP508S1 family devices implement up to 4 Kbytes of data memory. If an EA points to a location outside of this area, an all-zero word or byte is returned.

4.2.2.1 Data Space Width

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all Data Space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

4.2.2.2 Data Memory Organization and Alignment

To maintain backward compatibility with PIC® MCU devices and improve Data Space memory usage efficiency, the dsPIC33CH128MP508S1 family instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] results in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

A data byte read, reads the complete word that contains the byte, using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register that matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap is generated. If the error occurred on a read, the instruction underway is completed. If the error occurred on a write, the instruction is executed but the write does not occur. In either case, a trap is then executed, allowing the system and/or user application to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the LSB; the MSB is not modified.

A Sign-Extend (SE) instruction is provided to allow user applications to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, user applications can clear the MSB of any W register by executing a Zero-Extend (ZE) instruction on the appropriate address.

4.2.2.3 SFR Space

The first 4 Kbytes of the Near Data Space, from 0x0000 to 0x0FFF, is primarily occupied by Special Function Registers (SFRs). These are used by the dsPIC33CH128MP508S1 family core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'.

Note:	The actual set of peripheral features and interrupts varies by the device. Refer to the corresponding device tables and pinout diagrams for device-specific information.

4.2.2.4 Near Data Space

The 8-Kbyte area, between 0x0000 and 0x1FFF, is referred to as the Near Data Space. Locations in this space are directly addressable through a 13-bit absolute address field within all memory direct instructions. Additionally, the whole Data Space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field, or by using Indirect Addressing mode using a Working register as an Address Pointer.

TABLE 4-9: SLAVE SFR BLOCK 900h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
CCP			CCP2STATL	980	-----0--00xx0000	CCP3RAL	9B0	0000000000000000
CCP1CON1L	950	0-00000000000000	CCP2STATH	982	----------00000	CCP3RBL	9B4	0000000000000000
CCP1CON1H	952	00--000000000000	CCP2TMRL	984	0000000000000000	CCP3BUFL	9B8	0000000000000000
CCP1CON2L	954	00-0----00000000	CCP2TMRH	986	0000000000000000	CCP3BUFH	9BA	0000000000000000
CCP1CON2H	956	0------100-00000	CCP2PRL	988	1111111111111111	CCP4CON1L	9BC	0-00000000000000
CCP1CON3H	95A	0000------0-00--	CCP2PRH	98A	1111111111111111	CCP4CON1H	9BE	00--000000000000
CCP1STATL	95C	-----0--00xx0000	CCP2RAL	98C	0000000000000000	CCP4CON2L	9C0	00-0----00000000
CCP1STATH	95E	----------00000	CCP2RBL	990	0000000000000000	CCP4CON2H	9C2	0------100-00000
CCP1TMRL	960	0000000000000000	CCP2BUFL	994	0000000000000000	CCP4CON3H	9C6	0000------0-00--
CCP1TMRH	962	0000000000000000	CCP2BUFH	996	0000000000000000	CCP4STATL	9C8	-----0--00xx0000
CCP1PRL	964	1111111111111111	CCP3CON1L	998	0-00000000000000	CCP4STATH	9CA	----------00000
CCP1PRH	966	1111111111111111	CCP3CON1H	99A	00--000000000000	CCP4TMRL	9CC	0000000000000000
CCP1RAL	968	0000000000000000	CCP3CON2L	99C	00-0----00000000	CCP4TMRH	9CE	0000000000000000
CCP1RBL	96C	0000000000000000	CCP3CON2H	99E	0------100-00000	CCP4PRL	9D0	1111111111111111
CCP1BUFL	970	0000000000000000	CCP3CON3H	9A2	0000------0-00--	CCP4PRH	9D2	1111111111111111
CCP1BUFH	972	0000000000000000	CCP3STATL	9A4	-----0--00xx0000	CCP4RAL	9D4	0000000000000000
CCP2CON1L	974	0-00000000000000	CCP3STATH	9A6	----------00000	CCP4RBL	9D8	0000000000000000
CCP2CON1H	976	00--000000000000	CCP3TMRL	9A8	0000000000000000	CCP4BUFL	9DC	0000000000000000
CCP2CON2L	978	00-0----00000000	CCP3TMRH	9AA	0000000000000000	CCP4BUFH	9DE	0000000000000000
CCP2CON2H	97A	0------100-00000	CCP3PRL	9AC	1111111111111111			
CCP2CON3H	97E	0000------0-00--	CCP3PRH	9AE	1111111111111111			

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address and Reset values are in hexadecimal and binary, respectively.

TABLE 4-10: SLAVE SFR BLOCK A00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
DMA			DMACH0	AC4	---0-00000000000	DMACH1	ACE	---0-00000000000
DMACON	ABC	0-0----------0	DMAINT0	AC6	0000000000000--0	DMAINT1	AD0	0000000000000--0
DMABUF	ABE	0000000000000000	DMASRC0	AC8	0000000000000000	DMASRC1	AD2	0000000000000000
DMAL	AC0	0001000000000000	DMADST0	ACA	0000000000000000	DMADST1	AD4	0000000000000000
DMAH	AC2	0001000000000000	DMACNT0	ACC	0000000000000001	DMACNT1	AD6	0000000000000001

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address and Reset values are in hexadecimal and binary, respectively.

4.3.8 SLAVE ECC CONTROL/STATUS REGISTERS

REGISTER 4-9: ECCCONL: ECC FAULT INJECTION CONFIGURATION REGISTER LOW

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

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	FLTINMJ
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 15-1 **Unimplemented:** Read as ‘0’

bit 0 **FLTINJ:** Fault Injection Sequence Enable bit
1 = Enabled
0 = Disabled

REGISTER 4-10: ECCCONH: ECC FAULT INJECTION CONFIGURATION REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLT2PTR<7:0>							
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
FLT1PTR<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 15-8 **FLT2PTR<7:0>:** ECC Fault Injection Bit Pointer 2
11111111-00111000 = No Fault injection occurs
00110111 = Fault injection (bit inversion) occurs on bit 55 of ECC bit order
...
00000001 = Fault injection (bit inversion) occurs on bit 1 of ECC bit order
00000000 = Fault injection (bit inversion) occurs on bit 0 of ECC bit order

bit 7-0 **FLT1PTR<7:0>:** ECC Fault Injection Bit Pointer 1
11111111-00111000 = No Fault injection occurs
00110111 = Fault injection occurs on bit 55 of ECC bit order
...
00000001 = Fault injection occurs on bit 1 of ECC bit order
00000000 = Fault injection occurs on bit 0 of ECC bit order

REGISTER 4-82: RPOR22: PERIPHERAL PIN SELECT OUTPUT REGISTER 22

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP175R5(1)	RP175R4(1)	RP175R3(1)	RP175R2(1)	RP175R1(1)	RP175R0(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP174R5(1)	RP174R4(1)	RP174R3(1)	RP174R2(1)	RP174R1(1)	RP174R0(1)
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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP175R<5:0>:** Peripheral Output Function is Assigned to S1RP175 Output Pin bits(1)
(see Table 4-31 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP174R<5:0>:** Peripheral Output Function is Assigned to S1RP174 Output Pin bits(1)
(see Table 4-31 for peripheral function numbers)

Note 1: These are virtual output ports.

EXAMPLE 6-1: CODE EXAMPLE FOR USING MASTER PRIMARY PLL WITH 8 MHz INTERNAL FRC

```
//code example for 50 MIPS system clock using 8MHz FRC
// Select FRC on POR
#pragma  config FNOSC = FRC       // Oscillator Source Selection (Internal Fast RC (FRC))
#pragma  config IESO = OFF
// Enable Clock Switching
#pragma  config FCKSM = CSECMD
int      main()
{
// Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
CLKDIVbits.PLLPRE = 1;            // N1=1
PLLFBDbits.PLLFBDIV = 125;        // M = 125
PLLDIVbits.POST1DIV = 5;          // N2=5
PLLDIVbits.POST2DIV = 1;          // N3=1
// Initiate Clock Switch to FRC with PLL (NOSC=0b001)
__builtin_write_OSCCONH(0x01);
__builtin_write_OSCCONL(OSCCON | 0x01);
// Wait for Clock switch to occur
while    (OSCCONbits.OSWEN!= 0);
}
```

Note: *FPLLO = FPLLI * M/(N1 * N2 * N3); FPLLI = 8; M = 125; N1 = 1; N2 = 5; N3 = 1;* so *FPLLO = 8 * 125/(1 * 5 * 1) = 200 MHz* or *50 MIPS.*

EXAMPLE 6-2: CODE EXAMPLE FOR USING SLAVE PRIMARY PLL WITH 8 MHz INTERNAL FRC

```
//code example for 60 MIPS system clock using 8MHz FRC
// Select Internal FRC at POR
// Select FRC on POR
#pragma config  S1FNOSC = FRC      // Oscillator Source Selection (Internal Fast RC (FRC))
#pragma config  S1IESO = OFF       // Two-speed Oscillator Start-up Enable bit (Start up with
user-selected oscillator source)
// Enable Clock Switching
#pragma config  S1FCKSM = CSECMD
int     main()
{
// Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
CLKDIVbits.PLLPRE = 1;            // N1=1
PLLFBDbits.PLLFBDIV = 150;        // M = 150
PLLDIVbits.POST1DIV = 5;          // N2=5
PLLDIVbits.POST2DIV = 1;          // N3=1
// Initiate Clock Switch to FRC with PLL (NOSC=0b001)
__builtin_write_OSCCONH(0x01);
__builtin_write_OSCCONL(OSCCON | 0x01);
// Wait for Clock switch to occur
while   (OSCCONbits.OSWEN!= 0);
}
```

Note: *FPLLO = FPLLI * M/(N1 * N2 * N3); FPLLI = 8; M = 150; N1 = 1; N2 = 5; N3 = 1;* so *FPLLO = 8 * 150/(1 * 5 * 1) = 240 MHz* or *60 MIPS.*

REGISTER 6-13: CLKDIV: CLOCK DIVIDER REGISTER (SLAVE)

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE2(1)	DOZE1(1)	DOZE0(1)	DOZEN(2,3)	FRCDIV2	FRCDIV1	FRCDIV0
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PLLPRE3(4)	PLLPRE2(4)	PLLPRE1(4)	PLLPRE0(4)
bit 7							bit 0

Legend:	r = Reserved 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 **ROI:** Recover on Interrupt bit
1 = Interrupts will clear the DOZEN bit and the processor clock, and the peripheral clock ratio is set to 1:1
0 = Interrupts have no effect on the DOZEN bit

bit 14-12 **DOZE<2:0>:** Processor Clock Reduction Select bits(1)
111 = FP divided by 128
110 = FP divided by 64
101 = FP divided by 32
100 = FP divided by 16
011 = FP divided by 8 (default)
010 = FP divided by 4
001 = FP divided by 2
000 = FP divided by 1

bit 11 **DOZEN:** Doze Mode Enable bit(2,3)
1 = DOZE<2:0> field specifies the ratio between the peripheral clocks and the processor clocks
0 = Processor clock and peripheral clock ratio is forced to 1:1

bit 10-8 **FRCDIV<2:0>:** Internal Fast RC Oscillator Postscaler bits
111 = FRC divided by 256
110 = FRC divided by 64
101 = FRC divided by 32
100 = FRC divided by 16
011 = FRC divided by 8
010 = FRC divided by 4
001 = FRC divided by 2
000 = FRC divided by 1 (default)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **Reserved:** Read as '0'

Note 1: The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

2: This bit is cleared when the ROI bit is set and an interrupt occurs.

3: The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

4: PLLPRE<3:0> may be updated while the PLL is operating, but the VCO may overshoot.

REGISTER 6-15: PLLDIV: PLL OUTPUT DIVIDER REGISTER (SLAVE)

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	POST1DIV<2:0>(1,2)			—	POST2DIV<2:0>(1,2)		
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 15-10 **Unimplemented:** Read as '0'

bit 9-8 **VCODIV<1:0>:** PLL VCO Output Divider Select bits

11 = FVCO
10 = FVCO/2
01 = FVCO/3
00 = FVCO/4

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

bit 6-4 **POST1DIV<2:0>:** PLL Output Divider #1 Ratio bits(1,2)

POST1DIV<2:0> can have a valid value, from 1 to 7 (POST1DIVx value should be greater than or equal to the POST2DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **POST2DIV<2:0>:** PLL Output Divider #2 Ratio bits(1,2)

POST2DIV<2:0> can have a valid value, from 1 to 7 (POST2DIVx value should be less than or equal to the POST1DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

Note 1: The POST1DIVx and POST2DIVx divider values must not be changed while the PLL is operating.

2: The default values for POST1DIVx and POST2DIVx are 4 and 1, respectively, yielding a 150 MHz system source clock.

REGISTER 7-10: PMD1: SLAVE PERIPHERAL MODULE DISABLE 1 CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	T1MD	QEIMD	PWMMD	—
bit 15							bit 8

R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0
I2C1MD	—	U1MD	—	SPI1MD	—	—	ADC1MD
bit 7							bit 0

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

bit 15-12 **Unimplemented:** Read as '0'

bit 11 **T1MD:** Timer1 Module Disable bit
1 = Timer1 module is disabled
0 = Timer1 module is enabled

bit 10 **QEIMD:** QEI Module Disable bit
1 = QEI module is disabled
0 = QEI module is enabled

bit 9 **PWMMD:** PWM Module Disable bit
1 = PWM module is disabled
0 = PWM module is enabled

bit 8 **Unimplemented:** Read as '0'

bit 7 **I2C1MD:** I2C1 Module Disable bit
1 = I2C1 module is disabled
0 = I2C1 module is enabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **U1MD:** UART1 Module Disable bit
1 = UART1 module is disabled
0 = UART1 module is enabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **SPI1MD:** SPI1 Module Disable bit
1 = SPI1 module is disabled
0 = SPI1 module is enabled

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **ADC1MD:** ADC Module Disable bit
1 = ADC module is disabled
0 = ADC module is enabled

Table 12-2 shows the truth table that describes how the Quadrature signals are decoded.

TABLE 12-2: TRUTH TABLE FOR QUADRATURE ENCODER

Current Quadrature State		Previous Quadrature State		Action
QA	QB	QA	QB	
1	1	1	1	No count or direction change
1	1	1	0	Count up
1	1	0	1	Count down
1	1	0	0	Invalid state change; ignore
1	0	1	1	Count down
1	0	1	0	No count or direction change
1	0	0	1	Invalid state change; ignore
1	0	0	0	Count up
0	1	1	1	Count up
0	1	1	0	Invalid state change; ignore
0	1	0	1	No count or direction change
0	1	0	0	Count down
0	0	1	1	Invalid state change; ignore
0	0	1	0	Count down
0	0	0	1	Count up
0	0	0	0	No count or direction change

Figure 12-2 illustrates the simplified block diagram of the QEI module. The QEI module consists of decoder logic to interpret the Phase A (QEAx) and Phase B (QEBx) signals, and an up/down counter to accumulate the count. The counter pulses are generated when the Quadrature state changes. The count direction information must be maintained in a register until a direction change is detected. The module also includes digital noise filters, which condition the input signal.

The QEI module consists of the following major features:

- Four Input Pins: Two Phase Signals, an Index Pulse and a Home Pulse
- Programmable Digital Noise Filters on Inputs
- Quadrature Decoder providing Counter Pulses and Count Direction
- Count Direction Status
- 4x Count Resolution
- Index (INDXx) Pulse to Reset the Position Counter
- General Purpose 32-Bit Timer/Counter mode
- Interrupts generated by QEI or Counter Events
- 32-Bit Velocity Counter
- 32-Bit Position Counter
- 32-Bit Index Pulse Counter
- 32-Bit Interval Timer
- 32-Bit Position Initialization/Capture Register
- 32-Bit Compare Less Than and Greater Than Registers
- External Up/Down Count mode
- External Gated Count mode
- External Gated Timer mode
- Interval Timer mode

14.0 SERIAL PERIPHERAL INTERFACE (SPI)

Note 1: This data sheet summarizes the features of the dsPIC33CH128MP508 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Serial Peripheral Interface (SPI) with Audio Codec Support"** (DS70005136) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: The SPI is Identical for both Master core and Slave core. The x is common for both Master and Slave (where the x represents the number of the specific module being addressed). The number of SPI modules available on the Master and Slave is different and they are located in different SFR locations.

3: All associated register names are the same on the Master core and the Slave core. The Slave code will be developed in a separate project in MPLAB® X IDE with the device selection, dsPIC33CH128MP508**S1**, where the **S1** indicates the Slave device. The Master is SPI1 and SPI2, and the Slave is SPI1.

Table 14-1 shows an overview of the SPI module.

TABLE 14-1: SPI MODULE OVERVIEW

	Number of SPI Modules	Identical (Modules)
Master Core	2	Yes
Slave Core	1	Yes

The Serial Peripheral Interface (SPI) module is a synchronous serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The SPI module is compatible with the Motorola® SPI and SIOP interfaces. All devices in the dsPIC33CH128MP508 family include three SPI modules; two SPIs for the Master core and one for the Slave core. One of the SPI modules can work up to 50 MHz speed when selected as a non-PPS pin. For the Master core, it will be SPI2 and for the Slave core, it will be SPI1. The selection is done using the SPI2PIN bit (FDEVOPT<13>) for the Master and the S1SPI1PIN bit (FS1DEVOPT<13>) for the Slave. If the bit for SPI2PIN/S1SPI1PIN is '1', the PPS pin will be used. If the SPI2PIN/S1SPI1PIN is '0', it will use the dedicated SPI pads.

The module supports operation in two Buffer modes. In Standard mode, data is shifted through a single serial buffer. In Enhanced Buffer mode, data is shifted through a FIFO buffer. The FIFO level depends on the configured mode.

Note: FIFO depth for this device is four (in 8-Bit Data mode).

Variable length data can be transmitted and received, from 2 to 32 bits.

Note: Do not perform Read-Modify-Write operations (such as bit-oriented instructions) on the SPIxBUF register in either Standard or Enhanced Buffer mode.

The module also supports a basic framed SPI protocol while operating in either Master or Slave mode. A total of four framed SPI configurations are supported.

The module also supports Audio modes. Four different Audio modes are available.

- I^2S mode
- Left Justified mode
- Right Justified mode
- PCM/DSP mode

In each of these modes, the serial clock is free-running and audio data is always transferred.

If an audio protocol data transfer takes place between two devices, then usually one device is the Master and the other is the Slave. However, audio data can be transferred between two Slaves. Because the audio protocols require free-running clocks, the Master can be a third-party controller. In either case, the Master generates two free-running clocks: SCKx and LRC (Left, Right Channel Clock/$\overline{SS}$x/FSYNC).

The SPI serial interface consists of four pins:

- SDIx/S1SDIx: Serial Data Input
- SDOx/S1SDOx: Serial Data Output
- SCKx/S1SCKx: Shift Clock Input or Output
- $\overline{SS}$x/$\overline{S1SS}$x: Active-Low Slave Select or Frame Synchronization I/O Pulse

The SPI module can be configured to operate using two, three or four pins. In the 3-pin mode, $\overline{SS}$x/$\overline{S1SS}$x is not used. In the 2-pin mode, both SDOx/S1SDOx and $\overline{SS}$x/$\overline{S1SS}$x are not used.

21.0 SPECIAL FEATURES

Note: This data sheet summarizes the features of the dsPIC33CH128MP508 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section of the *"dsPIC33/PIC24 Family Reference Manual*", which is available from the Microchip web site (www.microchip.com).

The dsPIC33CH128MP508 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection and CodeGuard™ Security
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation
- Brown-out Reset (BOR)

21.1 Configuration Bits

In dsPIC33CH128MP508 family devices, the Configuration Words are implemented as volatile memory. This means that configuration data will get loaded to volatile memory (from the Flash Configuration Words) each time the device is powered up. Configuration data is stored at the end of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 21-1. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration Shadow registers during device Resets.

Note: Configuration data is reloaded on all types of device Master Resets. Slave Resets do not load the Configuration registers. It is recommended not to change the Slave Configuration register without resetting the Slave along with the Master (S1MSRE = 1).

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Words for configuration data in their code for the compiler. This is to make certain that program code is not stored in this address when the code is compiled. Program code executing out of configuration space will cause a device Reset. The Master code, as well as the Slave code, are located in Flash memory. Table 21-1 shows the Master and the Slave Configuration registers and their address locations in Flash memory.

Slave Configuration bits are located in the Master Flash and loaded during a Master Reset.

Note: Performing a page erase operation on the last page of program memory clears the Flash Configuration Words.

TABLE 21-1: CONFIGURATION WORD ADDRESSES

Register	64k Address	128k Address
Master/General Configuration Registers		
FSEC	00AF00	015F00
FBSLIM	00AF10	015F10
FSIGN	00AF14	015F14
FOSCSEL	00AF18	015F18
FOSC	00AF1C	015F1C
FWDT	00AF20	015F20
FPOR	00AF24	015F24
FICD	00AF28	015F28
FDMTIVTL	00AF2C	015F2C
FDMTIVTH	00AF30	015F30
FDMTCNTL	00AF34	015F34
FDMTCNTH	00AF38	015F38
FDMT	00AF3C	015F3C
FDEVOPT	00AF40	015F40
FALTREG	00AF44	015F44
FMBXM	00AF48	015F48
FMBXHS1	00AFC4	015F4C
FMBXHS2	00AF50	015F50
FMBXHSEN	00AF54	015F54
FCFGPRA0	00AF58	015F58
FCFGPRB0	00AF60	015F60
FCFGPRC0	00AF68	015F68
FCFGPRD0	00AF70	015F70
FCFGPRE0	00AF78	015F7C
Slave Configuration Registers		
FS1OSCSEL	00AF80	015F80
FS1OSC	00AF84	015F84
FS1WDT	00AF88	015F88
FS1POR	00AF8C	015F8C
FS1ICD	00AF90	015F90
FS1DEVOPT	00AF94	015F94
FS1ALTREG	00AF98	015F98

REGISTER 21-32: DEVREV: DEVICE REVISION REGISTER

R	R	R	R	R	R	R	R
DEVREV<23:16>							
bit 23							bit 16

R	R	R	R	R	R	R	R
DEVREV<15:8>							
bit 15							bit 8

R	R	R	R	R	R	R	R
DEVREV<7:0>							
bit 7							bit 0

Legend:	R = Read-only bit	U = Unimplemented bit

bit 23-0 **DEVREV<23:0>:** Device Revision bits

REGISTER 21-33: DEVID: DEVICE ID REGISTERS

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7(1)	DEV6(1)	DEV5(1)	DEV4(1)	DEV3(1)	DEV2(1)	DEV1(1)	DEV0(1)
bit 7							bit 0

Legend:	R = Read-only bit	U = Unimplemented bit

bit 23-16 **Unimplemented:** Read as '1'

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits

`1000 0111` = dsPIC33CH128MP508 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits(1)

Note 1: See Table 21-5 for the list of Device Identifier bits.

TABLE 24-7: DC CHARACTERISTICS: OPERATING CURRENT (IDD) (MASTER RUN/SLAVE SLEEP)

DC CHARACTERISTICS	Master (Run) + Slave (Sleep)		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typ.**	**Max.**	**Units**	**Conditions**		
Operating Current (IDD)(1)						
DC20b	7.9	9.8	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	8.0	13.4	mA	+25°C		
	8.2	19.5	mA	+85°C		
	12.2	26.3	mA	+125°C		
DC21b	10.3	12.4	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 80 MHz)
	10.5	16.0	mA	+25°C		
	10.6	22.1	mA	+85°C		
	14.6	28.7	mA	+125°C		
DC22b	14.2	16.5	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	14.4	20.3	mA	+25°C		
	14.5	26.3	mA	+85°C		
	18.4	32.6	mA	+125°C		
DC23b	22.3	25.4	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	22.5	29.4	mA	+25°C		
	22.4	34.9	mA	+85°C		
	26.4	40.7	mA	+125°C		
DC24b	25.6	29.0	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	25.8	33.1	mA	+25°C		
	25.7	38.2	mA	+85°C		
	29.4	43.8	mA	+125°C		

Note 1: IDD is primarily a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements are as follows:

- FIN = 8 MHz, FPFD = 8 MHz
- CLKO is configured as an I/O input pin in the Configuration Word
- All I/O pins are configured as output low
- $\overline{\text{MCLR}}$ = VDD, WDT and FSCM are disabled
- CPU, SRAM, program memory and data memory are operational
- No peripheral modules are operating or being clocked (all defined PMDx bits are set)
- CPU is executing `while(1)` statement
- JTAG is disabled

TABLE 24-18: I/O PIN INPUT SPECIFICATIONS

Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.(1)**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
DI10		Any I/O Pin and $\overline{MCLR}$	VSS	—	0.2 VDD	V	
DI18		I/O Pins with SDAx, SCLx	VSS	—	0.3 VDD	V	SMBus disabled
DI19		I/O Pins with SDAx, SCLx	VSS	—	0.8	V	SMBus enabled
	VIH	**Input High Voltage**					
DI20		I/O Pins Not 5V Tolerant(3)	0.8 VDD	—	VDD	V	
		5V Tolerant I/O Pins and $\overline{MCLR}$(3)	0.8 VDD	—	5.5	V	
		5V Tolerant I/O Pins with SDAx, SCLx(3)	0.8 VDD	—	5.5	V	SMBus disabled
		5V Tolerant I/O Pins with SDAx, SCLx(3)	2.1	—	5.5	V	SMBus enabled
		I/O Pins with SDAx, SCLx Not 5V Tolerant(3)	0.8 VDD	—	VDD	V	SMBus disabled
		I/O Pins with SDAx, SCLx Not 5V Tolerant(3)	2.1	—	VDD	V	SMBus enabled
DI30	ICNPU	**Input Change Notification Pull-up Current(2,4)**	175	360	545	µA	VDD = 3.6V, VPIN = VSS
DI31	ICNPD	**Input Change Notification Pull-Down Current(4)**	65	215	360	µA	VDD = 3.6V, VPIN = VDD

Note 1: Data in "Typ." column is at 3.3V, +25°C unless otherwise stated.

2: Negative current is defined as current sourced by the pin.

3: See the **"Pin Diagrams"** section for the 5V tolerant I/O pins.

4: All parameters are characterized but not tested during manufacturing.

TABLE 24-19: I/O PIN INPUT SPECIFICATIONS

Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended						
Param No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
DI50	IIL	**Input Leakage Current(1)**				
		I/O Pins 5V Tolerant(2)	-700	+700	nA	VPIN = VSS or VDD
		I/O Pins Not 5V Tolerant(2)	-700	+700	nA	
		$\overline{MCLR}$	-700	+700	nA	
		OSCI	-700	+700	nA	XT and HS modes

Note 1: Negative current is defined as current sourced by the pin.

2: See the **"Pin Diagrams"** section for the 5V tolerant I/O pins. All parameters are characterized but not tested during manufacturing.