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	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	69
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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch128mp208-e-pt

3.3.6 ECC CONTROL REGISTERS

REGISTER 3-9: 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 3-10: 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 3-55: RPINR21: PERIPHERAL PIN SELECT INPUT REGISTER 21

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
REFOIR7	REFOIR6	REFOIR5	REFOIR4	REFOIR3	REFOIR2	REFOIR1	REFOIR0
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
SS1R7	SS1R6	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
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 **REFOIR<7:0>:** Assign Reference Clock Input (REFOI) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **SS1R<7:0>:** Assign SPI1 Slave Select ($\overline{SS1}$) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-56: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK2R7	SCK2R6	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
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
SDI2R7	SDI2R6	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
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 **SCK2R<7:0>:** Assign SPI2 Clock Input (SCK2IN) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **SDI2R<7:0>:** Assign SPI2 Data Input (SDI2) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-78: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP53R5	RP53R4	RP53R3	RP53R2	RP53R1	RP53R0
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
—	—	RP52R5	RP52R4	RP52R3	RP52R2	RP52R1	RP52R0
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 **RP53<5:0>:** Peripheral Output Function is Assigned to RP53 Output Pin bits (see Table 3-33 for peripheral function numbers)

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

bit 5-0 **RP52R<5:0>:** Peripheral Output Function is Assigned to RP52 Output Pin bits (see Table 3-33 for peripheral function numbers)

REGISTER 3-79: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTER 11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP55R5	RP55R4	RP55R3	RP55R2	RP55R1	RP55R0
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
—	—	RP54R5	RP54R4	RP54R3	RP54R2	RP54R1	RP54R0
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 **RP55R<5:0>:** Peripheral Output Function is Assigned to RP55 Output Pin bits (see Table 3-33 for peripheral function numbers)

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

bit 5-0 **RP54R<5:0>:** Peripheral Output Function is Assigned to RP54 Output Pin bits (see Table 3-33 for peripheral function numbers)

REGISTER 3-126: C1TXREQH: CAN TRANSMIT REQUEST REGISTER HIGH

S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0
TXREQ<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	S/HC-0
TXREQ<23:16>							
bit 7							bit 0

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

bit 15-0 **TXREQ<31:16>:** Unimplemented

REGISTER 3-127: C1TXREQL: CAN TRANSMIT REQUEST REGISTER LOW

S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0
TXREQ<15:8>							
bit 15							bit 8

S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0s
TXREQ<7:1>							TXREQ0
bit 7							bit 0

Legend:	S = Settable bit	HC = Hardware Clearable 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 15-8 **TXREQ<15:8>:** Unimplemented

bit 7-1 **TXREQ<7:1>:** Message Send Request bits

TXEN = 1 (object configured as a transmit object):
Setting this bit to '1' requests sending a message. The bit will automatically clear when the message(s) queued in the object is (are) successfully sent. This bit can NOT be used for aborting a transmission.

TXEN = 0 (object configured as a receive object):
This bit has no effect.

bit 0 **TXREQ0:** Transmit Queue Message Send Request bit

Setting this bit to '1' requests sending a message. The bit will automatically clear when the message(s) queued in the object is (are) successfully sent. This bit can NOT be used for aborting a transmission.

REGISTER 3-163: ADCON5L: ADC CONTROL REGISTER 5 LOW

HSC/R-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRRDY	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRPWR	—	—	—	—	—	—	—
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 **SHRRDY:** Shared ADC Core Ready Flag bit
1 = ADC core is powered and ready for operation
0 = ADC core is not ready for operation

bit 14-8 **Unimplemented:** Read as '0'

bit 7 **SHRPWR:** Shared ADC Core Power Enable bit
1 = ADC core is powered
0 = ADC core is off

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

3.10 Peripheral Trigger Generator (PTG)

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 **"Peripheral Trigger Generator (PTG)"** (DS70000669) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com)

Table 3-43 shows an overview of the PTG module.

TABLE 3-43: PTG MODULE OVERVIEW

	No. of PTG Modules	Identical (Modules)
Master	1	NA
Slave	None	NA

The dsPIC33CH128MP508 family Peripheral Trigger Generator (PTG) module is a user-programmable sequencer that is capable of generating complex trigger signal sequences to coordinate the operation of other peripherals. The PTG module is designed to interface with the modules, such as an Analog-to-Digital Converter (ADC), output compare and PWM modules, timers and interrupt controllers.

3.10.1 FEATURES

- Behavior is Step Command-Driven:
  - Step commands are eight bits wide
- Commands are Stored in a Step Queue:
  - Queue depth is parameterized (8-32 entries)
  - Programmable Step execution time (Step delay)
- Supports the Command Sequence Loop:
  - Can be nested one-level deep
  - Conditional or unconditional loop
  - Two 16-bit loop counters
- 16 Hardware Input Triggers:
  - Sensitive to either positive or negative edges, or a high or low level
- One Software Input Trigger
- Generates up to 32 Unique Output Trigger Signals
- Generates Two Types of Trigger Outputs:
  - Individual
  - Broadcast
- Strobed Output Port for Literal Data Values:
  - 5-bit literal write (literal part of a command)
  - 16-bit literal write (literal held in the PTGL0 register)
- Generates up to Ten Unique Interrupt Signals
- Two 16-Bit General Purpose Timers
- Flexible Self-Contained Watchdog Timer (WDT) to Set an Upper Limit to Trigger Wait Time
- Single-Step Command Capability in Debug mode
- Selectable Clock (system, Pulse-Width Modulator (PWM) or ADC)
- Programmable Clock Divider

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

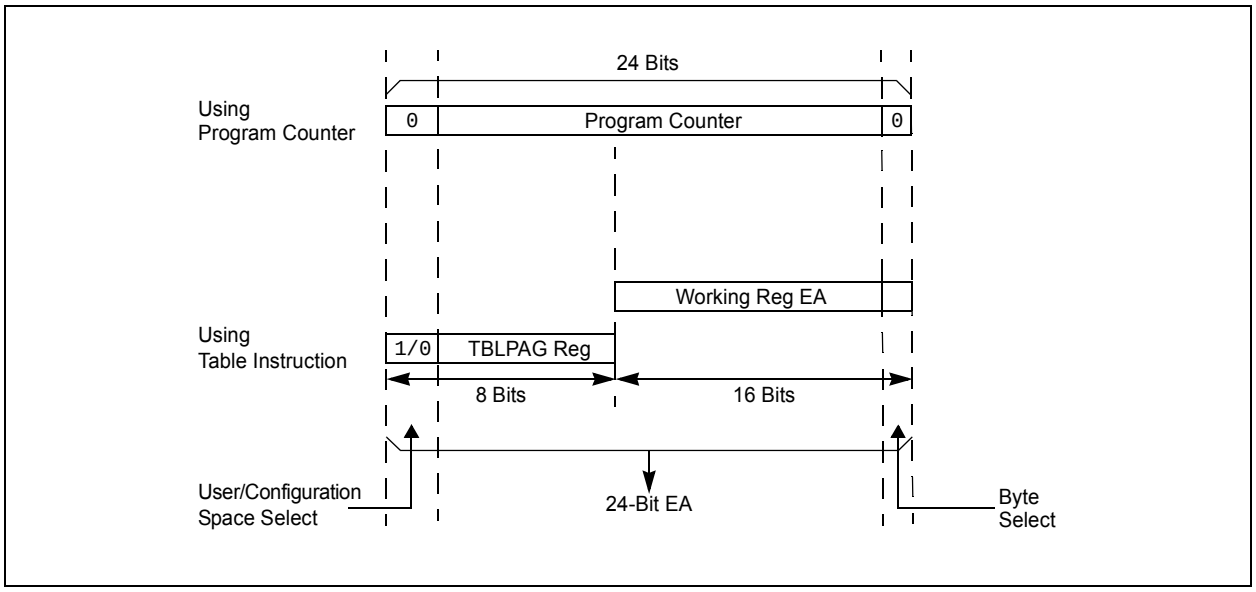
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE<23:16>							
bit 7							bit 0

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.

FIGURE 4-13: ADDRESSING FOR TABLE REGISTERS



4.6.5.5 Virtual Connections

The dsPIC33CH128MP508S1 family devices support six virtual S1RPn pins (S1RP170-S1RP175), which are identical in functionality to all other S1RPn pins, with the exception of pinouts. These six pins are internal to the devices and are not connected to a physical device pin.

These pins provide a simple way for inter-peripheral connection without utilizing a physical pin. For example, the output of the analog comparator can be connected to S1RP170 and the PWM control input can be configured for S1RP170 as well. This configuration allows the analog comparator to trigger PWM Faults without the use of an actual physical pin on the device.

4.6.5.6 Slave PPS Inputs to Master Core PPS

The dsPIC33CH128MP508S1 Slave core subsystem PPS has connections to the Master core subsystem virtual PPS (S1RPV5-S1RPV0) output blocks. These inputs are mapped as S1RP175, S1RP174, S1RP173, S1RP172, S1RP171 and S1RP170.

The S1RPn inputs, S1RP1-S1RP13, are connected to internal signals from both the Master and Slave core subsystems. Additionally, the Master core virtual PPS output blocks (RPV5-RPV0) are connected to the Slave core PPS circuitry.

There are virtual pins in PPS to share between Master and Slave:

- RP181 is for Master input (RPV5)
- RP180 is for Master input (RPV4)
- RP179 is for Master input (RPV3)
- RP178 is for Master input (RPV2)
- RP177 is for Master input (RPV1)
- RP176 is for Master input (RPV0)
- S1RP175 is for Slave input (S1RPV5)
- S1RP174 is for Slave input (S1RPV4)
- S1RP173 is for Slave input (S1RPV3)
- S1RP172 is for Slave input (S1RPV2)
- S1RP171 is for Slave input (S1RPV1)
- S1RP170 is for Slave input (S1RPV0)

The idea of the S1RPVn (Remappable Pin Virtual) is to interconnect between Master and Slave without an I/O pin. For example, the Master UART receiver can be connected to the Slave UART transmit using S1RPVn and data communication can happen from Slave to Master without using any physical pin.

REGISTER 4-106: ADTRIGnL/ADTRIGnH: ADC CHANNEL TRIGGER n(x) SELECTION REGISTERS LOW AND HIGH (x = 0 TO 19; n = 0 TO 4) (CONTINUED)

bit 4-0 **TRGSRCx<4:0>:** Common Interrupt Enable for Corresponding Analog Inputs bits (TRGSRC0 to TRGSRC20 – Even)

11111 = ADTRG31 (PPS input)
11110 = Master PTG
11101 = Slave CLC1
11100 = Master CLC1
11011 = Reserved
11010 = Reserved
11001 = Master PWM3 Trigger 2
11000 = Master PWM1 Trigger 2
10111 = Slave SCCP4 PWM/IC interrupt
10110 = Slave SCCP3 PWM/IC interrupt
10101 = Slave SCCP2 PWM/IC interrupt
10100 = Slave SCCP1 PWM/IC interrupt
10011 = Reserved
10010 = Reserved
10001 = Reserved
10000 = Reserved
01111 = Slave PWM8 Trigger 1
01110 = Slave PWM7 Trigger 1
01101 = Slave PWM6 Trigger 1
01100 = Slave PWM5 Trigger 1
01011 = Slave PWM4 Trigger 2
01010 = Slave PWM4 Trigger 1
01001 = Slave PWM3 Trigger 2
01000 = Slave PWM3 Trigger 1
00111 = Slave PWM2 Trigger 2
00110 = Slave PWM2 Trigger 1
00101 = Slave PWM1 Trigger 2
00100 = Slave PWM1 Trigger 1
00011 = Reserved
00010 = Level software trigger
00001 = Common software trigger
00000 = No trigger is enabled

REGISTER 6-5: PLLDIV: PLL OUTPUT DIVIDER REGISTER (MASTER)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	VCODIV1	VCODIV0
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
—	POST1DIV2(1,2)	POST1DIV1(1,2)	POST1DIV0(1,2)	—	POST2DIV2(1,2)	POST2DIV1(1,2)	POST2DIV0(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.

Example 6-6 illustrates code for using the Master PLL with an 8 MHz internal FRC.

EXAMPLE 6-6: CODE EXAMPLE FOR USING MASTER 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);

        // Wait for PLL to lock
        while (OSCCONbits.LOCK!= 1);
}
```

Note: *FPLLO = FPLLI * M/(N1 * N2 * N3); FPLLI = 8; M = 125; N1 = 1; N2 = 5; N3 = 1;* so *FPLLO = 10 * 100/(1 * 5 * 1) = 200 MHz* or *50 MIPS.*

Example 6-7 illustrates code for using the Slave PLL with an 8 MHz internal FRC.

EXAMPLE 6-7: CODE EXAMPLE FOR USING SLAVE PLL WITH 8 MHz INTERNAL FRC

```
//code example for 60 MIPS system clock using 8MHz FRC

// 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);
        // Wait for PLL to lock
        while (OSCCONbits.LOCK!= 1);
}
```

Note: *FPLLO = FPLLI * M/(N1 * N2 * N3); FPLLI = 8; M = 150; N1 = 1; N2 = 5; N3 = 1;* so *FPLLO = 10 * 100/(1 * 5 * 1) = 240 MHz* or *60 MIPS.*

TABLE 8-3: DMA CHANNEL TRIGGER SOURCES (SLAVE)

CHSEL<6:0>		Trigger (Interrupt)	CHSEL<6:0>		Trigger (Interrupt)	CHSEL<6:0>		Trigger (Interrupt)
0000000	00h	INT0 – External Interrupt 0	0100010	22h	PWM Generator 7	1000100	44h	CLC1 Interrupt
0000001	01h	SCCP1 Interrupt	0100011	23h	PWM Generator 8	1000101	45h	CLC2 Interrupt
0000010	02h	SPI1 Receiver	0100100	24h	PWM Event C	1000110	46h	SPI1 – Fault Interrupt
0000011	03h	SPI1 Transmitter	0100101	25h	(Reserved, do not use)	1000111	47h	(Reserved, do not use)
0000100	04h	UART1 Receiver	0100110	26h	(Reserved, do not use)	1001000	48h	(Reserved, do not use)
0000101	05h	UART1 Transmitter	0100111	27h	ADC1 Group Convert Done	1001001	49h	(Reserved, do not use)
0000110	06h	ECC Single Bit Error	0101000	28h	ADC Done AN0	1001010	4Ah	MSI Master Initiated Slave IRQ
0000111	07h	NVM Write Complete	0101001	29h	ADC Done AN1	1001011	4Bh	MSI Protocol A
0001000	08h	INT1 – External Interrupt 1	0101010	2Ah	ADC Done AN2	1001100	4Ch	MSI Protocol B
0001001	09h	SI2C1 – I2C1 Slave Event	0101011	2Bh	ADC Done AN3	1001101	4Dh	MSI Protocol C
0001010	0Ah	MI2C1 – I2C1 Master Event	0101100	2Ch	ADC Done AN4	1001110	4Eh	MSI Protocol D
0001010	0Bh	INT2 – External Interrupt 2	0101101	2Dh	ADC Done AN5	1001111	4Fh	MSI Protocol E
0001100	0Ch	SCCP2 Interrupt	0101110	2Eh	ADC Done AN6	1010000	50h	MSI Protocol F
0001101	0Dh	INT3 – External Interrupt 3	0101111	2Fh	ADC Done AN7	1010001	51h	MSI Protocol G
0001110	0Eh	(Reserved, do not use)	0110000	30h	ADC Done AN8	1010010	52h	MSI Protocol H
0001111	0Fh	(Reserved, do not use)	0110001	31h	ADC Done AN9	1010011	53h	MSI Slave Read FIFO Data Ready IRQ
0010000	10h	(Reserved, do not use)	0110010	32h	ADC Done AN10	1010100	54h	MSI Slave Write FIFO Empty IRQ
0010001	11h	(Reserved, do not use)	0110011	33h	ADC Done AN11	1010101	55h	MSI FIFO Fault (Over/Underflow)
0010010	12h	SCCP3 Interrupt	0110100	34h	ADC Done AN12	1010110	56h	MSI Master Reset IRQ
0010011	13h	(Reserved, do not use)	0110101	35h	ADC Done AN13	1010111	57h	PWM Event D
0010100	14h	(Reserved, do not use)	0110110	36h	ADC Done AN14	1011000	58h	PWM Event E
0010101	15h	SCCP4 Interrupt	0110111	37h	ADC Done AN15	1011001	59h	PWM Event F
0010110	16h	(Reserved, do not use)	0111000	38h	ADC Done AN16	1011010	5Ah	Master ICD Breakpoint Interrupt
0010111	17h	(Reserved, do not use)	0111001	39h	ADC Done AN17	1011011	5Bh	(Reserved, do not use)
0011000	18h	(Reserved, do not use)	0111010	3Ah	(Reserved, do not use)	1011100	5Ch	(Reserved, do not use)
0011001	19h	PWM Event A	0111010	3Bh	ADC Done AN19	1011101	5Dh	(Reserved, do not use)
0011010	1Ah	(Reserved, do not use)	0111100	3Ch	(Reserved, do not use)	1011110	5Eh	Master Clock Fail Interrupt
0011011	1Bh	PWM Event B	0111101	3Dh	(Reserved, do not use)	1011111	5Fh	ADC FIFO Ready Interrupt
0011100	1Ch	PWM Generator 1	0111110	3Eh	(Reserved, do not use)	1100000	60h	CLC3 Positive Edge Interrupt
0011101	1Dh	PWM Generator 2	0111111	3Fh	(Reserved, do not use)	1100001	61h	CLC4 Positive Edge Interrupt
0011110	1Eh	PWM Generator 3	1000000	40h	AD1FLTR1 – Oversample Filter 1	1100001	62h	(Reserved, do not use)
0011111	1Fh	PWM Generator 4	1000001	41h	AD1FLTR2 – Oversample Filter 2	...	...	
0100000	20h	PWM Generator 5	1000010	42h	AD1FLTR3 – Oversample Filter 3	1111111	7Fh	(Reserved, do not use)
0100001	21h	PWM Generator 6	1000011	43h	AD1FLTR4 – Oversample Filter 4			

10.5 Auxiliary Output

The SCCPx modules have an auxiliary (secondary) output that provides other peripherals access to internal module signals. The auxiliary output is intended to connect to other SCCP modules, or other digital peripherals, to provide these types of functions:

- Time Base Synchronization
- Peripheral Trigger and Clock Inputs
- Signal Gating

The type of output signal is selected using the AUXOUT<1:0> control bits (CCPxCON2H<4:3>). The type of output signal is also dependent on the module operating mode.

TABLE 10-5: AUXILIARY OUTPUT

AUXOUT<1:0>	CCSEL	MOD<3:0>	Comments	Signal Description
00	x	xxxx	Auxiliary output disabled	No Output
01	0	0000	Time Base modes	Time Base Period Reset or Rollover
10				Special Event Trigger Output
11				No Output
01	0	0001 through 1111	Output Compare modes	Time Base Period Reset or Rollover
10				Output Compare Event Signal
11				Output Compare Signal
01	1	xxxx	Input Capture modes	Time Base Period Reset or Rollover
10				Reflects the Value of the ICDIS bit
11				Input Capture Event Signal

TABLE 21-3: SLAVE CONFIGURATION REGISTERS MAP

Register Name	Bits 23-16	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
FS1OSCSEL	—	—	—	—	—	—	—	—	—	S1IESO	—	—	—	—	S1FNOSC<2:0>		
FS1OSC	—	—	—	—	—	—	—	—	r(1)	S1FCKSM<1:0>		—	—	—	S1OSCIOFNC	—	—
FS1WDT	—	S1FWDTEN	S1SWDTPS<4:0>					S1WDTWIN<1:0>		S1WINDIS	S1RCLKSEL<1:0>		S1RWDTPS<4:0>				
FS1POR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FS1ICD	—	S1NOBTSWP	—	S1ISOLAT	—	—	—	—	—	r(1)	—	—	—	—	—	S1ICS<1:0>	
FS1DEVOPT	—	S1MSRE	S1SSRE	S1SPI1PIN	—	—	—	—	—	—	—	—	—	S1ALTI2C1	—	—	—
FS1ALTREG	—	—	S1CTXT4<2:0>			—	S1CTXT3<2:0>			—	S1CTXT2<2:0>			—	S1CTXT1<2:0>		

Legend: — = unimplemented bit, read as '1'; r = reserved bit.

Note 1: Bit is reserved, maintain as '1'.

REGISTER 21-4: FOSCSEL CONFIGURATION REGISTER

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

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
IESO	—	—	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

Legend:	PO = Program Once 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 23-8 **Unimplemented:** Read as ‘1’

bit 7 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)

bit 6-3 **Unimplemented:** Read as ‘1’

bit 2-0 **FNOSC<2:0>:** Initial Oscillator Source Selection bits
111 = Internal Fast RC (FRC) Oscillator with Postscaler
110 = Backup Fast RC (BFRC)
101 = LPRC Oscillator
100 = Reserved
011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary (XT, HS, EC) Oscillator
001 = Internal Fast RC Oscillator with PLL (FRCPLL)
000 = Fast RC (FRC) Oscillator

REGISTER 21-17: FMBXHS1 CONFIGURATION REGISTER

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

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
MBXHSD3	MBXHSD2	MBXHSD1	MBXHSD0	MBXHSC3	MBXHSC2	MBXHSC1	MBXHSC0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
MBXHSB3	MBXHSB2	MBXHSB1	MBXHSB0	MBXHSA3	MBXHSA2	MBXHSA1	MBXHSA0
bit 7							bit 0

Legend:	PO = Program Once 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 23-16 **Unimplemented:** Read as '1'

bit 15-12 **MBXHSD<3:0>:** Mailbox Handshake Protocol Block D Register Assignment bits
1111 = MSIxMBXD15 is assigned to Mailbox Handshake Protocol Block D
...
0001 = MSIxMBXD1 is assigned to Mailbox Handshake Protocol Block D
0000 = MSIxMBXD0 is assigned to Mailbox Handshake Protocol Block D

bit 11-8 **MBXHSC<3:0>:** Mailbox Handshake Protocol Block C Register Assignment bits
1111 = MSIxMBXD15 is assigned to Mailbox Handshake Protocol Block C
...
0001 = MSIxMBXD1 is assigned to Mailbox Handshake Protocol Block C
0000 = MSIxMBXD0 is assigned to Mailbox Handshake Protocol Block C

bit 7-4 **MBXHSB<3:0>:** Mailbox Handshake Protocol Block B Register Assignment bits
1111 = MSIxMBXD15 is assigned to Mailbox Handshake Protocol Block B
...
0001 = MSIxMBXD1 is assigned to Mailbox Handshake Protocol Block B
0000 = MSIxMBXD0 is assigned to Mailbox Handshake Protocol Block B

bit 3-0 **MBXHSA<3:0>:** Mailbox Handshake Protocol Block A Register Assignment bits
1111 = MSIxMBXD15 is assigned to Mailbox Handshake Protocol Block A
...
0001 = MSIxMBXD1 is assigned to Mailbox Handshake Protocol Block A
0000 = MSIxMBXD0 is assigned to Mailbox Handshake Protocol Block A

TABLE 24-13: DC CHARACTERISTICS: PWM DELTA CURRENT(1,2,3)

DC CHARACTERISTICS	Master and Slave		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**	
DC100	6	8	mA	-40°C, 3.3V	PWM Output 500 MHz, PWM Input (AFPLLO = 500 MHz), AVCO = 1000 MHz, PLLFBD = 125, APLLDIV = 2
	6	6.7	mA	+25°C, 3.3V	
	6.3	8	mA	+125°C, 3.3V	
DC101	4.9	6	mA	-40°C, 3.3V	PWM Output 500 MHz, PWM Input (AFPLLO = 400 MHz), AVCO = 400 MHz, PLLFBD = 50, APLLDIV = 1
	4.9	5.5	mA	+25°C, 3.3V	
	4.9	5.6	mA	+125°C, 3.3V	
DC102	2.6	3.4	mA	-40°C, 3.3V	PWM Output 500 MHz, PWM Input (AFPLLO = 200 MHz), AVCO = 400 MHz, PLLFBD = 50, APLLDIV = 2
	2.7	3	mA	+25°C, 3.3V	
	2.7	3.2	mA	+125°C, 3.3V	
DC103	1.5	2.9	mA	-40°C, 3.3V	PWM Output 500 MHz, PWM Input (AFPLLO = 100 MHz), AVCO = 400 MHz, PLLFBD = 50, APLLDIV = 4
	1.5	2.1	mA	+25°C, 3.3V	
	1.5	2.2	mA	+125°C, 3.3V	

Note 1: The APLL current is not included. The APLL current will be the same if more than one PWM or all eight PWMs are running.

2: Delta current is for the one instance of PWM running.

3: PWM configured for Low-Resolution mode. All parameters are characterized but not tested during manufacturing.

TABLE 24-14: DC CHARACTERISTICS: APLL DELTA CURRENT

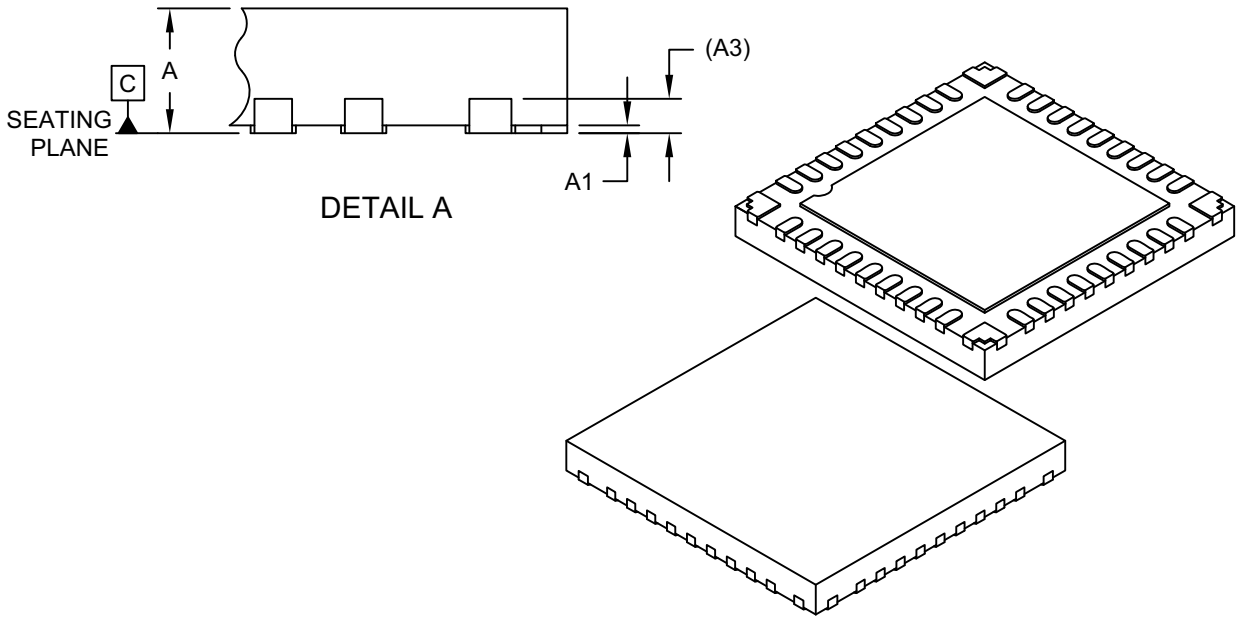
DC CHARACTERISTICS	Master or Slave(2)		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	**Typ.**	**Max.**	**Units**	**Conditions(1)**	
DC110	—	9.4	mA	-40°C.,3.3V	AFPLLO @ 500 MHz, AVCO = 1000 MHz, PLLFBD = 125, APLLDIV = 2
	7.2	9.4	mA	+25°C,3.3V	
	—	18	mA	+125°C,3.3V	
DC111	—	5.7	mA	-40°C.,3.3V	AFPLLO @ 400 MHz, AVCO = 400 MHz, PLLFBD = 50, APLLDIV = 1
	5	5.8	mA	+25°C,3.3V	
	—	14	mA	+125°C,3.3V	
DC112	—	4.7	mA	-40°C.,3.3V	AFPLLO @ 200 MHz, AVCO = 400 MHz, PLLFBD = 50, APLLDIV = 2
	2.9	4.7	mA	+25°C,3.3V	
	—	14	mA	+125°C,3.3V	
DC113	—	4	mA	-40°C.,3.3V	AFPLLO @ 100 MHz, AVCO = 400 MHz, PLLFBD = 50, APLLDIV = 4
	2.3	4	mA	+25°C,3.3V	
	—	12	mA	+125°C,3.3V	

Note 1: The APLL current will be the same if more than one PWM or DAC is run to the APLL clock. All parameters are characterized but not tested during manufacturing.

2: Current is for the APLL for the Master or Slave, not the combined current.

36-Lead Ultra Thin Plastic Quad Flat, No Lead Package (M5) - 5x5 mm Body [UQFN] With Corner Anchors

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		36	
Pitch	e		0.40 BSC	
Overall Height	A	0.50	0.55	0.60
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.152 REF	
Overall Length	D		5.00 BSC	
Exposed Pad Length	D2	3.60	3.70	3.80
Overall Width	E		5.00 BSC	
Exposed Pad Width	E2	3.60	3.70	3.80
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		0.25 REF	

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-436A–M5 Sheet 2 of 2