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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf6680t-i-l

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
B2CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	45, 230
B1D7(7)	B1D77	B1D76	B1D75	B1D74	B1D73	B1D72	B1D71	B1D70	xxxx xxxx	45, 230
B1D6(7)	B1D67	B1D66	B1D65	B1D64	B1D63	B1D62	B1D61	B1D60	xxxx xxxx	45, 230
B1D5(7)	B1D57	B1D56	B1D55	B1D54	B1D53	B1D52	B1D51	B1D50	xxxx xxxx	45, 230
B1D4(7)	B1D47	B1D46	B1D45	B1D44	B1D43	B1D42	B1D41	B1D40	xxxx xxxx	45, 230
B1D3(7)	B1D37	B1D36	B1D35	B1D34	B1D33	B1D32	B1D31	B1D30	xxxx xxxx	45, 230
B1D2(7)	B1D27	B1D26	B1D25	B1D24	B1D23	B1D22	B1D21	B1D20	xxxx xxxx	45, 230
B1D1(7)	B1D17	B1D16	B1D15	B1D14	B1D13	B1D12	B1D11	B1D10	xxxx xxxx	46, 230
B1D0(7)	B1D07	B1D06	B1D05	B1D04	B1D03	B1D02	B1D01	B1D00	xxxx xxxx	46, 230
B1DLC(7)	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	46, 230
B1EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	46, 230
B1EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	46, 230
B1SIDL(7)	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	46, 230
B1SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	46, 230
B1CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	46, 230
B0D7(7)	B0D77	B0D76	B0D75	B0D74	B0D73	B0D72	B0D71	B0D70	xxxx xxxx	46, 230
B0D6(7)	B0D67	B0D66	B0D65	B0D64	B0D63	B0D62	B0D61	B0D60	xxxx xxxx	46, 230
B0D5(7)	B0D57	B0D56	B0D55	B0D54	B0D53	B0D52	B0D51	B0D50	xxxx xxxx	46, 230
B0D4(7)	B0D47	B0D46	B0D45	B0D44	B0D43	B0D42	B0D41	B0D40	xxxx xxxx	46, 230
B0D3(7)	B0D37	B0D36	B0D35	B0D34	B0D33	B0D32	B0D31	B0D30	xxxx xxxx	46, 230
B0D2(7)	B0D27	B0D26	B0D25	B0D24	B0D23	B0D22	B0D21	B0D20	xxxx xxxx	46, 230
B0D1(7)	B0D17	B0D16	B0D15	B0D14	B0D13	B0D12	B0D11	B0D10	xxxx xxxx	46, 230
B0D0(7)	B0D07	B0D06	B0D05	B0D04	B0D03	B0D02	B0D01	B0D00	xxxx xxxx	46, 230
B0DLC(7)	—	RTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	46, 230
B0EIDL(7)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	46, 230
B0EIDH(7)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	46, 230
B0SIDL(7)	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	46, 230
B0SIDH(7)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	46, 230
B0CON(5, 7)	RXFUL/ TXBIF	RXM1/ TXABT	RTRRO/ TXLARB	FILHIT4/ TXERR	FILHIT3/ TXREQ	FILHIT2/ RTREN	FILHIT1/ TXPRI1	FILHIT0/ TXPRI0	0000 0000	46, 230
TXBIE(7)	—	—	—	TXB2IE	TXB1IE	TXB0IE	—	—	---0 00--	46, 230
BIE0(7)	B5IE	B4IE	B3IE	B2IE	B1IE	B0IE	RXB1IE	RXB0IE	0000 0000	46, 230
BSEL0(7)	B5TXEN	B4TXEN	B3TXEN	B2TXEN	B1TXEN	B0TXEN	—	—	0000 00--	46, 230
MSEL3(7)	FIL15_1	FIL15_0	FIL14_1	FIL14_0	FIL13_1	FIL13_0	FIL12_1	FIL12_0	0000 0000	46, 230
MSEL2(7)	FIL11_1	FIL11_0	FIL10_1	FIL10_0	FIL9_1	FIL9_0	FIL8_1	FIL8_0	0000 0000	46, 230
MSEL1(7)	FIL7_1	FIL7_0	FIL6_1	FIL6_0	FIL5_1	FIL5_0	FIL4_1	FIL4_0	0000 0101	46, 230
MSEL0(7)	FIL3_1	FIL3_0	FIL2_1	FIL2_0	FIL1_1	FIL1_0	FIL0_1	FIL0_0	0101 0000	46, 230
SDFLC(7)	—	—	—	DFLC4	DFLC3	DFLC2	DFLC1	DFLC0	---0 0000	46, 230
RXFCON1(7)	RXF15EN	RXF14EN	RXF13EN	RXF12EN	RXF11EN	RXF10EN	RXF9EN	RXF8EN	0000 0000	46, 230
RXFCON0(7)	RXF7EN	RXF6EN	RXF5EN	RXF4EN	RXF3EN	RXF2EN	RXF1EN	RXF0EN	0011 1111	47, 230

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{MCLR}$ is disabled.

7: This register reads all '0's until the ECAN module is set up in Mode 1 or Mode 2.

6.2 16-bit Mode

The external memory interface implemented in PIC18F8X8X devices operates only in 16-bit mode. The mode selection is not software configurable but is programmed via the configuration bits.

The WM<1:0> bits in the MEMCON register determine three types of connections in 16-bit mode. They are referred to as:

- 16-bit Byte Write
- 16-bit Word Write
- 16-bit Byte Select

These three different configurations allow the designer maximum flexibility in using 8-bit and 16-bit memory devices.

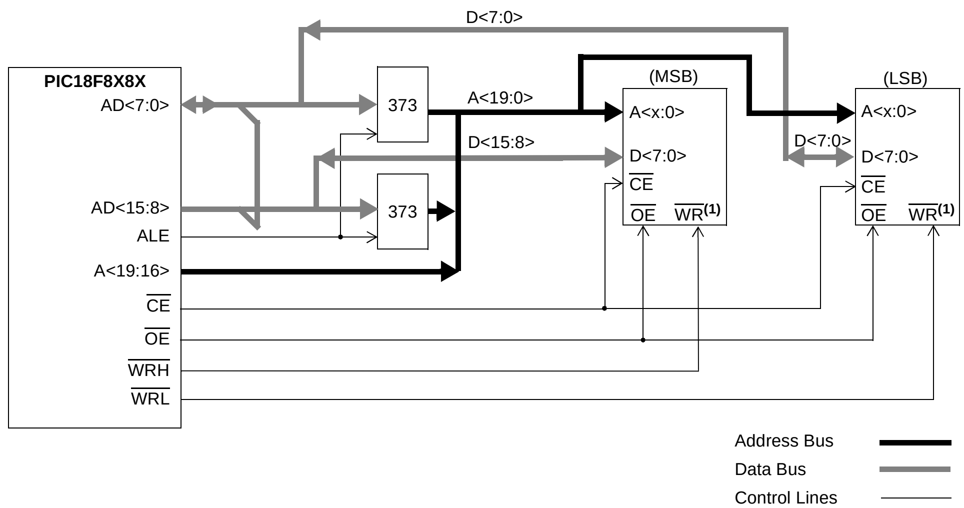
For all 16-bit modes, the Address Latch Enable (ALE) pin indicates that the Address bits (A<15:0>) are available on the external memory interface bus. Following the address latch, the Output Enable signal ($\overline{OE}$) will enable both bytes of program memory at once to form a 16-bit instruction word.

In Byte Select mode, JEDEC standard Flash memories will require BA0 for the byte address line, and one I/O line to select between Byte and Word mode. The other 16-bit modes do not need BA0. JEDEC standard static RAM memories will use the $\overline{UB}$ or $\overline{LB}$ signals for byte selection.

6.2.1 16-BIT BYTE WRITE MODE

Figure 6-1 shows an example of 16-bit Byte Write mode for PIC18F8X8X devices.

FIGURE 6-1: 16-BIT BYTE WRITE MODE EXAMPLE



Note 1: This signal only applies to table writes. See **Section 5.1 "Table Reads and Table Writes"**.

8.0 8 x 8 HARDWARE MULTIPLIER

8.1 Introduction

An 8 x 8 hardware multiplier is included in the ALU of the PIC18F6585/8585/6680/8680 devices. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored in the 16-bit product register pair (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register.

Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the device to be used in applications previously reserved for Digital Signal Processors.

Table 8-1 shows a performance comparison between enhanced devices using the single-cycle hardware multiply and performing the same function without the hardware multiply.

8.2 Operation

Example 8-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiply. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W         ;
MULWF   ARG2            ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
```

EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W         ;
MULWF   ARG2            ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
BTFSC   ARG2, SB        ; Test Sign Bit
SUBWF   PRODH           ; PRODH = PRODH
                        ;         - ARG1
MOVF    ARG2, W         ;
BTFSC   ARG1, SB        ; Test Sign Bit
SUBWF   PRODH           ; PRODH = PRODH
                        ;         - ARG2
```

TABLE 8-1: PERFORMANCE COMPARISON

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 μs
	Hardware multiply	1	1	100 ns	400 ns	1 μs
8 x 8 signed	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs
	Hardware multiply	6	6	600 ns	2.4 μs	6 μs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs
	Hardware multiply	24	24	2.4 μs	9.6 μs	24 μs
16 x 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs
	Hardware multiply	36	36	3.6 μs	14.4 μs	36 μs

REGISTER 17-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode.The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
0 = No overflow

Note: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables serial port and configures SCK, SDO, SDI, and $\overline{SS}$ as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0011 = SPI Master mode, clock = TMR2 output/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

Note: Bit combinations not specifically listed here are either reserved or implemented in I^{2}C mode only.

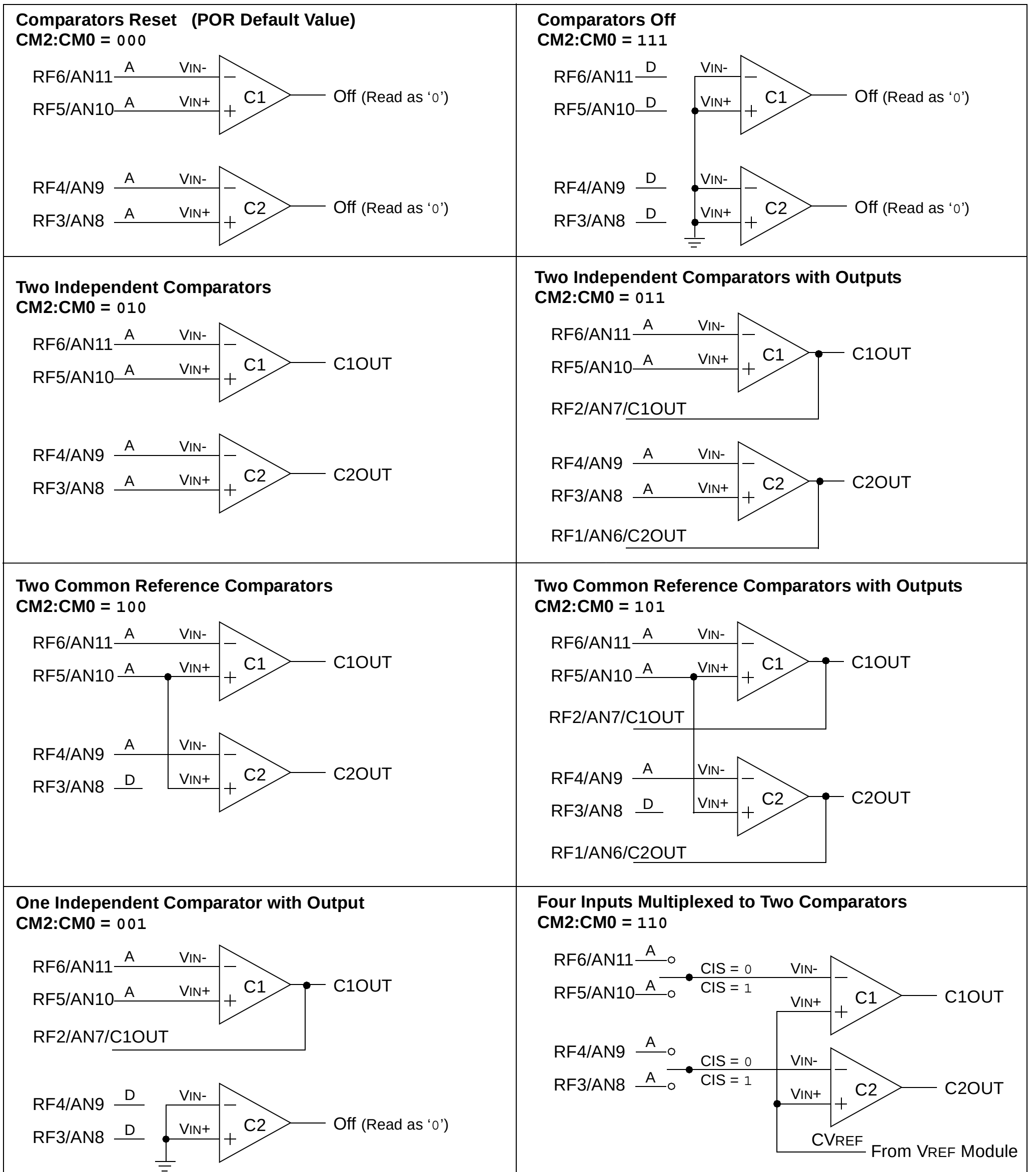
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

20.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select these modes. Figure 20-1 shows the eight possible modes. The TRISF register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in **Section 27.0 "Electrical Characteristics"**.

Note:	Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur.

FIGURE 20-1: COMPARATOR I/O OPERATING MODES



A = Analog Input, port reads zeros always D = Digital Input CIS (CMCON<3>) = Comparator Input Switch

REGISTER 23-2: CANSTAT: CAN STATUS REGISTER (CONTINUED)

bit 4-0 Mode 1,2:

EICODE4:EICODE0: Interrupt Code bits in Mode 1 and Mode 2

When an interrupt occurs, a prioritized coded interrupt value will be present in these bits. This code indicates the source of the interrupt. Unlike ICODE bits in Mode 0, these bits may not be copied directly to EWIN bits to map interrupted buffer to Access Bank area. If required, user software may maintain a table in program memory to map EICODE bits to EWIN bits and access interrupt buffer in Access Bank area.

	EICODE4:EICODE0 Value
No interrupt	00000
Error interrupt	00010
TXB2 interrupt	00100
TXB1 interrupt	00110
TXB0 interrupt	01000
RXB1 interrupt	10001/10000(2)
RXB0 interrupt	10000
Wake-up interrupt	01110
RX/TX B0 interrupt	10010(2)
RX/TX B1 interrupt	10011(2)
RX/TX B2 interrupt	10100(2)
RX/TX B3 interrupt	10101(2)
RX/TX B4 interrupt	10110(2)
RX/TX B4 interrupt	10111(2)

Note 1: To achieve maximum power saving and/or able to wake-up on CAN bus activity, switch CAN module to Disable mode before putting the device to Sleep.

2: In Mode 2, if the buffer is configured as a receiver, EICODE bits will always contain '10000' upon interrupt.

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

REGISTER 23-24: BnSIDH: TX/RX BUFFER n STANDARD IDENTIFIER REGISTERS, HIGH BYTE IN RECEIVE MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 0][(1)]

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7							bit 0

bit 7-0 **SID10:SID3:** Standard Identifier bits, if EXIDE (BnSIDL<3>) = 0;
Extended Identifier bits EID28:EID21, if EXIDE = 1.

Note 1: These registers are available in Mode 1 and 2 only.

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

REGISTER 23-25: BnSIDH: TX/RX BUFFER n STANDARD IDENTIFIER REGISTERS, HIGH BYTE IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 1][(1)]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7							bit 0

bit 7-0 **SID10:SID3:** Standard Identifier bits, if EXIDE (BnSIDL<3>) = 0;
Extended Identifier bits EID28:EID21, if EXIDE = 1.

Note 1: These registers are available in Mode 1 and 2 only.

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

TABLE 23-1: CAN CONTROLLER REGISTER MAP (CONTINUED)

Address[1]	Name	Address	Name	Address	Name	Address	Name
DFFh	—[4]	DDFh	—[4]	DBFh	—[4]	D9Fh	—[4]
DFEh	—[4]	DDEh	—[4]	DBEh	—[4]	D9Eh	—[4]
DFDh	—[4]	DDDh	—[4]	DBDh	—[4]	D9Dh	—[4]
DFCh	TXBIE	DDCh	—[4]	DBCh	—[4]	D9Ch	—[4]
DFBh	—[4]	DDBh	—[4]	DBBh	—[4]	D9Bh	—[4]
DFAh	BIE0	DDAh	—[4]	DBAh	—[4]	D9Ah	—[4]
DF9h	—[4]	DD9h	—[4]	DB9h	—[4]	D99h	—[4]
DF8h	BSEL0	DD8h	SDFLC	DB8h	—[4]	D98h	—[4]
DF7h	—[4]	DD7h	—[4]	DB7h	—[4]	D97h	—[4]
DF6h	—[4]	DD6h	—[4]	DB6h	—[4]	D96h	—[4]
DF5h	—[4]	DD5h	RXFCON1	DB5h	—[4]	D95h	—[4]
DF4h	—[4]	DD4h	RXFCON0	DB4h	—[4]	D94h	—[4]
DF3h	MSEL3	DD3h	—[4]	DB3h	—[4]	D93h	RXF15EIDL
DF2h	MSEL2	DD2h	—[4]	DB2h	—[4]	D92h	RXF15EIDH
DF1h	MSEL1	DD1h	—[4]	DB1h	—[4]	D91h	RXF15SIDL
DF0h	MSEL0	DD0h	—[4]	DB0h	—[4]	D90h	RXF15SIDH
DEFh	—[4]	DCFh	—[4]	DAFh	—[4]	D8Fh	—[4]
DEEh	—[4]	DCEh	—[4]	DAEh	—[4]	D8Eh	—[4]
DEDh	—[4]	DCDh	—[4]	DADh	—[4]	D8Dh	—[4]
DECh	—[4]	DCCh	—[4]	DACh	—[4]	D8Ch	—[4]
DEBh	—[4]	DCBh	—[4]	DABh	—[4]	D8Bh	RXF14EIDL
DEAh	—[4]	DCAh	—[4]	DAAh	—[4]	D8Ah	RXF14EIDH
DE9h	—[4]	DC9h	—[4]	DA9h	—[4]	D89h	RXF14SIDL
DE8h	—[4]	DC8h	—[4]	DA8h	—[4]	D88h	RXF14SIDH
DE7h	RXFBCON7	DC7h	—[4]	DA7h	—[4]	D87h	RXF13EIDL
DE6h	RXFBCON6	DC6h	—[4]	DA6h	—[4]	D86h	RXF13EIDH
DE5h	RXFBCON5	DC5h	—[4]	DA5h	—[4]	D85h	RXF13SIDL
DE4h	RXFBCON4	DC4h	—[4]	DA4h	—[4]	D84h	RXF13SIDH
DE3h	RXFBCON3	DC3h	—[4]	DA3h	—[4]	D83h	RXF12EIDL
DE2h	RXFBCON2	DC2h	—[4]	DA2h	—[4]	D82h	RXF12EIDH
DE1h	RXFBCON1	DC1h	—[4]	DA1h	—[4]	D81h	RXF12SIDL
DE0h	RXFBCON0	DC0h	—[4]	DA0h	—[4]	D80h	RXF12SIDH

Note 1: Shaded registers are available in Access Bank low area while the rest are available in Bank 15.

2: CANSTAT register is repeated in these locations to simplify application firmware. Unique names are given for each instance of the controller register due to the Microchip header file requirement.

3: These registers are not CAN registers.

4: Unimplemented registers are read as '0'.

23.6.2 ABORTING TRANSMISSION

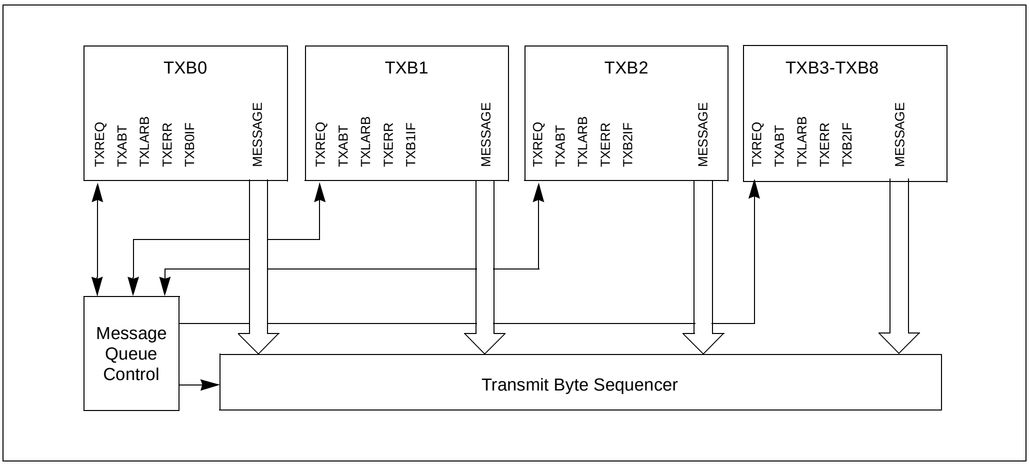
The MCU can request to abort a message by clearing the TXREQ bit associated with the corresponding message buffer (TXBnCON<3> or BnCON<3>). Setting the ABAT bit (CANCON<4>) will request an abort of all pending messages. If the message has not yet started transmission or if the message started but is interrupted by loss of arbitration or an error, the abort will be processed. The abort is indicated when the module sets the TXABT bit for the corresponding buffer (TXBnCON<6> or BnCON<6>). If the message has started to transmit, it will attempt to transmit the current message fully. If the current message is transmitted fully and is not lost to arbitration or an error, the TXABT bit will not be set because the message was transmitted successfully. Likewise, if a message is being transmitted during an abort request and the message is lost to arbitration or an error, the message will not be retransmitted and the TXABT bit will be set, indicating that the message was successfully aborted.

Once an abort is requested by setting ABAT or TXABT bits, it cannot be cleared to cancel the abort request. Only CAN module hardware or a POR condition can clear it.

23.6.3 TRANSMIT PRIORITY

Transmit priority is a prioritization within the PIC18F6585/8585/6680/8680 devices of the pending transmittable messages. This is independent from and not related to any prioritization implicit in the message arbitration scheme built into the CAN protocol. Prior to sending the SOF, the priority of all buffers that are queued for transmission is compared. The transmit buffer with the highest priority will be sent first. If more than one buffer has the same priority setting, the message is transmitted in the order of TXB2, TXB1, TXB0, B5, B4, B3, B2, B1, B0. There are four levels of transmit priority. If TXP bits for a particular message buffer are set to '11', that buffer has the highest possible priority. If TXP bits for a particular message buffer are '00', that buffer has the lowest possible priority.

FIGURE 23-2: TRANSMIT BUFFERS



24.2 Watchdog Timer (WDT)

The Watchdog Timer is a free-running, on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin. That means that the WDT will run even if the clock on the OSC1/CLKI and OSC2/CLKO/RA6 pins of the device has been stopped, for example, by execution of a `SLEEP` instruction.

During normal operation, a WDT time-out generates a device Reset (Watchdog Timer Reset). If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer wake-up). The $\overline{\text{TO}}$ bit in the RCON register will be cleared upon a WDT time-out.

The Watchdog Timer is enabled/disabled by a device configuration bit. If the WDT is enabled, software execution may not disable this function. When the WDTEN configuration bit is cleared, the SWDTEN bit enables/disables the operation of the WDT.

The WDT time-out period values may be found in **Section 27.0 "Electrical Characteristics"** under parameter #31. Values for the WDT postscaler may be assigned using the configuration bits.

> **Note 1:** The `CLRWDT` and `SLEEP` instructions clear the WDT and the postscaler if assigned to the WDT and prevent it from timing out and generating a device Reset condition.
>
> **2:** When a `CLRWDT` instruction is executed and the postscaler is assigned to the WDT, the postscaler count will be cleared but the postscaler assignment is not changed.

24.2.1 CONTROL REGISTER

Register 24-15 shows the WDTCON register. This is a readable and writable register which contains a control bit that allows software to override the WDT enable configuration bit, only when the configuration bit has disabled the WDT.

REGISTER 24-15: WDTCON REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN
bit 7							bit 0

bit 7-1 **Unimplemented**: Read as '0'

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit
1 = Watchdog Timer is on
0 = Watchdog Timer is turned off if the WDTEN configuration bit in the Configuration register = 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

24.3 Power-down Mode (Sleep)

Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit (RCON<3>) is cleared, the $\overline{TO}$ (RCON<4>) bit is set and the oscillator driver is turned off. The I/O ports maintain the status they had before the `SLEEP` instruction was executed (driving high, low, or high-impedance).

For lowest current consumption in this mode, place all I/O pins at either V_{DD} or V_{SS}, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are high-impedance inputs, high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (V_{IHMC}).

24.3.1 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin.
2. Watchdog Timer Wake-up (if WDT was enabled).
3. Interrupt from INT pin, RB port change or a peripheral interrupt.

The following peripheral interrupts can wake the device from Sleep:

1. PSP read or write.
2. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
3. TMR3 interrupt. Timer3 must be operating as an asynchronous counter.
4. CCP Capture mode interrupt.
5. Special event trigger (Timer1 in Asynchronous mode using an external clock).
6. MSSP (Start/Stop) bit detect interrupt.
7. MSSP transmit or receive in Slave mode (SPI/I^2C).
8. USART RX or TX (Synchronous Slave mode).
9. A/D conversion (when A/D clock source is RC).
10. EEPROM write operation complete.
11. LVD interrupt.
12. CAN wake-up interrupt.

Other peripherals cannot generate interrupts since during Sleep, no on-chip clocks are present.

External $\overline{MCLR}$ Reset will cause a device Reset. All other events are considered a continuation of program execution and will cause a "wake-up". The $\overline{TO}$ and $\overline{PD}$ bits in the RCON register can be used to determine the cause of the device Reset. The $\overline{PD}$ bit which is set on power-up is cleared when Sleep is invoked. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up).

When the `SLEEP` instruction is being executed, the next instruction (PC + 2) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address. In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

24.3.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If an interrupt condition (interrupt flag bit and interrupt enable bits are set) occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.
- If the interrupt condition occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from Sleep. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the `SLEEP` instruction was executed as a `NOP`.

To ensure that the WDT is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

25.0 INSTRUCTION SET SUMMARY

The PIC18 instruction set adds many enhancements to the previous PIC MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets.

Most instructions are a single program memory word (16 bits) but there are three instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 25-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 25-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by 'f')
2. The destination of the result (specified by 'd')
3. The accessed memory (specified by 'a')

The file register designator 'f' specifies which file register is to be used by the instruction.

The destination designator 'd' specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the WREG register. If 'd' is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by 'f')
2. The bit in the file register (specified by 'b')
3. The accessed memory (specified by 'a')

The bit field designator 'b' selects the number of the bit affected by the operation, while the file register designator 'f' represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by 'k')
- The desired FSR register to load the literal value into (specified by 'f')
- No operand required (specified by '—')

The **control** instructions may use some of the following operands:

- A program memory address (specified by 'n')
- The mode of the call or return instructions (specified by 's')
- The mode of the table read and table write instructions (specified by 'm')
- No operand required (specified by '—')

All instructions are a single word except for three double-word instructions. These three instructions were made double-word instructions so that all the required information is available in these 32 bits. In the second word, the 4 MSbs are '1's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 25-1 shows the general formats that the instructions can have.

All examples use the format 'nnh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

The Instruction Set Summary, shown in Table 25-2, lists the instructions recognized by the Microchip Assembler (MPASM™).

Section 25.1 "Instruction Set" provides a description of each instruction.

BNOV **Branch if Not Overflow**

Syntax: [*label*] BNOV n

Operands: $-128 \leq n \leq 127$

Operation: if overflow bit is '0'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0101	nnnn	nnnn

Description: If the Overflow bit is '0', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC+2+2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BNOV Jump`

Before Instruction
PC = address (HERE)

After Instruction
If Overflow = 0;
PC = address (Jump)
If Overflow = 1;
PC = address (HERE+2)

BNZ **Branch if Not Zero**

Syntax: [*label*] BNZ n

Operands: $-128 \leq n \leq 127$

Operation: if zero bit is '0'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0001	nnnn	nnnn

Description: If the Zero bit is '0', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC+2+2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BNZ Jump`

Before Instruction
PC = address (HERE)

After Instruction
If Zero = 0;
PC = address (Jump)
If Zero = 1;
PC = address (HERE+2)

27.2 DC Characteristics: Power-down and Supply Current PIC18FXX8X (Industrial, Extended) PIC18LFXX8X (Industrial)

PIC18LFXX8X (Industrial)	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC18FXX8X (Industrial, Extended)	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param. No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**	
	Power-down Current (IPD)[1]					
D020	PIC18LFXX8X	0.2	1	μA	-40°C	VDD = 2.0V, (Sleep mode)
		0.2	1	μA	+25°C	
		5.0	10	μA	+85°C	
D020A	PIC18LFXX8X	0.4	1	μA	-40°C	VDD = 3.0V, (Sleep mode)
		0.4	1	μA	+25°C	
		3.0	18	μA	+85°C	
D020B	All devices	0.7	2	μA	-40°C	VDD = 5.0V, (Sleep mode)
		0.7	2	μA	+25°C	
		15.0	32	μA	+85°C	

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

27.2 DC Characteristics: Power-down and Supply Current PIC18FXX8X (Industrial, Extended) PIC18LFXX8X (Industrial) (Continued)

PIC18LFXX8X (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18FXX8X (Industrial, Extended)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param. No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD)(2,3)						
D010	PIC18LFXX8X	500	500	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz, EC oscillator
		300	500	μA	+25°C		
		850	1000	μA	+85°C		
	PIC18LFXX8X	500	900	μA	-40°C	VDD = 3.0V	
		500	900	μA	+25°C		
		1	1.5	mA	+85°C		
	All devices	1	2	mA	-40°C	VDD = 5.0V	
		1	2	mA	+25°C		
		1.3	3	mA	+85°C		
	PIC18LFXX8X	1	2	mA	-40°C	VDD = 2.0V	FOSC = 4 MHz, EC oscillator
		1	2	mA	+25°C		
		1.5	2.5	mA	+85°C		
	PIC18LFXX8X	1.5	2	mA	-40°C	VDD = 3.0V	
		1.5	2	mA	+25°C		
		2	2.5	mA	+85°C		
	All devices	3	5	mA	-40°C	VDD = 5.0V	
		3	5	mA	+25°C		
		4	6	mA	+85°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

FIGURE 28-7: TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} (LP MODE)

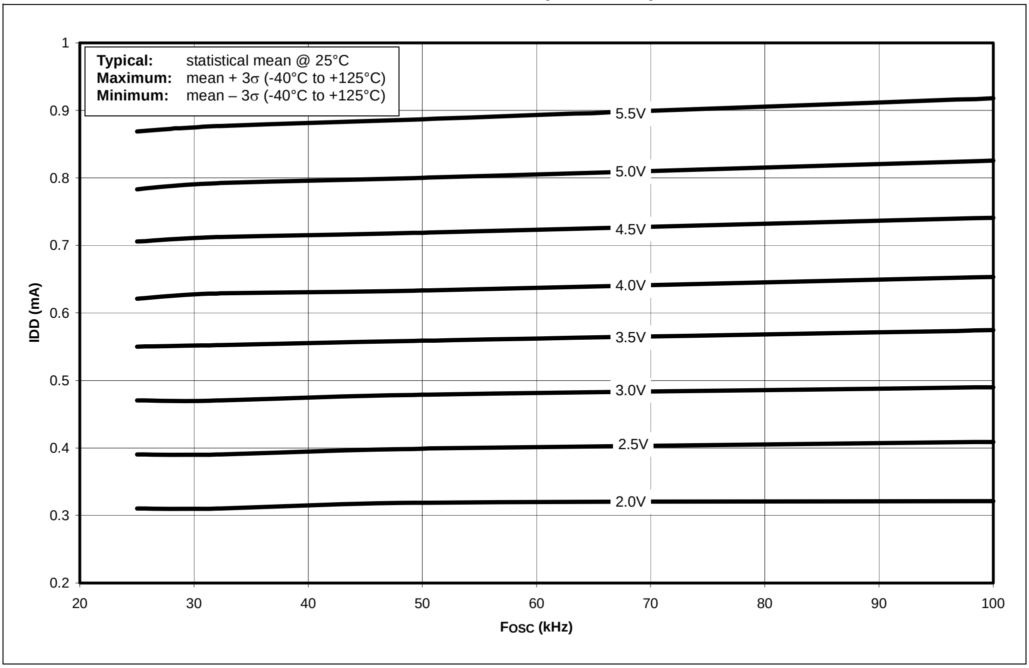
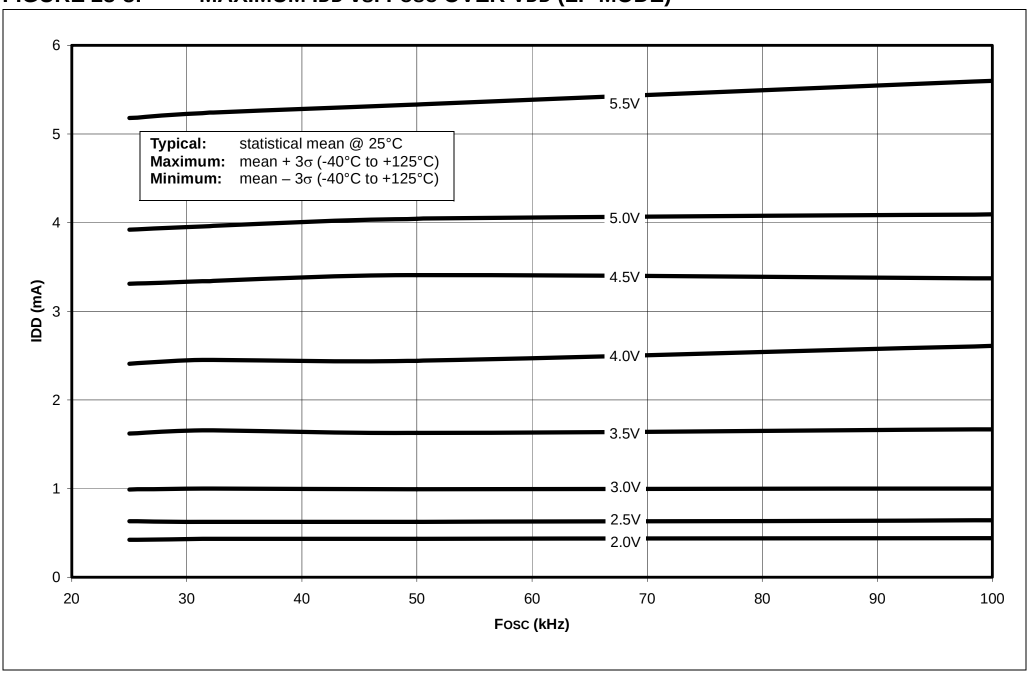
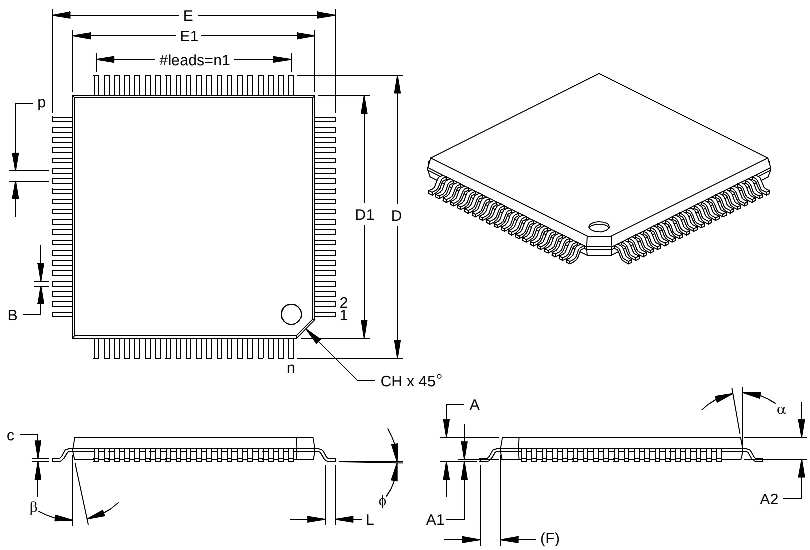


FIGURE 28-8: MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD} (LP MODE)



80-Lead Plastic Thin Quad Flatpack (PT) 12x12x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)

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



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		80			80	
Pitch	p		.020			0.50	
Pins per Side	n1		20			20	
Overall Height	A	.039	.043	.047	1.00	1.10	1.20
Molded Package Thickness	A2	.037	.039	.041	0.95	1.00	1.05
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Foot Length	L	.018	.024	.030	0.45	0.60	0.75
Footprint (Reference)	(F)		.039			1.00	
Foot Angle	φ	0	3.5	7	0	3.5	7
Overall Width	E	.541	.551	.561	13.75	14.00	14.25
Overall Length	D	.541	.551	.561	13.75	14.00	14.25
Molded Package Width	E1	.463	.472	.482	11.75	12.00	12.25
Molded Package Length	D1	.463	.472	.482	11.75	12.00	12.25
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.007	.009	.011	0.17	0.22	0.27
Pin 1 Corner Chamfer	CH	.025	.035	.045	0.64	0.89	1.14
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-026
Drawing No. C04-092

NOTES: