



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	ОТР
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c716-20-p

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

#### 2.2 **Data Memory Organization**

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 and RP0 are the bank select bits.

RP1 <sup>(1)</sup>	RP0 (STATUS<6:5>)
= 00 $\rightarrow$	Bank 0
$= 01 \rightarrow$	Bank 1
= $10 \rightarrow$	Bank 2 (not implemented)
= 11 $\rightarrow$	Bank 3 (not implemented)
Note 1	<ul> <li>Maintain this bit clear to ensure upward compatibility with future products.</li> </ul>

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some "high use" Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

#### GENERAL PURPOSE REGISTER 2.2.1 FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR (see Section 2.5 "Indirect Addressing, INDF and FSR Registers").

IGURE 2-3:	REGISTER	FILE MAP
		••••••••

	F	lle	
۸	44	ro	~

F

File			File
Address			Address
00h	INDF <sup>(1)</sup>	INDF(")	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	DATACCP	TRISCCP	87h
08h			88h
09h			89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh	TMR1L	PCON	8Eh
0Fh	TMR1H		8Fh
10h	T1CON		90h
11h	TMR2		91h
12h	T2CON	PR2	92h
13h			93h
14h			94h
15h	CCPR1L		95h
16h	CCPR1H		96h
17h	CCP1CON		97h
18h			98h
19h			99h
1Ah			9Ah
1Bh			9Bh
1Ch			9Ch
1Dh			9Dh
1Eh	ADRES		9Eh
1Fh	ADCON0	ADCON1	9Fh
20h		General	A0h
		Purpose	
	General	Registers	BFh
	Registers	52 Dytes	COb
	96 Bytes		Con
7Fh			FFh
	Bank 0	Bank 1	1
Un	implemented d	ata memory loc	ations,
read	<b>l as</b> '0'.		
NOTE 1: NO	ot a physical re	gister.	

#### 4.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- · Internal or external clock select
- Edge select for external clock
- 8-bit software programmable prescaler
- · Interrupt on overflow from FFh to 00h

Figure 4-1 is a simplified block diagram of the Timer0 module.

Additional information on timer modules is available in the PIC<sup>®</sup> Mid-Range Reference Manual, (DS33023).

#### 4.1 **Timer0** Operation

Timer0 can operate as a timer or as a counter.

Timer mode is selected by clearing bit T0CS (OPTION\_REG<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting bit T0CS (OPTION\_REG<5>). In Counter mode, Timer0 will increment on every rising or falling edge of pin RA4/ T0CKI. The incrementing edge is determined by the Edge Select Timer0 Source bit TOSE (OPTION\_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed below.

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (Tosc). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

Additional information on external clock requirements is available in the PIC<sup>®</sup> Mid-Range Reference Manual, (DS33023).

#### 4.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer, respectively (Figure 4-2). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available, which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer and vice-versa.

The prescaler is not readable or writable.

The PSA and PS2:PS0 bits (OPTION\_REG<3:0>) determine the prescaler assignment and prescale ratio.

Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable.

Setting bit PSA will assign the prescaler to the Watchdog Timer (WDT). When the prescaler is assigned to the WDT, prescale values of 1:1, 1:2, ..., 1:128 are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF 1, MOVWF 1, BSF 1, x....etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT.

Note: Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count, but will not change the prescaler assignment.



#### FIGURE 4-1: TIMER0 BLOCK DIAGRAM

NOTES:

## 7.0 CAPTURE/COMPARE/PWM (CCP) MODULE(S)

Each CCP (Capture/Compare/PWM) module contains a 16-bit register, which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM master/slave Duty Cycle register. Table 7-1 shows the timer resources of the CCP module modes.

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

#### FIGURE 7-1: CCP1CON REGISTER (ADDRESS 17h)

U-0 U-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 DC1B1 DC1B0 CCP1M3 CCP1M2 CCP1M1 CCP1M0 R = Readable bit W = Writable bit bit7 bit0 U = Unimplemented bit, read as '0' -n = Value at POR Reset bit 7-6: Unimplemented: Read as '0' bit 5-4: DC1B1:DC1B0: PWM Least Significant bits Capture Mode: Unused Compare Mode: Unused PWM Mode: These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPR1L. bit 3-0: CCP1M3:CCP1M0: CCP1 Mode Select bits 0000 = Capture/Compare/PWM off (resets CCP1 module) 0100 = Capture mode, every falling edge 0101 = Capture mode, every rising edge 0110 = Capture mode, every 4th rising edge 0111 = Capture mode, every 16th rising edge 1000 = Compare mode, set output on match (CCP1IF bit is set) 1001 = Compare mode, clear output on match (CCP1IF bit is set) 1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected) 1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled)) 11xx = PWM mode

#### FIGURE 7-2: TRISCCP REGISTER (ADDRESS 87H)



Additional information on the CCP module is available in the PIC<sup>®</sup> Mid-Range Reference Manual, (DS33023).

# TABLE 7-1:CCP MODE – TIMER<br/>RESOURCE

Timer Resource
Timer1
Timer1
Timer2

### 7.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RB3/CCP1 pin is either:

- driven High
- driven Low
- remains Unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

#### FIGURE 7-4: COMPARE MODE OPERATION BLOCK DIAGRAM



#### 7.2.1 CCP PIN CONFIGURATION

The user must configure the RB3/CCP1 pin as the CCP output by clearing the TRISCCP<2> bit.

Note:	Clearing the CCP1CON register will force the RB3/CCP1 compare output latch to
	the default low level. This is neither the
	PORTB I/O data latch nor the DATACCP
	latch.

#### 7.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

#### 7.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

#### 7.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The Special Event Trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The Special Event Trigger output of CCP1 also starts an A/D conversion (if the A/D module is enabled).

**Note:** The Special Event Trigger from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>).

#### TABLE 7-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
07h	DATACCP			—	—	—	DCCP	_	DT1CK	XXXX XXXX	xxxx xuxu
0Bh,8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 0002	0000 000u
0Ch	PIR1		ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	-0000	-0000
0Eh	TMR1L	Holding	olding Register for the Least Significant Byte of the 16-bit TMR1 Register					ter	XXXX XXXX	uuuu uuuu	
0Fh	TMR1H	Holding	Registe	r for the Mo	st Significa	nt Byte of th	e 16-bit TN	/IR1 Regist	er	XXXX XXXX	uuuu uuuu
10h	T1CON			T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	uu uuuu
15h	CCPR1L	Capture	apture/Compare/PWM Register 1 (LSB)					XXXX XXXX	uuuu uuuu		
16h	CCPR1H	Capture	/Compa	re/PWM Re	gister 1 (MS	SB)				XXXX XXXX	uuuu uuuu
17h	CCP1CON			DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	00 0000	00 0000
87h	TRISCCP			—	—	—	TCCP	_	TT1CK	xxxx x1x1	xxxx x1x1
8Ch	PIE1		ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	-0000	-0000

Legend: x = unknown, u = unchanged, — = unimplemented read as '0'. Shaded cells are not used by Capture and Timer1.

#### 8.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.5TAD per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for TAD are:

- 2Tosc
- 8Tosc
- 32Tosc
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6  $\mu s.$ 

Table 8-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

### 8.3 Configuring Analog Port Pins

The ADCON1 and TRISA registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

- Note 1: When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.
  - 2: Analog levels on any pin that is defined as a digital input (including the AN3:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

#### TABLE 8-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock	Source (TAD)	Device Frequency				
Operation	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz	
2Tosc	00	100 ns <sup>(2)</sup>	400 ns <sup>(2)</sup>	1.6 μs	6 µs	
8Tosc	01	400 ns <sup>(2)</sup>	1.6 μs	6.4 μs	24 μs <sup>(3)</sup>	
32Tosc	10	1.6 μs	6.4 μs	25.6 μs <sup>(3)</sup>	96 μs <b><sup>(3)</sup></b>	
RC <sup>(5)</sup>	11	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1,4)</sup>	2-6 μs <sup>(1)</sup>	

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 4  $\mu s.$ 

- **2:** These values violate the minimum required TAD time.
- **3:** For faster conversion times, the selection of another clock source is recommended.
- 4: When device frequency is greater than 1 MHz, the RC A/D conversion clock source is recommended for Sleep operation only.

5: For extended voltage devices (LC), please refer to Electrical Specifications section.

#### 9.2.3 RC OSCILLATOR

For timing insensitive applications, the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. In addition to this, the oscillator frequency will vary from unit-to-unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-4 shows how the R/C combination is connected to the PIC16CXXX.





#### 9.3 Reset

The PIC16CXXX differentiates between various kinds of Reset:

- Power-on Reset (POR)
- MCLR Reset during normal operation
- MCLR Reset during Sleep
- WDT Reset (during normal operation)
- WDT Wake-up (during Sleep)
- Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), on the MCLR and WDT Reset, on MCLR Reset during Sleep and Brownout Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The TO and PD bits are set or cleared differently in different Reset situations as indicated in Table 9-4. These bits are used in software to determine the nature of the Reset. See Table 9-6 for a full description of Reset states of all registers.

A simplified block diagram of the on-chip Reset circuit is shown in Figure 9-6.

The PIC microcontrollers have a  $\overline{\text{MCLR}}$  noise filter in the  $\overline{\text{MCLR}}$  Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive  $\overline{\text{MCLR}}$  pin low.

# PIC16C712/716







# PIC16C712/716



FIGURE 9-12: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

![](_page_9_Figure_3.jpeg)

#### FIGURE 9-13: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

![](_page_9_Figure_5.jpeg)

When the SLEEP instruction is being executed, the next instruction (PC + 1) 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 (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

#### WAKE-UP USING INTERRUPTS 9.13.2

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 the interrupt 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 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 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 PD bit. If the 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.

![](_page_10_Figure_9.jpeg)

GIE = 1 assumed. In this case after wake-up, the processor jumps to the interrupt routine. If GIE = 0, execution will continue in-line.

4: CLKOUT is not available in these osc modes, but shown here for timing reference.

#### 9.14 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

Note:	Microchip	does	not	recommend	code
	protecting	windov	ved d	levices.	

#### **ID** Locations 9.15

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify. It is recommended that only the 4 Least Significant bits of the ID location are used.

For ROM devices, these values are submitted along with the ROM code.

### **10.0 INSTRUCTION SET SUMMARY**

Each PIC16CXXX instruction is a 14-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 PIC16CXXX instruction set summary in Table 10-2 lists **byte-oriented**, **bitoriented**, and **literal and control** operations. Table 10-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

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

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

# TABLE 10-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= $0$ or 1) The assembler will generate code with x = $0$ . It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
PC	Program Counter
TO	Time-out bit
PD	Power-down bit
Z	Zero bit
DC	Digit Carry bit
С	Carry bit

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

- Byte-oriented operations
- Bit-oriented operations
- Literal and control operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1  $\mu$ s. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2  $\mu$ s.

Table 10-2 lists the instructions recognized by the MPASM assembler.

Figure 10-1 shows the general formats that the instructions can have.

Note:	To maintain upward compatibility with
	future PIC16CXXX products, do not use
	the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

# FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file re	egister op 8 7	perations 6	0			
OPCODE	d	f (FILE #)	-			
d = 0 for destination W d = 1 for destination f f = 7-bit file register address						
Bit-oriented file register operations						
OPCODE	b (Bl	T #) f (FILE #)				
Literal and control General	operatio	ns				
13	8	7	0			
OPCODE		k (literal)				
k = 8-bit immediate value						
CALL and GOTO instructions only						
<u>13 11 10 0</u>						
OPCODE k (literal)						
k = 11-bit immediate value						

A description of each instruction is available in the PIC<sup>®</sup> Mid-Range Reference Manual, (DS33023).

#### 11.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel<sup>®</sup> 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

#### 11.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 family of microcontrollers and dsPIC30F family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

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

#### 11.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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

# 11.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C 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 dsPIC30F instruction set
- · Support for fixed-point and floating-point data
- · Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

#### 11.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC<sup>®</sup> DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, as well as internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

#### 11.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft<sup>®</sup> Windows<sup>®</sup> 32-bit operating system were chosen to best make these features available in a simple, unified application.

#### 11.8 MPLAB ICE 4000 High-Performance In-Circuit Emulator

The MPLAB ICE 4000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PIC MCUs and dsPIC DSCs. Software control of the MPLAB ICE 4000 In-Circuit Emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, and up to 2 Mb of emulation memory.

The MPLAB ICE 4000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 11.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming<sup>™</sup> (ICSP<sup>™</sup>) protocol, offers costeffective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

### 11.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an SD/MMC card for file storage and secure data applications.

#### 11.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

#### 11.12 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM<sup>™</sup> and dsPICDEM<sup>™</sup> demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ<sup>®</sup> security ICs, CAN, IrDA<sup>®</sup>, PowerSmart<sup>®</sup> battery management, SEEVAL<sup>®</sup> evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page (www.microchip.com) and the latest *"Product Selector Guide"* (DS00148) for the complete list of demonstration, development and evaluation kits.

## 12.4 AC (Timing) Characteristics

#### 12.4.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created using one of the following formats:

1.	TppS2ppS
----	----------

2. TppS

Т			
F	Frequency	т	Time
Lowerc	case letters (pp) and their meanings:		
рр			
сс	CCP1	osc	OSC1
ck	CLKOUT	rd	RD
CS	CS	rw	RD or WR
di	SDI	sc	SCK
do	SDO	SS	SS
dt	Data in	tO	TOCKI
io	I/O port	t1	T1CKI
mc	MCLR	wr	WR
Upperc	case letters and their meanings:		
S			
F	Fall	Р	Period
Н	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

![](_page_16_Figure_1.jpeg)

# FIGURE 12-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

![](_page_16_Figure_3.jpeg)

![](_page_16_Figure_4.jpeg)

![](_page_16_Figure_5.jpeg)

# TABLE 12-4:RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER,<br/>AND BROWN-OUT RESET REQUIREMENTS

Parameter	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
NU.							
30	TmcL	MCLR Pulse Width (low)	2	_		μs	VDD = 5V, -40°C to +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024 Tosc	—	-	TOSC = OSC1 period
33*	TPWRT	Power-up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°C
34	Tioz	I/O High-impedance from MCLR Low or WDT Reset		_	2.1	μs	
35	TBOR	Brown-out Reset Pulse Width	100	—	_	μs	$VDD \le BVDD (D005)$

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# TABLE 12-7:A/D CONVERTER CHARACTERISTICS:<br/>PIC16C712/716-04 (COMMERCIAL, INDUSTRIAL, EXTENDED)<br/>PIC16C712/716-20 (COMMERCIAL, INDUSTRIAL, EXTENDED)<br/>PIC16LC712/716-04 (COMMERCIAL, INDUSTRIAL)

Param No.	Sym.	Characteristic		Min.	Тур†	Max.	Units	Conditions
A01	NR	Resolution		—	_	8-bits	bit	VREF = VDD = 5.12V, VSS £ VAIN £ VREF
A02	Eabs	Total Absolute error		-	_	< ± 1	LSb	VREF = VDD = 5.12V, VSS £ VAIN £ VREF
A03	EIL	Integral linearity error		-	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS £ VAIN £ VREF
A04	Edl	Differential linearity error		-	_	< ± 1	LSb	VREF = VDD = 5.12V, VSS £ VAIN £ VREF
A05	EFS	Full scale error		-	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS £ VAIN £ VREF
A06	EOFF	Offset error		-	_	< ± 1	LSb	VREF = VDD = 5.12V, VSS £ VAIN £ VREF
A10	—	Monotonicity		_	guaranteed (Note 3)	_	—	VSS £ VAIN £ VREF
A20	VREF	Reference voltage		2.5V	—	Vdd + 0.3	V	
A25	VAIN	Analog input voltage		Vss - 0.3	—	Vref + 0.3	V	
A30	ZAIN	Recommended impeda analog voltage source	nce of	-	_	10.0	kΩ	
A40	IAD	A/D conversion cur-	Standard	—	180	_	μΑ	Average current consump-
		rent (VDD)	Extended (LC)	—	90	—	μΑ	tion when A/D is on. <b>(Note 1)</b>
A50	IREF	VREF input current (Not	e 2)	10	_	1000	μΑ μΑ	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see <b>Section 9.1</b> <b>"Configuration Bits"</b> . During A/D Conversion
								cycle

2: \* These parameters are characterized but not tested.

3: † Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current.

The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

3: The A/D conversion result never decreases with an increase in the Input Voltage, and has no missing codes.

### **13.0 PACKAGING INFORMATION**

#### 13.1 Package Marking Information

#### 18-Lead PDIP

![](_page_18_Figure_4.jpeg)

#### 18-Lead CERDIP Windowed

![](_page_18_Picture_6.jpeg)

#### 18-Lead SOIC (.300")

![](_page_18_Picture_8.jpeg)

#### 20-Lead SSOP

![](_page_18_Picture_10.jpeg)

![](_page_18_Picture_11.jpeg)

#### Example

![](_page_18_Figure_13.jpeg)

#### Example

![](_page_18_Picture_15.jpeg)

#### Example

![](_page_18_Picture_17.jpeg)

Legend	: XXX Y YY WW NNN @3 *	Customer-specific information Year code (last digit of calendar year) Year code (last 2 digits of calendar year) Week code (week of January 1 is week '01') Alphanumeric traceability code Pb-free JEDEC designator for Matte Tin (Sn) This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.
Note:	In the event the full Microchip part number cannot be marked on one line, it wi be carried over to the next line, thus limiting the number of available characters for customer-specific information.	

TRISC Register	12
Postscaler, Timer2	
Select (TOUTPS3:TOUTPS0 Bits)	36
Postscaler, WDT	29
Assignment (PSA Bit) 14, Block Diagram	29
Block Diagram	20
Switching Between Timer() and WDT	30
Power-down Mode. See Sleep	00
Power-on Reset (POR) 51, 54, 55, 58,	59
Oscillator Start-up Timer (OST) 51,	55
POR Status (POR Bit)	18
Power Control (PCON) Register	58
Power-down (PD Bit) 13,	54
Power-on Reset Circuit, External	55
Power-up Timer (PWRT) 51,	55
Time-out (TO Bit)	52 57
Time-out Sequence	57
Time-out Sequence on Power-up	60
Timing Diagram	83
Prescaler, Capture	40
Prescaler, Timer0	29
Assignment (PSA Bit) 14,	29
Block Diagram	30
Rate Select (PS2:PS0 Bits) 14,	29
Switching Between Timer0 and WDT	30
Prescaler, TIMERT	32
Prescaler Timer2	42
Select (T2CKPS1:T2CKPS0 Bits)	36
Product Identification System	103
Program Counter	
PCL Register 11,	19
PCLATH Register 11, 19,	62
Reset Conditions	<b>F</b> O
Program Memory	58
	. 9
Interrupt Vector	58 . 9 9
Interrupt Vector	58 9 9 19
Interrupt Vector	58 9 9 19 9
Interrupt Vector	58 9 9 19 9 9
Interrupt Vector	58 9 9 .19 .9 .9 .65 67
Interrupt Vector	58 9 9 9 .9 .65 67 42
Interrupt Vector	58 9 9 9 9 9 .65 .67 .42 .42
Interrupt Vector Paging	58 9 9 9 9 9 9 9 
Interrupt Vector Paging	58 9 9 19 9 65 67 42 42 42 42
Interrupt Vector Paging	58 9 9 19 9 65 67 42 42 42 42 42 42
Interrupt Vector Paging	58 9 9 19 9 65 67 42 42 42 42 42 42
Interrupt Vector Paging	58 9 9 19 9 65 67 42 42 42 42 42 42 42 42 42 42
Interrupt Vector Paging	58 . 9 . 9 19 . 9 65 67 42 42 42 42 42 42 42 42 42 42 42 42 42
Interrupt Vector Paging	58 9 9 9 9 9 9 9 9 65 42 42 42 42 42 42 42 4
Interrupt Vector Paging	58 9 9 9 9 65 67 42 42 42 42 42 42 43 42 43 42 16 17
Interrupt Vector Paging	58 . 9 . 9 19 . 9 65 67 42 42 42 42 42 42 42 43 42 42 43 42 16 17
Interrupt Vector Paging	58 . 9 . 9 . 9 . 9 . 9 . 9 . 9 . 9 . 9 . 65 . 67 . 42 . 42 . 42 . 42 . 43 . 42 . 43 . 42 . 16 . 17
Interrupt Vector Paging	58 . 9 . 9 19 . 9 65 67 42 42 42 42 42 42 42 42 42 42 42 16 17 42
Interrupt Vector Paging	58 . 9 . 9 19 . 9 . 9 65 67 42 42 42 42 42 42 43 42 43 42 16 17 42
Interrupt Vector Paging	58 . 9 . 9 19 . 9 . 9 . 9 . 9 . 9 . 9 . 9 . 9 . 9

N	
RAM. See Data Memory	
Reader Response	104
Register File	10
Register File Map	10
Reset	51, 54
Block Diagram	56

Brown-out Reset (BOR). See Brown-out Reset (BOR	)
MCLR Reset. See MCLR	
Power-on Reset (POR). See Power-on Reset (POR)	
Reset Conditions for All Registers	59
Reset Conditions for PCON Register	58
Reset Conditions for Program Counter	58
Reset Conditions for STATUS Register	58
Timing Diagram	83
WDT Reset. See Watchdog Timer (WDT)	
Revision History	95

# S

Sleep	4 4
Software Simulator (MPLAB SIM) 70	0
Special Event Trigger. See Compare	
Special Features of the CPU 51	1
Special Function Registers 11	1
Speed, Operating	1
Stack 19	9
STATUS Register 11, 13, 62	2
C Bit	3
DC Bit 13	3
IRP Bit 13	3
PD Bit 13, 54	4
RP1:RP0 Bits13	3
TO Bit 13, 54	4
Z Bit	3

#### Т

1	
T1CON Register	11, 31
T1CKPS1:T1CKPS0 Bits	
T1OSCEN Bit	31
T1SYNC Bit	31
TMR1CS Bit	31
TMR1ON Bit	31
T2CON Register	11, 36
T2CKPS1:T2CKPS0 Bits	
TMR2ON Bit	36
TOUTPS3:TOUTPS0 Bits	36
Timer0	29
Block Diagram	29
Clock Source Edge Select (T0SE Bit)	14, 29
Clock Source Select (T0CS Bit)	14, 29
Overflow Enable (T0IE Bit)	
Overflow Flag (TOIF Bit)	15, 62
Overflow Interrupt	30, 62
Prescaler. See Prescaler, Timer0	,
Timing Diagram	84
TMR0 Register	11
Timer1	31
Block Diagram	32
Capacitor Selection	34
Clock Source Select (TMR1CS Bit)	31
External Clock Input Sync (T1SYNC Bit)	31
Module On/Off (TMR1ON Bit)	31
Oscillator	31, 34
Oscillator Enable (T1OSCEN Bit)	
Overflow Enable (TMR1IE Bit)	16
Overflow Flag (TMR1IF Bit)	17
Overflow Interrupt	31, 34
Prescaler. See Prescaler, Timer1	·
Special Event Trigger (CCP)	34, 41
T1CON Register	11, 31
Timing Diagram	
TMR1H Register	11, 31
5	

### **READER RESPONSE**

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: RE:	Technical Publications Manager Reader Response	Total Pages Sent
Fror	n: Name	
	Company	
	Address	
	City / State / ZIP / Country	
	Telephone: ()	FAX: ()
Арр	lication (optional):	
Wou	Ild you like a reply?YN	
Dev	ice: PIC16C712/716	Literature Number: DS41106C
Que	estions:	
1.	What are the best features of this document?	
2		tuora davalanment naada?
Ζ.	How does this document meet your hardware and sor	tware development needs?
3.	Do you find the organization of this document easy to	follow? If not, why?
4.	What additions to the document do you think would er	nhance the structure and subject?
5.	What deletions from the document could be made with	hout affecting the overall usefulness?
6.	Is there any incorrect or misleading information (what	and where)?
7	How would you improve this document?	
7.		