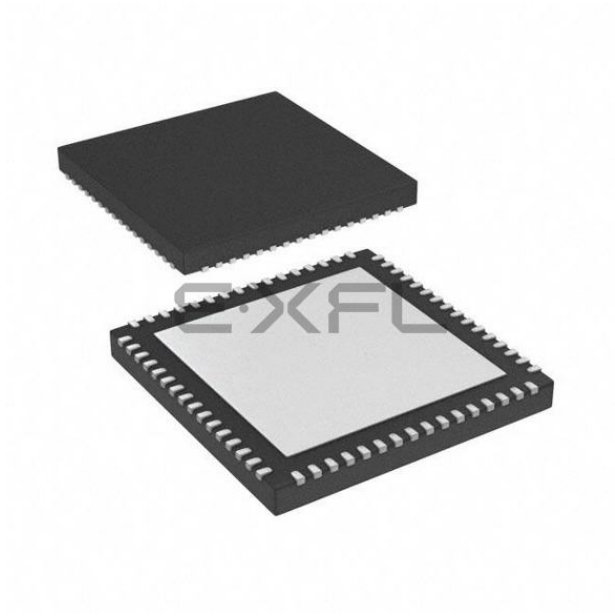
Microchip Technology - DSPIC33CK128MP506T-I/MR Datasheet



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
Speed	100MHz
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	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 20x12b; D/A 3x12b
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/dspic33ck128mp506t-i-mr

FIGURE 2-6: OFF-LINE UPS

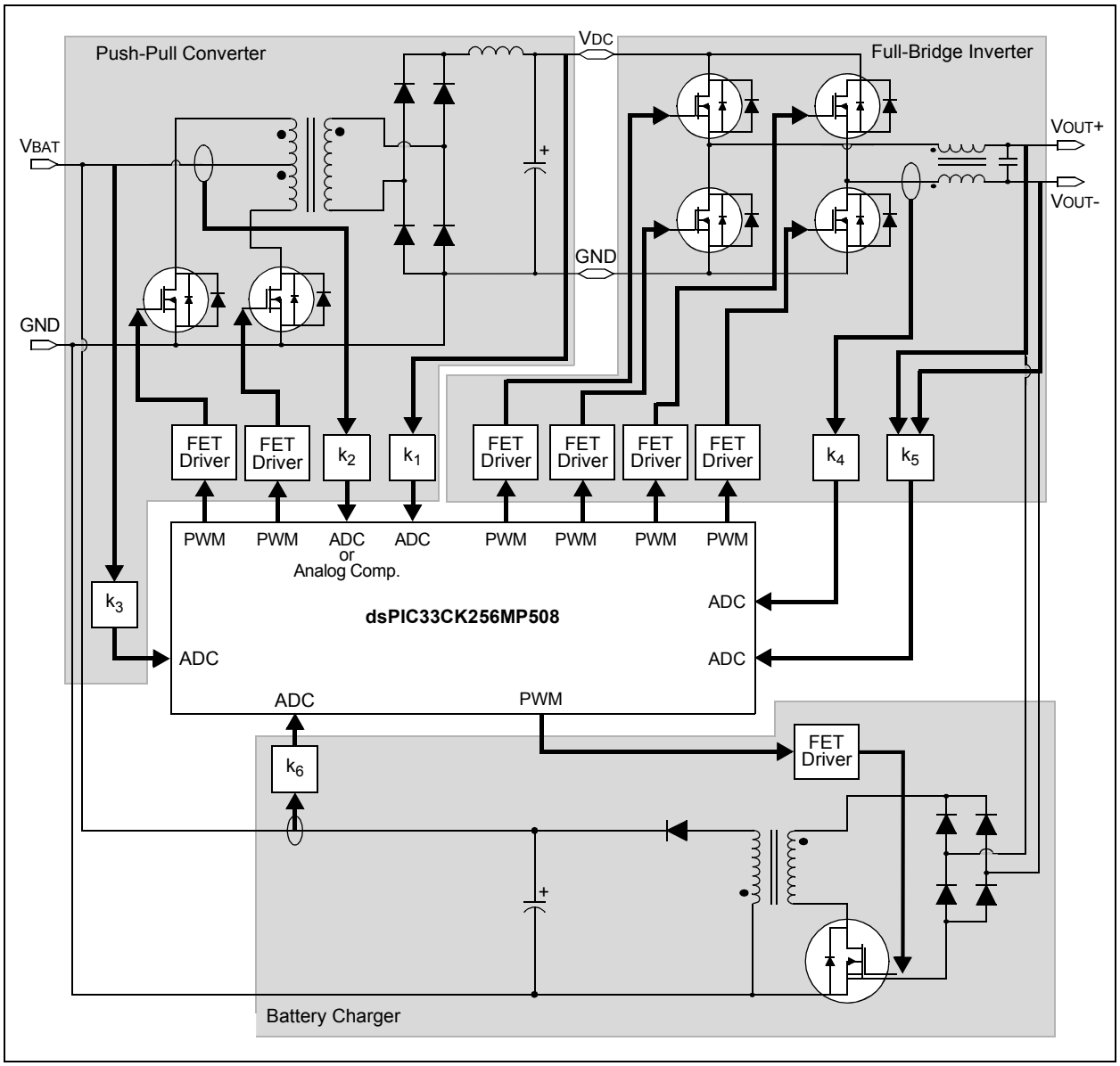
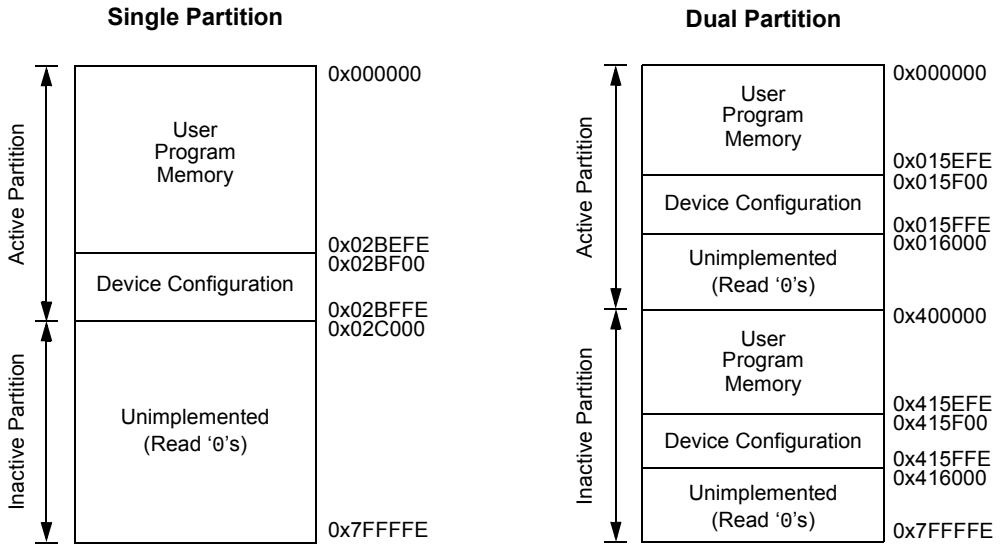
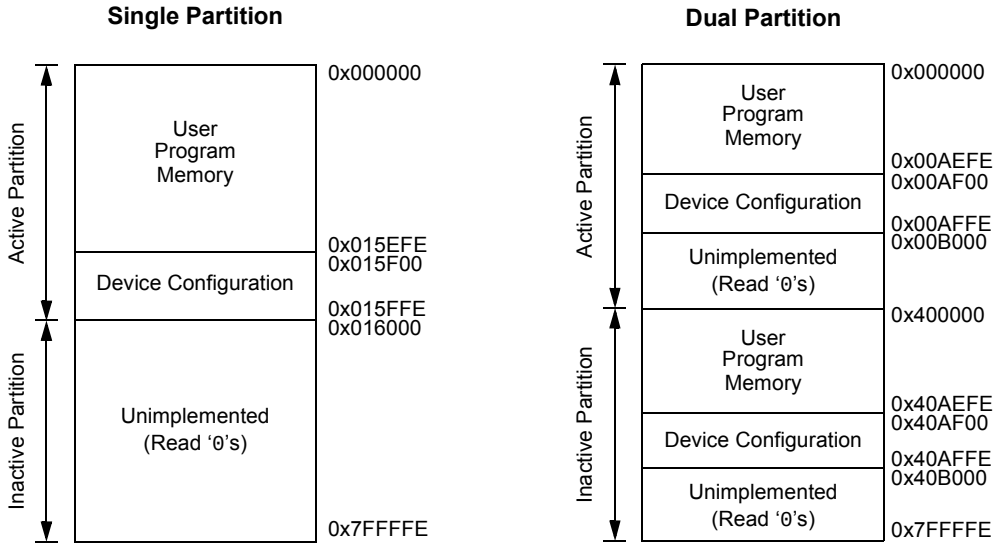


FIGURE 4-2: PROGRAM MEMORY MAP FOR dsPIC33CK256MP50/20X DEVICES(1)



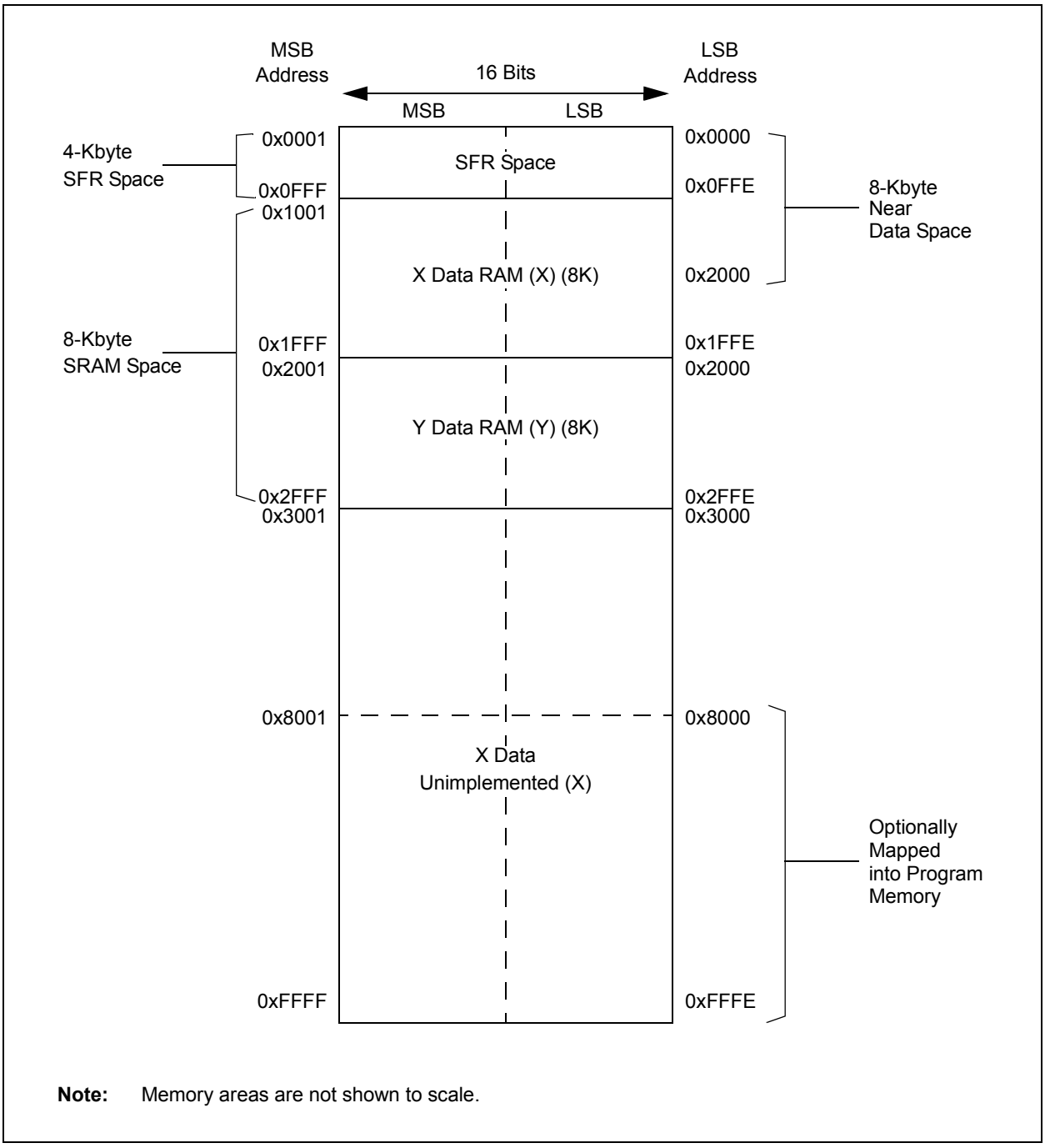
Note 1: Memory areas are not shown to scale.

FIGURE 4-3: PROGRAM MEMORY MAP FOR dsPIC33CK128MP50X/20X DEVICES(1)



Note 1: Memory areas are not shown to scale.

FIGURE 4-9: DATA MEMORY MAP FOR dsPIC33CK64MPX0X AND dsPIC33CK32MPX0X DEVICES



Note: Memory areas are not shown to scale.

5.3.2 ERROR CORRECTING CODE (ECC)

In order to improve program memory performance and durability, these devices include Error Correcting Code (ECC) functionality as an integral part of the Flash memory controller. ECC can determine the presence of single bit errors in program data, including which bit is in error, and correct the data automatically without user intervention. ECC cannot be disabled.

When data is written to program memory, ECC generates a 7-bit Hamming code parity value for every two (24-bit) instruction words. The data is stored in blocks of 48 data bits and 7 parity bits; parity data is not memory-mapped and is inaccessible. When the data is read back, the ECC calculates the parity on it and compares it to the previously stored parity value. If a parity mismatch occurs, there are two possible outcomes:

- Single bit error has occurred and has been automatically corrected on readback.
- Double-bit error has occurred and the read data is not changed.

Single bit error occurrence can be identified by the state of the ECCSBEIF (IFS0<13>) bit. An interrupt can be generated when the corresponding interrupt enable bit is set, ECCSBEIE (IEC0<13>). The ECCSTATL register contains the parity information for single bit errors. The SECOUT<7:0> bits field contains the expected calculated SEC parity and the SECIN<7:0> bits contain the actual value from a Flash read operation. The SECSYNDx bits (ECCSTATH<7:0>) indicate the bit position of the single bit error within the 48-bit pair of instruction words. When no error is present, SECINx equals SECOUTx and SECSYNDx is zero.

Double-bit errors result in a generic hard trap. The ECCDBE bit (INTCON4<1>) will be set to identify the source of the hard trap. If no Interrupt Service Routine is implemented for the hard trap, a device Reset will also occur. The ECCSTATH register contains double-bit error status information. The DEDOUT bit is the expected calculated DED parity and DEDIN is the actual value from a Flash read operation. When no error is present, DEDIN equals DEDOUT.

5.3.3 ECC FAULT INJECTION

To test Fault handling, an EEC error can be generated. Both single and double-bit errors can be generated in both the read and write data paths. Read path Fault injection first reads the Flash data and then modifies it prior to entering the ECC logic. Write path Fault injection modifies the actual data prior to it being written into the target Flash and will cause an EEC error on a subsequent Flash read. The following procedure is used to inject a Fault:

1. Load the Flash target address into the ECCADDR register.
2. Select 1st Fault bit determined by FLT1PTRx (ECCCONH<7:0>). The target bit is inverted to create the Fault.
3. If a double Fault is desired, select the 2nd Fault bit determined by FLT2PTRx (ECCCONH<15:8>), otherwise set to all ‘1’s.
4. Write the NVMKEY unlock sequence (see **Section 5.5.3 “Program Flash Memory Control Registers”**).
5. Enable the ECC Fault injection logic by setting the FLTINJ bit (ECCCONL<0>).
6. Perform a read or write to the Flash target address.

6. The Peripheral Pin Select (PPS) pin mapping rules are as follows:
   a) Only one “output” function can be active on a given pin at any time, regardless if it is a dedicated or remappable function (one pin, one output).
   b) It is possible to assign a “remappable output” function to multiple pins and externally short or tie them together for increased current drive.
   c) If any “dedicated output” function is enabled on a pin, it will take precedence over any remappable “output” function.
   d) If any “dedicated digital” (input or output) function is enabled on a pin, any number of “input” remappable functions can be mapped to the same pin.
   e) If any “dedicated analog” function(s) are enabled on a given pin, “digital input(s)” of any kind will all be disabled, although a single “digital output”, at the user’s cautionary discretion, can be enabled and active as long as there is no signal contention with an external analog input signal. For example, it is possible for the ADC to convert the digital output logic level, or to toggle a digital output on a comparator or ADC input, provided there is no external analog input, such as for a built-in self-test.
   f) Any number of “input” remappable functions can be mapped to the same pin(s) at the same time, including to any pin with a single output from either a dedicated or remappable “output”.
   g) The TRISx registers control *only* the digital I/O output buffer. Any other dedicated or remappable active “output” will automatically override the TRISx setting. The TRISx register *does not* control the digital logic “input” buffer. Remappable digital “inputs” do not automatically override TRISx settings, which means that the TRISx bit must be set to input for pins with only remappable input function(s) assigned.
   h) All analog pins are enabled by default after any Reset and the corresponding digital input buffer on the pin has been disabled. Only the Analog Select for PORTx (ANSELx) registers control the digital input buffer, *not* the TRISx register. The user must disable the analog function on a pin using the Analog Select for PORTx registers in order to use any “digital input(s)” on a corresponding pin, no exceptions.

8.5.9 I/O PORTS RESOURCES

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page contains the latest updates and additional information.

8.5.9.1 Key Resources

- **“I/O Ports with Edge Detect”** (DS70005322) in the *“dsPIC33/PIC24 Family Reference Manual”*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *“dsPIC33/PIC24 Family Reference Manual”* Sections
- Development Tools

REGISTER 8-60: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP45R5	RP45R4	RP45R3	RP45R2	RP45R1	RP45R0
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
—	—	RP44R5	RP44R4	RP44R3	RP44R2	RP44R1	RP44R0
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 **RP45R<5:0>:** Peripheral Output Function is Assigned to RP45 Output Pin bits (see Table 8-7 for peripheral function numbers)

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

bit 5-0 **RP44R<5:0>:** Peripheral Output Function is Assigned to RP44 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-61: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP47R5	RP47R4	RP47R3	RP47R2	RP47R1	RP47R0
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
—	—	RP46R5	RP46R4	RP46R3	RP46R2	RP46R1	RP46R0
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 **RP47R<5:0>:** Peripheral Output Function is Assigned to RP47 Output Pin bits (see Table 8-7 for peripheral function numbers)

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

bit 5-0 **RP46R<5:0>:** Peripheral Output Function is Assigned to RP46 Output Pin bits (see Table 8-7 for peripheral function numbers)

Example 9-3 illustrates code for using the PLL (50 MIPS) with the Primary Oscillator.

EXAMPLE 9-3: CODE EXAMPLE FOR USING PLL (50 MIPS) WITH PRIMARY OSCILLATOR (POSC)

```
//code example for 50 MIPS system clock using POSC with 10 MHz external crystal

// Select FRC on POR
#pragma config  FNOSC = FRC           // Oscillator Source Selection (Internal Fast RC (FRC))
#pragma config IESO = OFF
/// Enable Clock Switching and Configure POSC in XT mode
#pragma config POSCMD = XT
#pragma config FCKSM = CSECMD

int     main()
{
       // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
       CLKDIVbits.PLLPRE = 1;         // N1=1
       PLLFBDbits.PLLFBDIV = 100;     // M = 100
       PLLDIVbits.POST1DIV = 5;       // N2=5
       PLLDIVbits.POST2DIV = 1;       // N3=1

       // Initiate Clock Switch to Primary Oscillator with PLL (NOSC=0b011)
       __builtin_write_OSCCONH(0x03);
       __builtin_write_OSCCONL(OSCCON | 0x01);

       // Wait for Clock switch to occur
       while (OSCCONbits.OSWEN!= 0);

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

REGISTER 9-4: OSCTUN: FRC OSCILLATOR TUNING REGISTER

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/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN<5:0>					
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 15-6 **Unimplemented:** Read as '0'

bit 5-0 **TUN<5:0>:** FRC Oscillator Tuning bits

011111 = Maximum frequency deviation of 1.74% (MHz)
011110 = Center frequency + 1.693% (MHz)
...
000001 = Center frequency + 0.047% (MHz)
000000 = Center frequency (8.00 MHz nominal)
111111 = Center frequency – 0.047% (MHz)
...
100001 = Center frequency – 1.693% (MHz)
100000 = Minimum frequency deviation of -1.74% (MHz)

FIGURE 10-1: DMA FUNCTIONAL BLOCK DIAGRAM

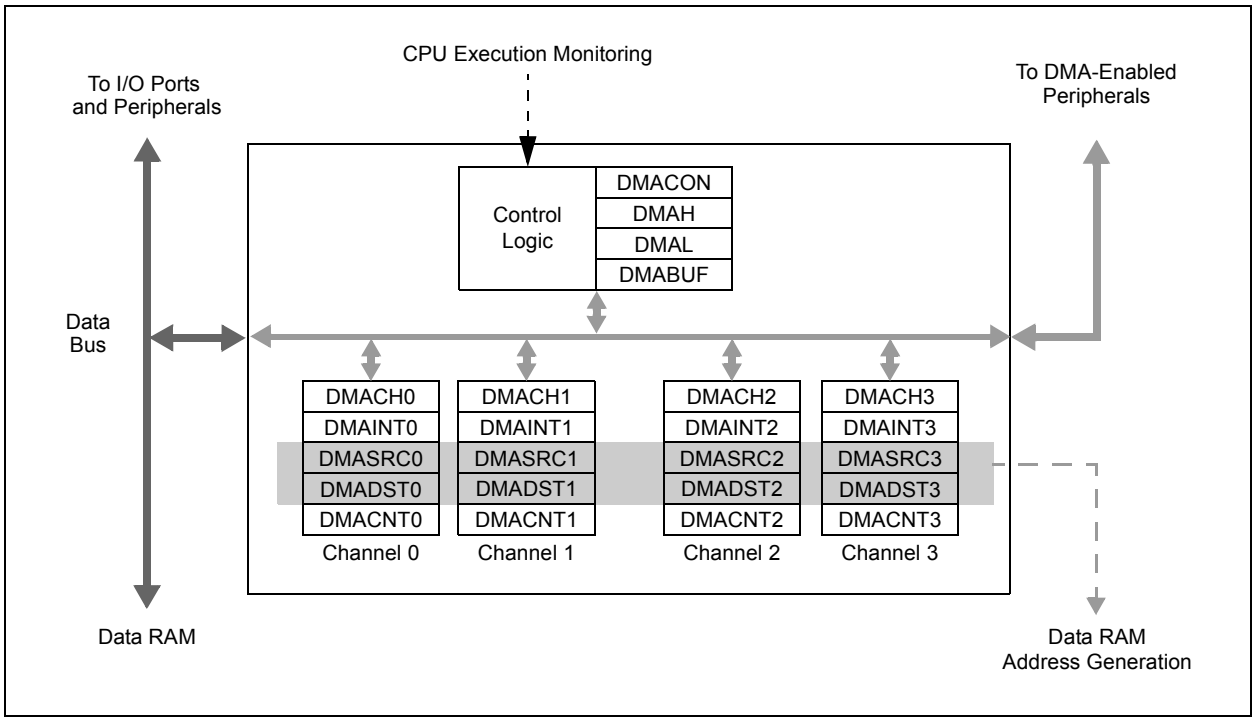
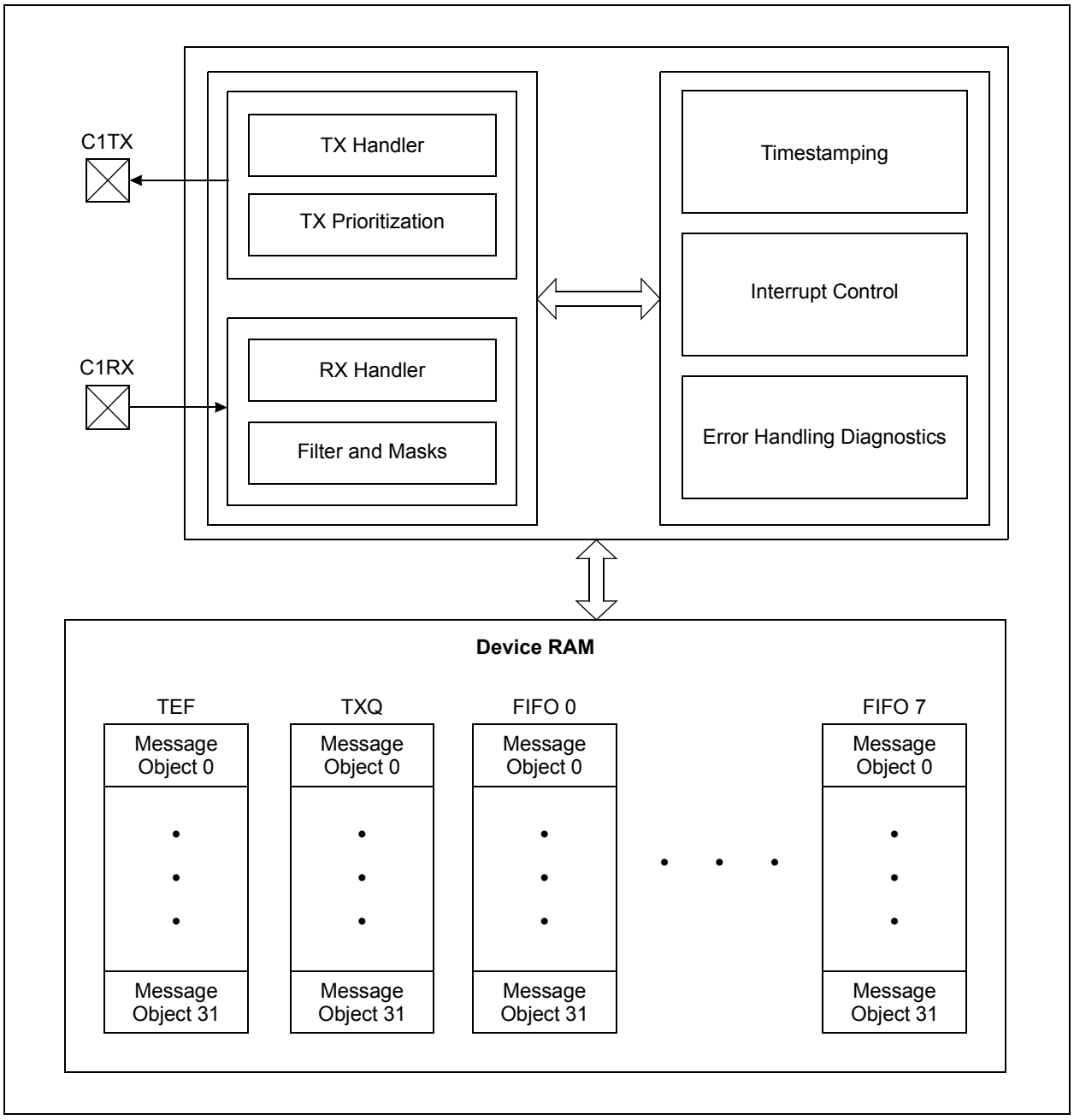


FIGURE 11-1: CAN FD MODULE BLOCK DIAGRAM



REGISTER 11-21: C1TXIFH: CAN TRANSMIT INTERRUPT STATUS REGISTER HIGH(1)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TFIF<31:24>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TFIF<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 **TFIF<31:16>:** Unimplemented

Note 1: C1TXIFH: FIFO: TFIFx = 'or' of the enabled TX FIFO flags (flags need to be cleared in the FIFO register).

REGISTER 11-22: C1TXIFL: CAN TRANSMIT INTERRUPT STATUS REGISTER LOW(1)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TFIF<15:8>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TFIF<7:0>(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 15-8 **TFIF<15:8>:** Unimplemented

bit 7-0 **TFIF<7:0>:** Transmit FIFO/TXQ Interrupt Pending bits(2)

1 = One or more enabled transmit FIFO/TXQ interrupts are pending

0 = No enabled transmit FIFO/TXQ interrupts are pending

Note 1: C1TXIFL: FIFO: TFIFx = 'or' of the enabled TX FIFO flags (flags need to be cleared in the FIFO register).

2: TFIF0 is for the transmit queue.

13.3 Control Registers

REGISTER 13-1: ADCON1L: ADC CONTROL REGISTER 1 LOW

R/W-0	U-0	R/W-0	U-0	r-0	U-0	U-0	U-0
ADON(1)	—	ADSIDL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 **ADON:** ADC Enable bit(1)
1 = ADC module is enabled
0 = ADC module is off

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

bit 13 **ADSIDL:** ADC Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

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

bit 11 **Reserved:** Maintain as '0'

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

Note 1: Set the ADON bit only after the ADC module has been configured. Changing ADC Configuration bits when ADON = 1 will result in unpredictable behavior.

16.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note 1: This data sheet summarizes the features of the dsPIC33CK256MP508 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Multiprotocol Universal Asynchronous Receiver Transmitter (UART) Module"** (DS70005288) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The Universal Asynchronous Receiver Transmitter (UART) is a flexible serial communication peripheral used to interface dsPIC® microcontrollers with other equipment, including computers and peripherals. The UART is a full-duplex, asynchronous communication channel that can be used to implement protocols, such as RS-232 and RS-485. The UART also supports the following hardware extensions:

- LIN/J2602
- IrDA®
- Direct Matrix Architecture (DMX)
- Smart Card

The primary features of the UART are:

- Full or Half-Duplex Operation
- Up to 8-Deep TX and RX First In, First Out (FIFO) Buffers
- 8-Bit or 9-Bit Data Width
- Configurable Stop Bit Length
- Flow Control
- Auto-Baud Calibration
- Parity, Framing and Buffer Overrun Error Detection
- Address Detect
- Break Transmission
- Transmit and Receive Polarity Control
- Manchester Encoder/Decoder
- Operation in Sleep mode
- Wake from Sleep on Sync Break Received Interrupt

18.0 INTER-INTEGRATED CIRCUIT (I^2C)

Note 1: This data sheet summarizes the features of the dsPIC33CK256MP508 family of devices. It is not intended to be a comprehensive reference source. For more information, refer to **"Inter-Integrated Circuit (I^2C)"** (DS70000195) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The Inter-Integrated Circuit (I^2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

The I^2C module supports these features:

- Independent Master and Slave Logic
- 7-Bit and 10-Bit Device Addresses
- General Call Address as Defined in the I^2C Protocol
- Clock Stretching to Provide Delays for the Processor to Respond to a Slave Data Request
- Both 100 kHz and 400 kHz Bus Specifications
- Configurable Address Masking
- Multi-Master modes to Prevent Loss of Messages in Arbitration
- Bus Repeater mode, Allowing the Acceptance of All Messages as a Slave, regardless of the Address
- Automatic SCL

A block diagram of the module is shown in Figure 18-1.

18.1 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communication protocol for the device being communicated with. Typically, the sequence of events is as follows:

1. Assert a Start condition on SDAx and SCLx.
2. Send the I^2C device address byte to the Slave with a write indication.
3. Wait for and verify an Acknowledge from the Slave.
4. Send the first data byte (sometimes known as the command) to the Slave.
5. Wait for and verify an Acknowledge from the Slave.
6. Send the serial memory address low byte to the Slave.
7. Repeat Steps 4 and 5 until all data bytes are sent.
8. Assert a Repeated Start condition on SDAx and SCLx.
9. Send the device address byte to the Slave with a read indication.
10. Wait for and verify an Acknowledge from the Slave.
11. Enable Master reception to receive serial memory data.
12. Generate an ACK or NACK condition at the end of a received byte of data.
13. Generate a Stop condition on SDAx and SCLx.

REGISTER 22-1: CCPxCON1L: CCPx CONTROL 1 LOW REGISTERS (CONTINUED)

bit 3-0 **MOD<3:0>:** CCPx Mode Select bits

For CCSEL = 1 (Input Capture modes):

1xxx = Reserved
011x = Reserved
0101 = Capture every 16th rising edge
0100 = Capture every 4th rising edge
0011 = Capture every rising and falling edge
0010 = Capture every falling edge
0001 = Capture every rising edge
0000 = Capture every rising and falling edge (Edge Detect mode)

For CCSEL = 0 (Output Compare/Timer modes):

1111 = External Input mode: Pulse generator is disabled, source is selected by ICS<2:0>
1110 = Reserved
110x = Reserved
10xx = Reserved
0111 = Reserved
0110 = Reserved
0101 = Dual Edge Compare mode, buffered
0100 = Dual Edge Compare mode
0011 = 16-Bit/32-Bit Single Edge mode, toggles output on compare match
0010 = 16-Bit/32-Bit Single Edge mode, drives output low on compare match
0001 = 16-Bit/32-Bit Single Edge mode, drives output high on compare match
0000 = 16-Bit/32-Bit Timer mode, output functions are disabled

REGISTER 23-4: CLCxGLSL: CLCx GATE LOGIC INPUT SELECT LOW REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N
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
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
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 **G2D4T:** Gate 2 Data Source 4 True Enable bit
1 = Data Source 4 signal is enabled for Gate 2
0 = Data Source 4 signal is disabled for Gate 2

bit 14 **G2D4N:** Gate 2 Data Source 4 Negated Enable bit
1 = Data Source 4 inverted signal is enabled for Gate 2
0 = Data Source 4 inverted signal is disabled for Gate 2

bit 13 **G2D3T:** Gate 2 Data Source 3 True Enable bit
1 = Data Source 3 signal is enabled for Gate 2
0 = Data Source 3 signal is disabled for Gate 2

bit 12 **G2D3N:** Gate 2 Data Source 3 Negated Enable bit
1 = Data Source 3 inverted signal is enabled for Gate 2
0 = Data Source 3 inverted signal is disabled for Gate 2

bit 11 **G2D2T:** Gate 2 Data Source 2 True Enable bit
1 = Data Source 2 signal is enabled for Gate 2
0 = Data Source 2 signal is disabled for Gate 2

bit 10 **G2D2N:** Gate 2 Data Source 2 Negated Enable bit
1 = Data Source 2 inverted signal is enabled for Gate 2
0 = Data Source 2 inverted signal is disabled for Gate 2

bit 9 **G2D1T:** Gate 2 Data Source 1 True Enable bit
1 = Data Source 1 signal is enabled for Gate 2
0 = Data Source 1 signal is disabled for Gate 2

bit 8 **G2D1N:** Gate 2 Data Source 1 Negated Enable bit
1 = Data Source 1 inverted signal is enabled for Gate 2
0 = Data Source 1 inverted signal is disabled for Gate 2

bit 7 **G1D4T:** Gate 1 Data Source 4 True Enable bit
1 = Data Source 4 signal is enabled for Gate 1
0 = Data Source 4 signal is disabled for Gate 1

bit 6 **G1D4N:** Gate 1 Data Source 4 Negated Enable bit
1 = Data Source 4 inverted signal is enabled for Gate 1
0 = Data Source 4 inverted signal is disabled for Gate 1

bit 5 **G1D3T:** Gate 1 Data Source 3 True Enable bit
1 = Data Source 3 signal is enabled for Gate 1
0 = Data Source 3 signal is disabled for Gate 1

bit 4 **G1D3N:** Gate 1 Data Source 3 Negated Enable bit
1 = Data Source 3 inverted signal is enabled for Gate 1
0 = Data Source 3 inverted signal is disabled for Gate 1

TABLE 24-2: PTG COMMAND OPTIONS

bit 3-0	Step Command	OPTION<3:0>	Command Description
	PTGCTRL(1)	0000	NOP.
		0001	Reserved; do not use.
		0010	Disable Step delay timer (PTGSD).
		0011	Reserved; do not use.
		0100	Reserved; do not use.
		0101	Reserved; do not use.
		0110	Enable Step delay timer (PTGSD).
		0111	Reserved; do not use.
		1000	Start and wait for the PTG Timer0 to match the PTGT0LIM register.
		1001	Start and wait for the PTG Timer1 to match the PTGT1LIM register.
		1010	Wait for the software trigger (level, PTGSWT = 1).
		1011	Wait for the software trigger (positive edge, PTGSWT = 0 to 1).
		1100	Copy the PTGC0LIM register contents to the strobe output.
		1101	Copy the PTGC1LIM register contents to the strobe output.
		1110	Copy the PTGL0 register contents to the strobe output.
		1111	Generate the triggers indicated in the PTGBTE register.
	PTGADD(1)	0000	Add the PTGADJ register contents to the PTGC0LIM register.
		0001	Add the PTGADJ register contents to the PTGC1LIM register.
		0010	Add the PTGADJ register contents to the PTGT0LIM register.
		0011	Add the PTGADJ register contents to the PTGT1LIM register.
		0100	Add the PTGADJ register contents to the PTGSDLIM register.
		0101	Add the PTGADJ register contents to the PTGL0 register.
		0110	Reserved; do not use.
		0111	Reserved; do not use.
	PTGCOPY(1)	1000	Copy the PTGHOLD register contents to the PTGC0LIM register.
		1001	Copy the PTGHOLD register contents to the PTGC1LIM register.
		1010	Copy the PTGHOLD register contents to the PTGT0LIM register.
		1011	Copy the PTGHOLD register contents to the PTGT1LIM register.
		1100	Copy the PTGHOLD register contents to the PTGSDLIM register.
		1101	Copy the PTGHOLD register contents to the PTGL0 register.
		1110	Reserved; do not use.
		1111	Reserved; do not use.

Note 1: All reserved commands or options will execute, but they do not have any affect (i.e., execute as a NOP instruction).

REGISTER 28-7: DMTPSCNTL: DMT POST-CONFIGURE COUNT STATUS REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSCNT<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
PSCNT<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 **PSCNT<15:0>:** Lower DMT Instruction Count Value Configuration Status bits
This is always the value of the FDMTCNTL Configuration register.

REGISTER 28-8: DMTPSCNTH: DMT POST-CONFIGURE COUNT STATUS 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
PSCNT<31:24>							
bit 15							bit 8

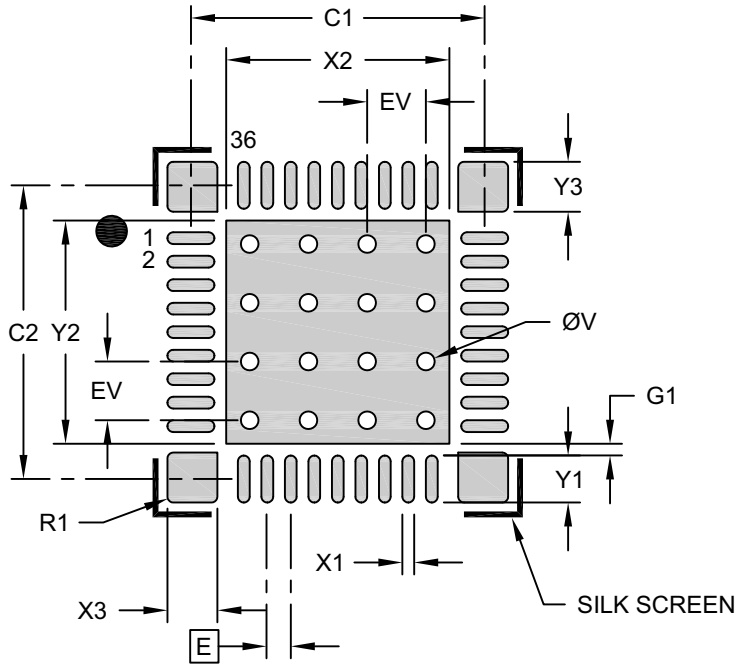
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSCNT<23:16>							
bit 7							bit 0

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 **PSCNT<31:16>:** Higher DMT Instruction Count Value Configuration Status bits
This is always the value of the FDMTCNTH Configuration register.

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	X2			3.80
Optional Center Pad Length	Y2			3.80
Contact Pad Spacing	C1		5.00	
Contact Pad Spacing	C2		5.00	
Contact Pad Width (X36)	X1			0.20
Contact Pad Length (X36)	Y1			0.80
Corner Pad Width (X4)	X3			0.20
Corner Pad Length (X36)	Y3			0.85
Corner Pad Radius	R1		0.10	
Contact Pad to Center Pad (X36)	G1	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2436A–M5

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support