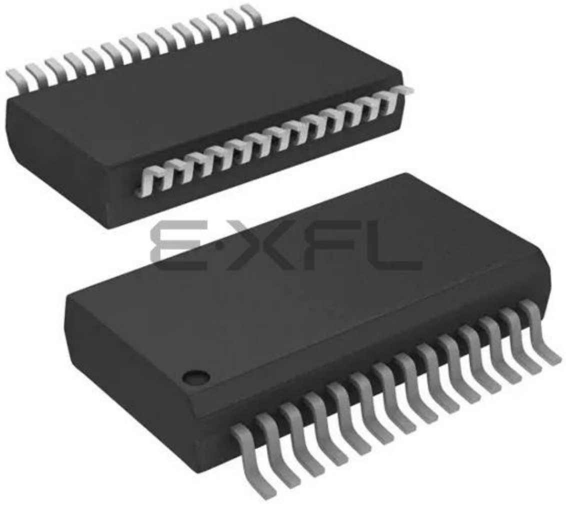
# Microchip Technology - DSPIC33CK128MP502T-I/SS Datasheet



**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 | 21 |
| 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 12x12b; D/A 3x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 28-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/dspic33ck128mp502t-i-ss |

## 1.0 DEVICE OVERVIEW

**Note 1:** This data sheet summarizes the features of the dsPIC33CK256MP508 family of devices. It is not intended to be a comprehensive resource. 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).

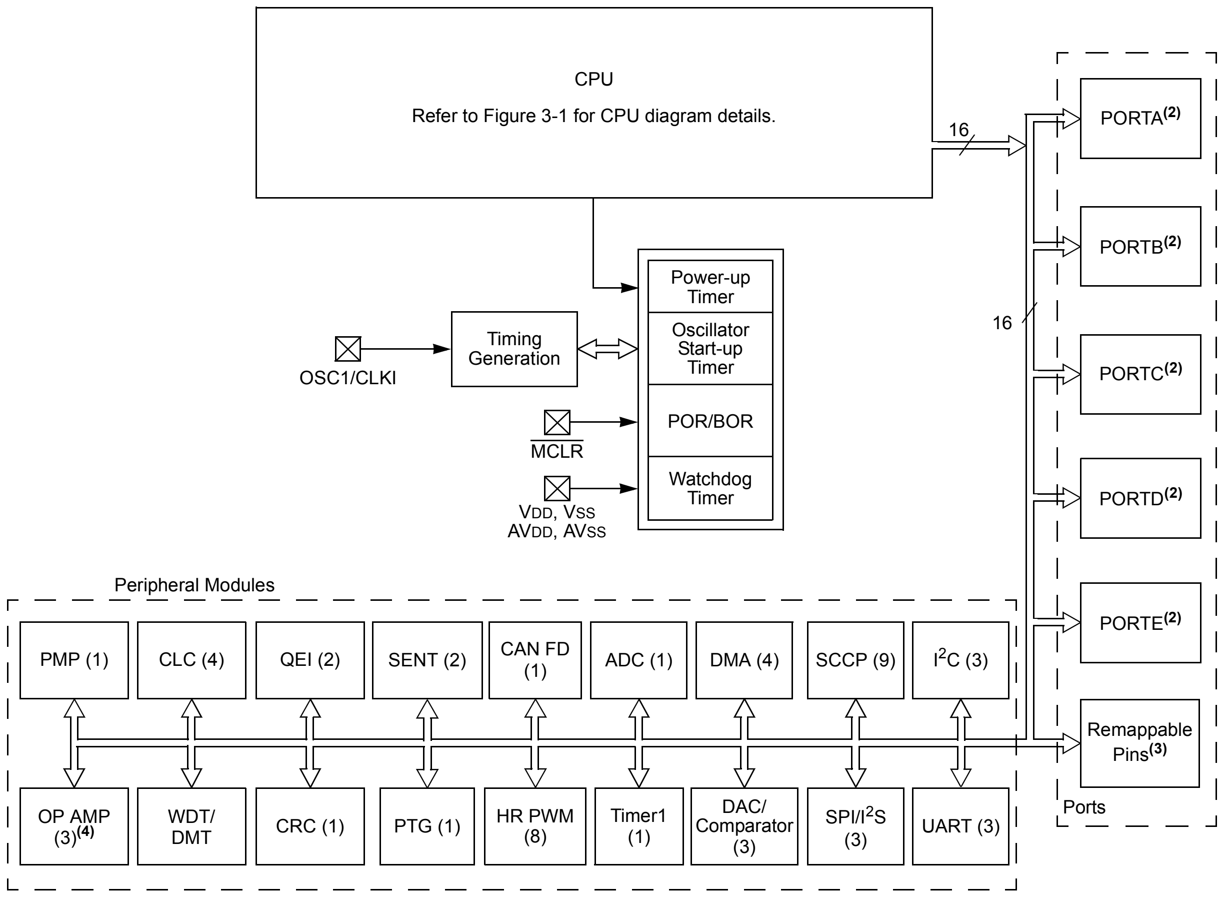
**2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

This document contains device-specific information for the dsPIC33CK256MP508 Digital Signal Controller (DSC) and Microcontroller (MCU) devices.

dsPIC33CK256MP508 devices contain extensive Digital Signal Processor (DSP) functionality with a high-performance, 16-bit MCU architecture.

Figure 1-1 shows a general block diagram of the core and peripheral modules of the dsPIC33CK256MP508 family. Table 1-1 lists the functions of the various pins shown in the pinout diagrams.

**FIGURE 1-1: dsPIC33CK256MP508 FAMILY BLOCK DIAGRAM**(1)



**Note 1:** The numbers in the parentheses are the number of instantiations of the module indicated.

**2:** Not all I/O pins or features are implemented on all device pinout configurations.

**3:** Some peripheral I/Os are only accessible through Peripheral Pin Select (PPS).

**4:** 28-lead devices have only two op amp instances.

### TABLE 4-16: SFR BLOCK F00h

| Register | Address | All Resets | Register | Address | All Resets | Register | Address | All Resets |
|---|---|---|---|---|---|---|---|---|
| **UART3** | | | I2C3TRN | F70 | `--------11111111` | WDTCONH | FB6 | `0000000000000000` |
| U3MODE | F00 | `--000-0000000000` | I2C3RCV | F74 | `--------00000000` | **REFO** | | |
| U3MODEH | F02 | `00---00000000000` | **Reset and Oscillator** | | | REFOCONL | FB8 | `--000-00----0000` |
| U3STA | F04 | `0000000010000000` | RCON | F80 | `xx----x01x0xxxxx` | REFOCONH | FBA | `0000000000000000` |
| U3STAH | F06 | `0000-00000101110` | OSCCON | F84 | `0000-yyy0-0-0--0` | REFOTRIML | FBC | `000000000-------` |
| U3BRG | F08 | `0000000000000000` | CLKDIV | F86 | `00110000--000001` | **Processor** | | |
| U3BRGH | F0A | `------------0000` | PLLFBD | F88 | `----000010010110` | PCTRAPL | FC0 | `xxxxxxxxxxxxxxxx` |
| U3RXREG | F0C | `--------xxxxxxxx` | PLLDIV | F8A | `------00-001-001` | PCTRAPH | FC2 | `--------xxxxxxxx` |
| U3TXREG | F10 | `--------xxxxxxxx` | OSCTUN | F8C | `----------000000` | FEXL | FC4 | `xxxxxxxxxxxxxxxx` |
| U3P1 | F14 | `-------000000000` | ACLKCON1 | F8E | `00-----0--000001` | FEXH | FC6 | `--------xxxxxxxx` |
| U3P2 | F16 | `-------000000000` | APLLFBD1 | F90 | `----000010010110` | FEX2L | FC8 | `xxxxxxxxxxxxxxxx` |
| U3P3 | F18 | `0000000000000000` | APLLDIV1 | F92 | `------00-001-001` | FEX2H | FCA | `--------xxxxxxxx` |
| U3P3H | F1A | `--------00000000` | CANCLKCON | F9A | `0---0000-0000000` | VISI | FCC | `xxxxxxxxxxxxxxxx` |
| U3TXCHK | F1C | `--------00000000` | DCOTUN | F9C | `--000000--000000` | DPCL | FCE | `xxxxxxxxxxxxxxxx` |
| U3RXCHK | F1E | `--------00000000` | DCOCON | F9E | `--0-xxxx--------` | DPCH | FD0 | `--------xxxxxxxx` |
| U3SCCON | F20 | `----------00000-` | **PMD** | | | APPO | FD2 | `xxxxxxxxxxxxxxxx` |
| U3SCINT | F22 | `--00-000--00-000` | PMD1 | FA4 | `----000-00000-00` | APPI | FD4 | `xxxxxxxxxxxxxxxx` |
| U3INT | F24 | `--------00---0--` | PMD2 | FA6 | `-------000000000` | APPS | FD6 | `-----------xxxxx` |
| **I2C3** | | | PMD3 | FA8 | `-------00-0-000-` | STROUTL | FD8 | `xxxxxxxxxxxxxxxx` |
| I2C3CONL | F5C | `--01000000000000` | PMD4 | FAA | `------------0---` | STROUTH | FDA | `xxxxxxxxxxxxxxxx` |
| I2C3CONH | F5E | `---------0000000` | PMD6 | FAE | `----0000--------` | STROVCNT | FDC | `xxxxxxxxxxxxxxxx` |
| I2C3 STAT | F60 | `000--00000000000` | PMD7 | FB0 | `-----000----0---` | JDATAL | FFA | `0000000000000000` |
| I2C3ADD | F64 | `------0000000000` | PMD8 | FB2 | `--000--0--00000-` | JDATAH | FFC | `0000000000000000` |
| I2C3MSK | F68 | `------0000000000` | **WDT** | | | | | |
| I2C3BRG | F6C | `0000000000000000` | WDTCONL | FB4 | `---0000000000000` | | | |

**Legend:** x = unknown or indeterminate value; "-" =unimplemented bits; y = value set by Configuration bits. Address values are in hexadecimal. Reset values are in binary.

## 6.1 Reset 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.

### 6.1.1 KEY RESOURCES

- **"Reset"** (DS70602) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

### TABLE 7-1: INTERRUPT VECTOR DETAILS (CONTINUED)

| Interrupt Source | Vector # | IRQ # | IVT Address | Interrupt Bit Location | | |
|---|---|---|---|---|---|---|
| | | | | Flag | Enable | Priority |
| QEI1 – QEI Position Counter Compare | 56 | 48 | 0x000074 | IFS3<0> | IEC3<0> | IPC12<2:0> |
| U1E – UART1 Error | 57 | 49 | 0x000076 | IFS3<1> | IEC3<1> | IPC12<6:4> |
| U2E – UART2 Error | 58 | 50 | 0x000078 | IFS3<2> | IEC3<2> | IPC12<10:8> |
| CRC – CRC Generator | 59 | 51 | 0x00007A | IFS3<3> | IEC3<3> | IPC12<14:12> |
| C1TX – CAN1 TX Data Request | 60 | 52 | 0x00007C | IFS3<4> | IEC3<4> | IPC13<2:0> |
| Reserved | 61 | 53 | 0x00007E | — | — | — |
| QEI2 – QEI Position Counter Compare | 62 | 54 | 0x000080 | IFS3<6> | IEC3<6> | IPC13<10:8> |
| Reserved | 63 | 55 | 0x000082 | — | — | — |
| U3E – UART3 Error | 64 | 56 | 0x000084 | IFS3<8> | IEC3<8> | IPC14<2:0> |
| U3RX – UART3 Receiver | 65 | 57 | 0x000086 | IFS3<9> | IEC3<9> | IPC14<6:4> |
| U3TX – UART3 Transmitter | 66 | 58 | 0x000088 | IFS3<10> | IEC3<10> | IPC14<10:8> |
| SPI3RX – SPI3 Receiver | 67 | 59 | 0x00008A | IFS3<11> | IEC3<11> | IPC14<14:12> |
| SPI3TX – SPI3 Transmitter | 68 | 60 | 0x00008C | IFS3<12> | IEC3<12> | IPC15<2:0> |
| ICD – In-Circuit Debugger | 69 | 61 | 0x00008E | IFS3<13> | IEC3<13> | IPC15<6:4> |
| JTAG – JTAG Programming | 70 | 62 | 0x000090 | IFS3<14> | IEC3<14> | IPC15<10:8> |
| PTGSTEP – PTG Step | 71 | 63 | 0x000092 | IFS3<15> | IEC3<15> | IPC15<14:12> |
| I2C1BC – I2C1 Bus Collision | 72 | 64 | 0x000094 | IFS4<0> | IEC4<0> | IPC16<2:0> |
| I2C2BC – I2C2 Bus Collision | 73 | 65 | 0x000096 | IFS4<1> | IEC4<1> | IPC16<6:4> |
| Reserved | 74 | 66 | 0x000098 | — | — | — |
| PWM1 – PWM Generator 1 | 75 | 67 | 0x00009A | IFS4<3> | IEC4<3> | IPC16<14:12> |
| PWM2 – PWM Generator 2 | 76 | 68 | 0x00009C | IFS4<4> | IEC4<4> | IPC17<2:0> |
| PWM3 – PWM Generator 3 | 77 | 69 | 0x00009E | IFS4<5> | IEC4<5> | IPC17<6:4> |
| PWM4 – PWM Generator 4 | 78 | 70 | 0x0000A0 | IFS4<6> | IEC4<6> | IPC17<10:8> |
| PWM5 – PWM Generator 5 | 79 | 71 | 0x0000A2 | IFS4<7> | IEC4<7> | IPC17<14:12> |
| PWM6 – PWM Generator 6 | 80 | 72 | 0x0000A4 | IFS4<8> | IEC4<8> | IPC18<2:0> |
| PWM7 – PWM Generator 7 | 81 | 73 | 0x0000A6 | IFS4<9> | IEC4<9> | IPC18<6:4> |
| PWM8 – PWM Generator 8 | 82 | 74 | 0x0000A8 | IFS4<10> | IEC4<10> | IPC18<10:8> |
| CND – Change Notice D | 83 | 75 | 0x0000AA | IFS4<11> | IEC4<11> | IPC18<14:12> |
| CNE – Change Notice E | 84 | 76 | 0x0000AC | IFS4<12> | IEC4<12> | IPC19<2:0> |
| CMP1 – Comparator 1 | 85 | 77 | 0x0000AE | IFS4<13> | IEC4<13> | IPC19<6:4> |
| CMP2 – Comparator 2 | 86 | 78 | 0x0000B0 | IFS4<14> | IEC4<14> | IPC19<10:8> |
| CMP3 – Comparator 3 | 87 | 79 | 0x0000B2 | IFS4<15> | IEC4<15> | IPC19<14:12> |
| Reserved | 88 | 80 | 0x0000B4 | — | — | — |
| PTGWDT – PTG Watchdog Timer Time-out | 89 | 81 | 0x0000B6 | IFS5<1> | IEC5<1> | IPC20<6:4> |
| PTG0 – PTG Trigger 0 | 90 | 82 | 0x0000B8 | IFS5<2> | IEC5<2> | IPC20<10:8> |
| PTG1 – PTG Trigger 1 | 91 | 83 | 0x0000BA | IFS5<3> | IEC5<3> | IPC20<14:12> |
| PTG2 – PTG Trigger 2 | 92 | 84 | 0x0000BC | IFS5<4> | IEC5<4> | IPC21<2:0> |
| PTG3 – PTG Trigger 3 | 93 | 85 | 0x0000BE | IFS5<5> | IEC5<6> | IPC21<6:4> |
| SENT1 – SENT1 TX/RX | 94 | 86 | 0x0000C0 | IFS5<6> | IEC5<6> | IPC21<10:8> |
| SENT1E – SENT1 Error | 95 | 87 | 0x0000C2 | IFS5<7> | IEC5<7> | IPC21<14:12> |
| SENT2 – SENT2 TX/RX | 96 | 88 | 0x0000C4 | IFS5<8> | IEC5<8> | IPC22<2:0> |
| SENT2E – SENT2 Error | 97 | 89 | 0x0000C6 | IFS5<9> | IEC5<9> | IPC22<6:4> |
| ADC – ADC Global Interrupt | 98 | 90 | 0x0000C8 | IFS5<10> | IEC5<10> | IPC22<10:8> |

Equation 9-3 provides the relationship between the APLL Input Frequency (AFPLLI) and the AVCO Output Frequency (AFVCO).

### EQUATION 9-3: AFVCO CALCULATION

$$AF_{VCO} = AF_{PLLI} \times \left(\frac{M}{N1}\right) = AF_{PLLI} \times \left(\frac{APLLFBDIV<7:0>}{APLLPRE<3:0>}\right)$$

Equation 9-4 provides the relationship between the APLL Input Frequency (AFPLLI) and APLL Output Frequency (AFPLLO).

### EQUATION 9-4: AFPLLO CALCULATION

$$AF_{PLLO} = AF_{PLLI} \times \left(\frac{M}{N1 \times N2 \times N3}\right) = AF_{PLLI} \times \left(\frac{APLLFBDIV<7:0>}{APLLPRE<3:0> \times APOST1DIV<2:0> \times APOST2DIV<2:0>}\right)$$

Where:

$M$ = $APLLFBDIV<7:0>$  
$N1$ = $APLLPRE<3:0>$  
$N2$ = $APOST1DIV<2:0>$  
$N3$ = $APOST2DIV<2:0>$

### EXAMPLE 9-2: CODE EXAMPLE FOR USING AUXILIARY PLL WITH THE INTERNAL FRC OSCILLATOR

```
//code example for AFVCO = 1 GHz and AFPLLO = 500 MHz using 8 MHz internal FRC
// Configure the source clock for the APLL
ACLKCON1bits.FRCSEL = 1;              // Select internal FRC as the clock source
// Configure the APLL prescaler, APLL feedback divider, and both APLL postscalers.
ACLKCON1bits.APLLPRE = 1;             // N1 = 1
APLLFBD1bits.APLLFBDIV = 125;         // M = 125
APLLDIV1bits.APOST1DIV = 2;           // N2 = 2
APLLDIV1bits.APOST2DIV = 1;           // N3 = 1
// Enable APLL
ACLKCON1bits.APLLEN = 1;
```

**Note:** Even with the APLLEN bit set, another peripheral must generate a clock request before the APLL will start.

#### 10.1.6 CHANNEL PRIORITY

Each DMA channel functions independently of the others, but also competes with the others for access to the data and DMA buses. When access collisions occur, the DMA Controller arbitrates between the channels using a user-selectable priority scheme. Two schemes are available:

- Round Robin: When two or more channels collide, the lower numbered channel receives priority on the first collision. On subsequent collisions, the higher numbered channels each receive priority based on their channel number.
- Fixed: When two or more channels collide, the lowest numbered channel always receives priority, regardless of past history; however, any channel being actively processed is not available for an immediate retrigger. If a higher priority channel is continually requesting service, it will be scheduled for service after the next lower priority channel with a pending request.

## 10.2 Typical Setup

To set up a DMA channel for a basic data transfer:

1. Enable the DMA Controller (DMAEN = 1) and select an appropriate channel priority scheme by setting or clearing PRSSEL.
2. Program DMAH and DMAL with appropriate upper and lower address boundaries for data RAM operations.
3. Select the DMA channel to be used and disable its operation (CHEN = 0).
4. Program the appropriate source and destination addresses for the transaction into the channel's DMASRCn and DMADSTn registers. For PIA mode addressing, use the base address value.
5. Program the DMACNTn register for the number of triggers per transfer (One-Shot or Continuous modes) or the number of words (bytes) to be transferred (Repeated modes).
6. Set or clear the SIZE bit to select the data size.
7. Program the TRMODE<1:0> bits to select the Data Transfer mode.
8. Program the SAMODE<1:0> and DAMODE<1:0> bits to select the addressing mode.
9. Enable the DMA channel by setting CHEN.
10. Enable the trigger source interrupt.

## 10.3 Peripheral Module Disable

The channels of the DMA Controller can be individually powered down using the Peripheral Module Disable (PMD) registers.

## 10.4 Registers

The DMA Controller uses a number of registers to control its operation. The number of registers depends on the number of channels implemented for a particular device.

There are always four module-level registers (one control and three buffer/address):

- DMACON: DMA Engine Control Register (Register 10-1)
- DMAH and DMAL: DMA High and Low Address Limit Registers
- DMABUF: DMA Transfer Data Buffer

Each of the DMA channels implements five registers (two control and three buffer/address):

- DMACHn: DMA Channel n Control Register (Register 10-2)
- DMAINTn: DMA Channel n Interrupt Register (Register 10-3)
- DMASRCn: DMA Data Source Address Pointer for Channel n Register
- DMADSTn: DMA Data Destination Source for Channel n Register
- DMACNTn: DMA Transaction Counter for Channel n Register

For dsPIC33CK256MP508 devices, there are a total of 34 registers.

### REGISTER 11-11: C1TSCONH: CAN TIMESTAMP CONTROL 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 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | TSRES | TSEOF | TBCEN |
| 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 15-3 **Unimplemented:** Read as ‘0’

bit 2 **TSRES:** Timestamp Reset bit (CAN FD frames only)
1 = At sample point of the bit following the FDF bit
0 = At sample point of Start-of-Frame (SOF)

bit 1 **TSEOF:** Timestamp End-of-Frame (EOF) bit
1 = Timestamp when frame is taken valid (11898-1 10.7):
- RX no error until last, but one bit of EOF
- TX no error until the end of EOF

0 = Timestamp at “beginning” of frame:
- Classical Frame: At sample point of SOF
- FD Frame: see TSRES bit

bit 0 **TBCEN:** Time Base Counter Enable bit
1 = Enables TBC
0 = Stops and resets TBC

### REGISTER 11-12: C1TSCONL: CAN TIMESTAMP CONTROL REGISTER LOW

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | — | TBCPRE<9:8> | |
| 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 |
| TBCPRE<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 15-10 **Unimplemented:** Read as ‘0’

bit 9-0 **TBCPRE<9:0>:** CAN Time Base Counter Prescaler bits
1023 = TBC increments every 1024 clocks
...
0 = TBC increments every 1 clock

### REGISTER 11-27: C1FIFOBAH: CAN MESSAGE MEMORY BASE ADDRESS 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 |
| FIFOBA<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 |
| FIFOBA<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 **FIFOBA<31:16>:** Message Memory Base Address bits
Defines the base address for the transmit event FIFO followed by the message objects.

### REGISTER 11-28: C1FIFOBAL: CAN MESSAGE MEMORY BASE ADDRESS 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 |
| FIFOBA<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-0 | R-0 |
| FIFOBA<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 **FIFOBA<15:0>:** Message Memory Base Address bits
Defines the base address for the transmit event FIFO followed by the message objects.

### REGISTER 12-9: LOGCONy: COMBINATORIAL PWM LOGIC CONTROL REGISTER y[2]

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PWMS1y3[1] | PWMS1y2[1] | PWMS1y1[1] | PWMS1y0[1] | PWMS2y3[1] | PWMS2y2[1] | PWMS2y1[1] | PWMS2y0[1] |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| S1yPOL | S2yPOL | PWMLFy1 | PWMLFy0 | — | PWMLFyD2[3] | PWMLFyD1[3] | PWMLFyD0[3] |
| 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 **PWMS1y<3:0>:** Combinatorial PWM Logic Source #1 Selection bits[1]

1111 = PWM8L
1110 = PWM8H
1101 = PWM7L
1100 = PWM7H
1011 = PWM6L
1010 = PWM6H
1001 = PWM5L
1000 = PWM5H
0111 = PWM4L
0110 = PWM4H
0101 = PWM3L
0100 = PWM3H
0011 = PWM2L
0010 = PWM2H
0001 = PWM1L
0000 = PWM1H

bit 11-8 **PWMS2y<3:0>:** Combinatorial PWM Logic Source #2 Selection bits[1]

1111 = PWM8L
1110 = PWM8H
1101 = PWM7L
1100 = PWM7H
1011 = PWM6L
1010 = PWM6H
1001 = PWM5L
1000 = PWM5H
0111 = PWM4L
0110 = PWM4H
0101 = PWM3L
0100 = PWM3H
0011 = PWM2L
0010 = PWM2H
0001 = PWM1L
0000 = PWM1H

**Note 1:** Logic function input will be connected to ‘0’ if the PWM channel is not present.

**2:** ‘y’ denotes a common instance (A-F).

**3:** Instances of y = A, C, E of LOGCONy assign logic function output to the PWMxL pin. Instances of y = B, D, F of LOGCONy assign logic function to the PWMxH pin.

### REGISTER 14-4: DACxCONH: DACx CONTROL HIGH REGISTER

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | TMCB<9:8> | |
| 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 |
|---|---|---|---|---|---|---|---|
| TMCB<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 |

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-0 **TMCB<9:0>:** DACx Leading-Edge Blanking bits

These register bits specify the blanking period for the comparator, following changes to the DAC output during Change-of-State (COS), for the input signal selected by the HCFSEL<3:0> bits in Register 14-9.

### REGISTER 14-5: DACxCONL: DACx CONTROL LOW REGISTER

| R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| DACEN | IRQM1[1,2] | IRQM0[1,2] | — | — | CBE | DACOEN | FLTREN |
| 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 |
|---|---|---|---|---|---|---|---|
| CMPSTAT | CMPPOL | INSEL2 | INSEL1 | INSEL0 | HYSPOL | HYSSEL1 | HYSSEL0 |
| 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 |

bit 15 **DACEN:** Individual DACx Module Enable bit

1 = Enables DACx module
0 = Disables DACx module to reduce power consumption; any pending Slope mode and/or underflow conditions are cleared

bit 14-13 **IRQM<1:0>:** Interrupt Mode select bits[1,2]

11 = Generates an interrupt on either a rising or falling edge detect
10 = Generates an interrupt on a falling edge detect
01 = Generates an interrupt on a rising edge detect
00 = Interrupts are disabled

bit 12-11 **Unimplemented:** Read as ‘0’

**Note 1:** Changing these bits during operation may generate a spurious interrupt.
**2:** The edge selection is a post-polarity selection via the CMPPOL bit.

### REGISTER 14-6: DACxDATH: DACx DATA HIGH 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 |
| DACDAT<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 |
| DACDAT<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 |

bit 15-0 **DACDAT<15:0>:** DACx Data bits

This register specifies the high DACx data value. Valid values are from 0x0205 to 0x3890.

### REGISTER 14-7: DACxDATL: DACx DATA 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 |
| DACLOW<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 |
| DACLOW<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 |

bit 15-0 **DACLOW<15:0>:** DACx Low Data bits

In Hysteretic mode, Slope Generator mode and Triangle mode, this register specifies the low data value and/or limit for the DACx module. Valid values are from 0x0205 to 0x3890.

## 15.1 QEI Control and Status Registers

### REGISTER 15-1: QEIxCON: QEIx CONTROL REGISTER

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| QEIEN | — | QEISIDL | PIMOD2[(1,5)] | PIMOD1[(1,5)] | PIMOD0[(1,5)] | IMV1[(2)] | IMV0[(2)] |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | INTDIV2[(3)] | INTDIV1[(3)] | INTDIV0[(3)] | CNTPOL | GATEN | CCM1 | CCM0 |
| 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 **QEIEN:** Quadrature Encoder Interface Module Enable bit

1 = Module counters are enabled
0 = Module counters are disabled, but SFRs can be read or written

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **QEISIDL:** QEI Stop in Idle Mode bit

1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-10 **PIMOD<2:0>:** Position Counter Initialization Mode Select bits[(1,5)]

111 = Modulo Count mode for position counter and every Index event resets the position counter[(4)]
110 = Modulo Count mode for position counter
101 = Resets the position counter when the position counter equals the QEIxGEC register
100 = Second Index event after Home event initializes position counter with contents of QEIxIC register
011 = First Index event after Home event initializes position counter with contents of QEIxIC register
010 = Next Index input event initializes the position counter with contents of QEIxIC register
001 = Every Index input event resets the position counter
000 = Index input event does not affect the position counter

bit 9-8 **IMV<1:0>:** Index Match Value bits[(2)]

11 = Index match occurs when QEBx = 1 and QEAx = 1
10 = Index match occurs when QEBx = 1 and QEAx = 0
01 = Index match occurs when QEBx = 0 and QEAx = 1
00 = Index match occurs when QEBx = 0 and QEAx = 0

bit 7 **Unimplemented:** Read as ‘0’

**Note 1:** When CCMx = 10 or CCMx = 11, all of the QEI counters operate as timers and the PIMOD<2:0> bits are ignored.

**2:** When CCMx = 00, and QEAx and QEBx values match the Index Match Value (IMV), the POSxCNTH and POSxCNTL registers are reset.

**3:** The selected clock rate should be at least twice the expected maximum quadrature count rate.

**4:** Not all devices support this mode.

**5:** The QCAPEN and HCAPEN bits must be cleared during PIMODx Modes 2 through 7 to ensure proper functionality. Not all devices support HCAPEN.

### REGISTER 16-4: UxSTAH: UARTx STATUS REGISTER HIGH (CONTINUED)

bit 2 **XON:** UART in XON Mode bit

Only valid when FLO<1:0> control bits are set to XON/XOFF mode.
1 = UART has received XON
0 = UART has not received XON or XOFF was received

bit 1 **URXBE:** UART RX Buffer Empty Status bit

1 = Receive buffer is empty; writing '1' when URXEN = 0 will reset the RX FIFO Pointers and counters
0 = Receive buffer is not empty

bit 0 **URXBF:** UART RX Buffer Full Status bit

1 = Receive buffer is full
0 = Receive buffer is not full

**Note 1:** The receive watermark interrupt is not set if PERIF or FERIF is set and the corresponding IE bit is set.

### REGISTER 16-9: UxP1: UARTx TIMING PARAMETER 1 REGISTER

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
| — | — | — | — | — | — | — | P1<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 |
| P1<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 15-9 **Unimplemented:** Read as ‘0’

bit 8-0 **P1<8:0>:** Parameter 1 bits

DMX TX:
Number of Bytes to Transmit – 1 (not including Start code).

LIN Master TX:
PID to transmit (bits<5:0>).

Asynchronous TX with Address Detect:
Address to transmit. A ‘1’ is automatically inserted into bit 9 (bits<7:0>).

Smart Card Mode:
Guard Time Counter bits. This counter is operated on the bit clock whose period is always equal to one ETU (bits<8:0>).

Other Modes:
Not used.

### REGISTER 16-17: UxINT: UARTx INTERRUPT REGISTER

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

| R/W-0, HS | R/W-0, HS | U-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| WUIF | ABDIF | — | — | — | ABDIE | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | HS = Hardware Settable 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 **Unimplemented:** Read as ‘0’

bit 7 **WUIF:** Wake-up Interrupt Flag bit
1 = Sets when WAKE = 1 and RX makes a ‘1’-to-‘0’ transition; triggers event interrupt (must be cleared by software)
0 = WAKE is not enabled or WAKE is enabled, but no wake-up event has occurred

bit 6 **ABDIF:** Auto-Baud Completed Interrupt Flag bit
1 = Sets when ABD sequence makes the final ‘1’-to-‘0’ transition; triggers event interrupt (must be cleared by software)
0 = ABAUD is not enabled or ABAUD is enabled but auto-baud has not completed

bit 5-3 **Unimplemented:** Read as ‘0’

bit 2 **ABDIE:** Auto-Baud Completed Interrupt Enable Flag bit
1 = Allows ABDIF to set an event interrupt
0 = ABDIF does not set an event interrupt

bit 1-0 **Unimplemented:** Read as ‘0’

### REGISTER 19-4: PMADDR: PARALLEL MASTER PORT ADDRESS 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 |
|---|---|---|---|---|---|---|---|
| CS2(1) | CS1(1) | ADDR<13:8> | | | | | |
| ADDR15(1) | ADDR14(1) | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|
| ADDR<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 **CS2:** Chip Select 2 bit(1)

`1` = Chip Select 2 is active
`0` = Chip Select 2 is inactive (ADDR15 function is selected)

bit 15 **ADDR15:** Target Address bit 15(1)

bit 14 **CS1:** Chip Select 1 bit(1)

`1` = Chip Select 1 is active
`0` = Chip Select 1 is inactive (ADDR14 function is selected)

bit 14 **ADDR14:** Target Address bit 14(1)

bit 13-0 **ADDR<13:0>:** Target Address bits

**Note 1:** The use of these pins as PMA15/PMA14 or CS2/CS1 is selected by the CSF<1:0> bits (PMCON<7:6>).

### REGISTER 25-2: CRCCONH: CRC CONTROL REGISTER HIGH

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | DWIDTH4 | DWIDTH3 | DWIDTH2 | DWIDTH1 | DWIDTH0 |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | PLEN4 | PLEN3 | PLEN2 | PLEN1 | PLEN0 |
| 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 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **DWIDTH<4:0>:** Data Word Width Configuration bits
Configures the width of the data word (Data Word Width – 1).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PLEN<4:0>:** Polynomial Length Configuration bits
Configures the length of the polynomial (Polynomial Length – 1).

### REGISTER 29-4: PMD4: PERIPHERAL MODULE DISABLE 4 CONTROL 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 | U-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | REFOMD | — | — | — |
| 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 15-4 **Unimplemented:** Read as ‘0’

bit 3 **REFOMD:** Reference Clock Module Disable bit

1 = Reference clock module is disabled
0 = Reference clock module is enabled

bit 2-0 **Unimplemented:** Read as ‘0’

## 32.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

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

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 32.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 32.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 32.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

### TABLE 33-5: OPERATING CURRENT (IDD)[2]

**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.[1] | Max. | Units | Conditions | | |
|---|---|---|---|---|---|---|
| DC20 | 7.76 | 10.7 | mA | -40°C | 3.3V | 10 MIPS (N1 = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz) |
| | 7.49 | 10 | mA | +25°C | | |
| | 7.82 | 12.2 | mA | +85°C | | |
| | 10.32 | 20.45 | mA | +125°C | | |
| DC21 | 10.36 | 13.1 | mA | -40°C | 3.3V | 20 MIPS (N1 = 1, N2 = 5, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 80 MHz) |
| | 10.09 | 12.45 | mA | +25°C | | |
| | 10.42 | 14.5 | mA | +85°C | | |
| | 12.89 | 23.45 | mA | +125°C | | |
| DC22 | 14.54 | 17.45 | mA | -40°C | 3.3V | 40 MIPS (N1 = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz) |
| | 14.26 | 16.7 | mA | +25°C | | |
| | 14.58 | 18.9 | mA | +85°C | | |
| | 17.06 | 27.4 | mA | +125°C | | |
| DC23 | 22.2 | 25.4 | mA | -40°C | 3.3V | 70 MIPS (N1 = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz) |
| | 21.91 | 24.9 | mA | +25°C | | |
| | 22.21 | 27 | mA | +85°C | | |
| | 24.65 | 35.1 | mA | +125°C | | |
| DC24 | 27.36 | 30.7 | mA | -40°C | 3.3V | 90 MIPS (N1 = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz) |
| | 26.96 | 30.5 | mA | +25°C | | |
| | 26.68 | 31.7 | mA | +85°C | | |
| | 29.01 | 39.9 | mA | +125°C | | |
| DC25 | 27.14 | 30.9 | mA | -40°C | 3.3V | 100 MIPS (N1 = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz) |
| | 26.54 | 30.1 | mA | +25°C | | |
| | 26.79 | 31.7 | mA | +85°C | | |
| | 29.23 | 40 | mA | +125°C | | |

**Note 1:** Data in the “Typ.” column are for design guidance only and are not tested.

**2:** Base Run current (IDD) is measured as follows:

- Oscillator is switched to EC+PLL mode in software
- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to VDD – 0.3V
- OSC2 is configured as an I/O in the Configuration Words (OSCIOFCN (FOSC<2>) = `0`)
- FSCM is disabled (FCKSM<1:0> (FOSC<7:6>) = `01`)
- Watchdog Timer is disabled (FWDT<15> = `0` and WDTCONL<15> = `0`)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all ‘`1`’s)
- JTAG is disabled (JTAGEN (FICD<5>) = `0`)
- `NOP` instructions are executed in `while(1)` loop