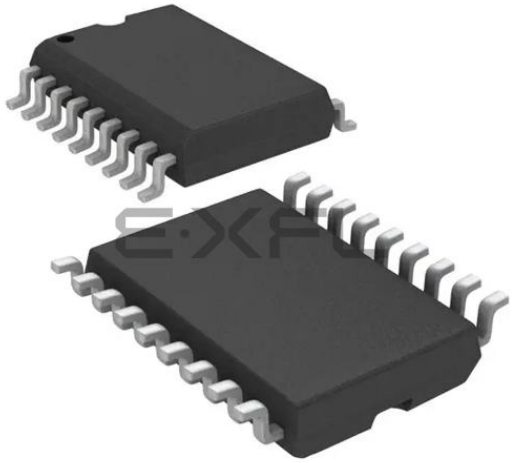


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	13
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	68 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	A/D 4x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c711-04i-so

TABLE 4-2: PIC16C715 SPECIAL FUNCTION REGISTER SUMMARY (Cont.'d)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR, PER	Value on all other resets (3)
Bank 1											
80h(1)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(1)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h(1)	STATUS	IRP(4)	RP1(4)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h(1)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	—	Unimplemented								—	—
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah(1,2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the PC					---0 0000	---0 0000
8Bh(1)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	—	ADIE	—	—	—	—	—	—	-0-- ----	-0-- ----
8Dh	—	Unimplemented								—	—
8Eh	PCON	MPEEN	—	—	—	—	$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$	u--- -1qq	u--- -1uu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	—	Unimplemented								—	—
93h	—	Unimplemented								—	—
94h	—	Unimplemented								—	—
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	—	Unimplemented								—	—
99h	—	Unimplemented								—	—
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	—	Unimplemented								—	—
9Fh	ADCON1	—	—	—	—	—	—	PCFG1	PCFG0	---- --00	---- --00

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from either bank.
2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
3: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.
4: The IRP and RP1 bits are reserved on the PIC16C715, always maintain these bits clear.

© 1997 Microchip Technology Inc.

4.2.2.3 INTCON REGISTER

Applicable Devices	710	71	711	715

The INTCON Register is a readable and writable register which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

Note: Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

FIGURE 4-9: INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **GIE:**(1) Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

bit 6: **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables A/D interrupt
0 = Disables A/D interrupt

bit 5: **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4: **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3: **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2: **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1: **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0: **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Note 1: For the PIC16C71, if an interrupt occurs while the GIE bit is being cleared, the GIE bit may be unintentionally re-enabled by the RETFIE instruction in the user's Interrupt Service Routine. Refer to Section 8.5 for a detailed description.

Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed “on the fly” during program execution.

Note: To avoid an unintended device RESET, the following instruction sequence (shown in Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

```
BCF    STATUS, RP0  ;Bank 0
CLRF   TMR0         ;Clear TMR0 & Prescaler
BSF    STATUS, RP0  ;Bank 1
CLRWDT              ;Clears WDT
MOVLW  b'xxxx1xxx'  ;Selects new prescale value
MOVWF  OPTION_REG   ;and assigns the prescaler to the WDT
BCF    STATUS, RP0  ;Bank 0
```

To change prescaler from the WDT to the Timer0 module use the sequence shown in Example 6-2.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT               ;Clear WDT and prescaler
BSF      STATUS, RP0 ;Bank 1
MOVLW    b'xxxx0xxx' ;Select TMR0, new prescale value and
MOVWF    OPTION_REG  ;clock source
BCF      STATUS, RP0 ;Bank 0
```

TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0

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
01h	TMR0	Timer0 module’s register								xxxx xxxx	uuuu uuuu
0Bh,8Bh,	INTCON	GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

© 1997 Microchip Technology Inc.

FIGURE 7-2: ADCON0 REGISTER (ADDRESS 1Fh), PIC16C715

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	—	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7-6: **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from an RC oscillation)

bit 5: **Unused**

bit 6-3: **CHS1:CHS0:** Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 0, (RA0/AN0)
101 = channel 1, (RA1/AN1)
110 = channel 2, (RA2/AN2)
111 = channel 3, (RA3/AN3)

bit 2: **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

If ADON = 1
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

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

bit 0: **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shutoff and consumes no operating current

FIGURE 7-3: ADCON1 REGISTER, PIC16C710/71/711 (ADDRESS 88h), PIC16C715 (ADDRESS 9Fh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	PCFG1	PCFG0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7-2: **Unimplemented:** Read as '0'

bit 1-0: **PCFG1:PCFG0**: A/D Port Configuration Control bits

PCFG1:PCFG0	RA1 & RA0	RA2	RA3	VREF
00	A	A	A	VDD
01	A	A	VREF	RA3
10	A	D	D	VDD
11	D	D	D	VDD

A = Analog input

D = Digital I/O

© 1997 Microchip Technology Inc.

7.4 A/D Conversions

Example 7-2 shows how to perform an A/D conversion. The RA pins are configured as analog inputs. The analog reference (VREF) is the device VDD. The A/D interrupt is enabled, and the A/D conversion clock is FRC. The conversion is performed on the RA0 pin (channel 0).

Note:	The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The ADRES register will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES register will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, an acquisition is automatically started on the selected channel.

EXAMPLE 7-2: A/D CONVERSION

```
     BSF     STATUS, RP0         ; Select Bank 1
     CLRF    ADCON1              ; Configure A/D inputs
     BCF     STATUS, RP0         ; Select Bank 0
     MOVLW   0xC1                ; RC Clock, A/D is on, Channel 0 is selected
     MOVWF   ADCON0              ;
     BSF     INTCON, ADIE        ; Enable A/D Interrupt
     BSF     INTCON, GIE         ; Enable all interrupts
;
;  Ensure that the required sampling time for the selected input channel has elapsed.
;  Then the conversion may be started.
;
     BSF     ADCON0, GO          ; Start A/D Conversion
       :                         ; The ADIF bit will be set and the GO/DONE bit
       :                         ;  is cleared upon completion of the A/D Conversion.
```


© 1997 Microchip Technology Inc.

TABLE 7-3: REGISTERS/BITS ASSOCIATED WITH A/D, PIC16C710/71/711

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
0Bh,8Bh	INTCON	GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
89h	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
08h	ADCON0	ADCS1	ADCS0	—	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADIF	ADON	00-0 0000	00-0 0000
88h	ADCON1	—	—	—	—	—	—	PCFG1	PCFG0	---- --00	---- --00
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	---x 0000	---u 0000
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.

TABLE 7-4: REGISTERS/BITS ASSOCIATED WITH A/D, PIC16C715

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
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	—	—	—	—	-0-- ----	-0-- ----
8Ch	PIE1	—	ADIE	—	—	—	—	—	—	-0-- ----	-0-- ----
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON 0	ADCS 1	ADCS 0	CHS2	CHS1	CHS0	GO/ $\overline{\text{DONE}}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON 1	—	—	—	—	—	—	PCFG1	PCFG0	---- --00	---- --00
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	---x 0000	---u 0000
85h	TRISA	—	—	—	TRISA4	TRISA 3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.

© 1997 Microchip Technology Inc.

TABLE 8-7: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C71

$\overline{TO}$	$\overline{PD}$	
1	1	Power-on Reset
0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
0	1	WDT Reset
0	0	WDT Wake-up
u	u	$\overline{MCLR}$ Reset during normal operation
1	0	$\overline{MCLR}$ Reset during SLEEP or interrupt wake-up from SLEEP

TABLE 8-8: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C710/711

$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	x	x	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{MCLR}$ Reset during normal operation
1	1	1	0	$\overline{MCLR}$ Reset during SLEEP or interrupt wake-up from SLEEP

TABLE 8-9: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C715

$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	
1	0	x	1	1	Power-on Reset
x	0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
x	0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	1	0	x	x	Brown-out Reset
1	1	1	0	1	WDT Reset
1	1	1	0	0	WDT Wake-up
1	1	1	u	u	$\overline{MCLR}$ Reset during normal operation
1	1	1	1	0	$\overline{MCLR}$ Reset during SLEEP or interrupt wake-up from SLEEP
0	1	1	1	1	Parity Error Reset
0	0	x	x	x	Illegal, $\overline{PER}$ is set on POR
0	x	0	x	x	Illegal, $\overline{PER}$ is set on BOR

© 1997 Microchip Technology Inc.

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write register 'f'

Example

```
BCF     FLAG_REG, 7
```

Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 1 → (f<b>)

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write register 'f'

Example

```
BSF     FLAG_REG,   7
```

Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

BTFSC **Bit Test, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

Description: If bit 'b' in register 'f' is '1' then the next instruction is executed.
If bit 'b', in register 'f', is '0' then the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	NOP

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1> = 0,
PC = address TRUE
if FLAG<1>=1,
PC = address FALSE

© 1997 Microchip Technology Inc.

BTFSS **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '0' then the next instruction is executed.
If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2TCY instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	NOP

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1> = 0,
PC = address FALSE
if FLAG<1> = 1,
PC = address TRUE

CALL **Call Subroutine**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k', Push PC to Stack	Process data	Write to PC
2nd Cycle	NOP	NOP	NOP	NOP

Example

```
HERE    CALL    THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE+1

© 1997 Microchip Technology Inc.

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

11	1010	kkkk	kkkk

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example:

```
XORLW   0xAF
```

Before Instruction

W = 0xB5

After Instruction

W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

00	0110	dfff	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to dest

Example

```
XORWF   REG  1
```

Before Instruction

REG = 0xAF
W = 0xB5

After Instruction

REG = 0x1A
W = 0xB5

© 1997 Microchip Technology Inc.

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PIC16/17. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

10.11 Software Simulator (MPLAB-SIM)

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

10.12 C Compiler (MPLAB-C)

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the MPLAB IDE memory display.

10.13 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP, edition for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

10.14 MP-DriveWay™ – Application Code Generator

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PIC16/17 device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

10.15 SEEVAL® Evaluation and Programming System

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

10.16 KEELOQ® Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

© 1997 Microchip Technology Inc.

Applicable Devices	710	71	711	715

11.4 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

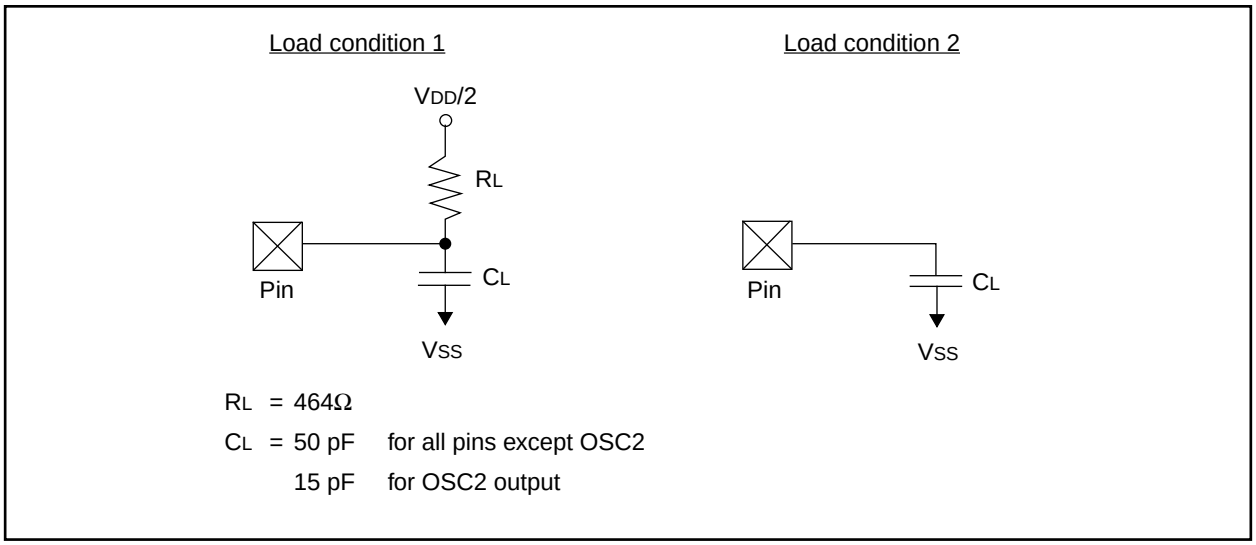
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance

FIGURE 11-1: LOAD CONDITIONS



© 1997 Microchip Technology Inc.

Applicable Devices	710	71	711	715

TABLE 11-6: A/D CONVERTER CHARACTERISTICS:
PIC16C710/711-04 (COMMERCIAL, INDUSTRIAL, EXTENDED)
PIC16C710/711-10 (COMMERCIAL, INDUSTRIAL, EXTENDED)
PIC16C710/711-20 (COMMERCIAL, INDUSTRIAL, EXTENDED)
PIC16LC710/711-04 (COMMERCIAL, INDUSTRIAL, EXTENDED)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	—	—	8-bits	bit	VREF = VDD, VSS ≤ AIN ≤ VREF
A02	EABS	Absolute error	—	—	< ± 1	LSb	VREF = VDD, VSS ≤ AIN ≤ VREF
A03	EIL	Integral linearity error	—	—	< ± 1	LSb	VREF = VDD, VSS ≤ AIN ≤ VREF
A04	EDL	Differential linearity error	—	—	< ± 1	LSb	VREF = VDD, VSS ≤ AIN ≤ VREF
A05	EFS	Full scale error	—	—	< ± 1	LSb	VREF = VDD, VSS ≤ AIN ≤ VREF
A06	EOFF	Offset error	—	—	< ± 1	LSb	VREF = VDD, VSS ≤ AIN ≤ VREF
A10	—	Monotonicity	—	guaranteed	—	—	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 impedance of analog voltage source	—	—	10.0	kΩ	
A40	IAD	A/D conversion current (VDD)	—	180	—	µA	Average current consumption when A/D is on. (Note 1)
A50	IREF	VREF input current (Note 2)	10 —	— —	1000 10	µA µA	During VAIN acquisition. Based on differential of VHOLD to VAIN. To charge CHOLD see Section 7.1. During A/D Conversion cycle

* 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.

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.

Applicable Devices	710	71	711	715

FIGURE 12-29: TYPICAL IDD vs. FREQUENCY (HS MODE, 25°C)

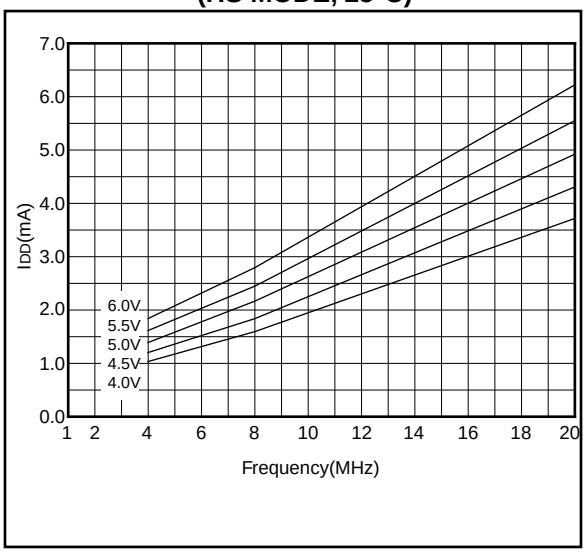
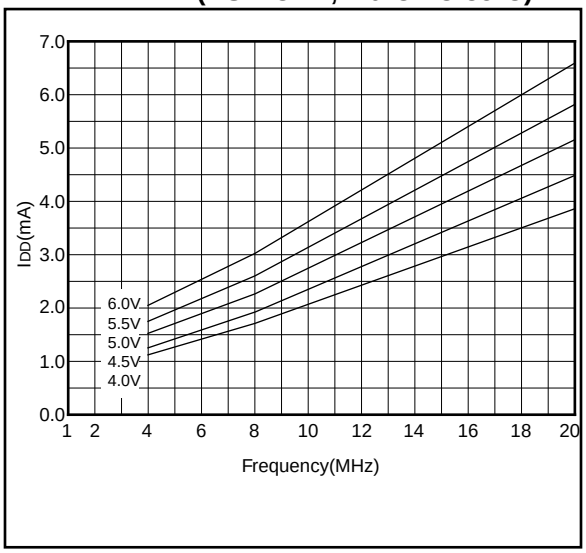


FIGURE 12-30: MAXIMUM IDD vs. FREQUENCY (HS MODE, -40°C TO 85°C)



© 1997 Microchip Technology Inc.

Applicable Devices	710	71	711	715

13.4 Timing Parameter Symbology

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

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

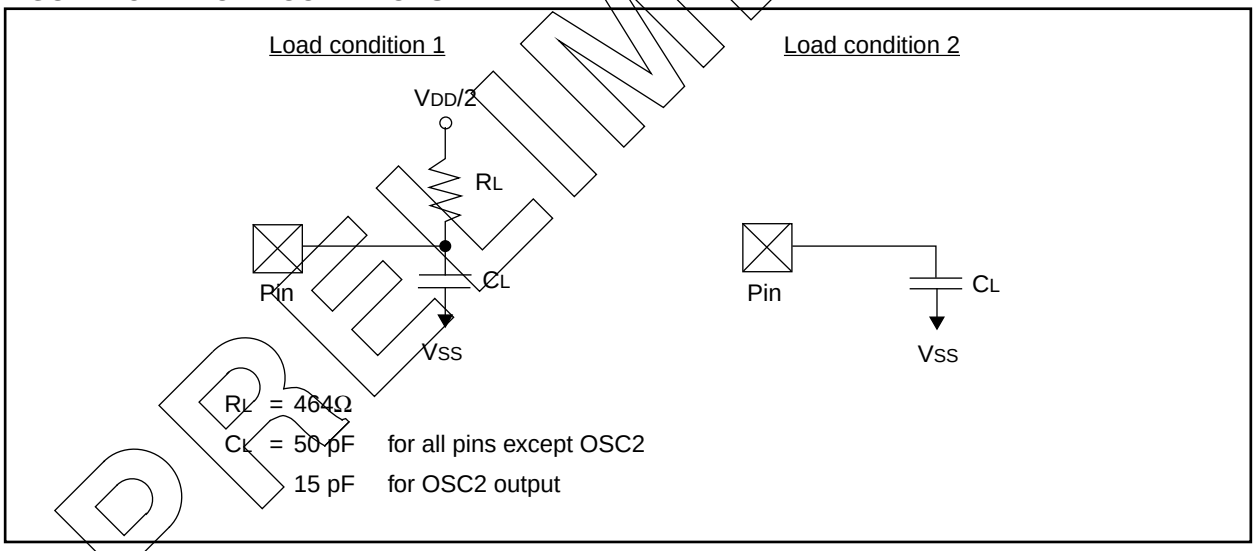
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance

FIGURE 13-1: LOAD CONDITIONS



PRELIMINARY

Applicable Devices	710	71	711	715

FIGURE 14-22: TYPICAL XTAL STARTUP TIME vs. VDD (LP MODE, 25°C)

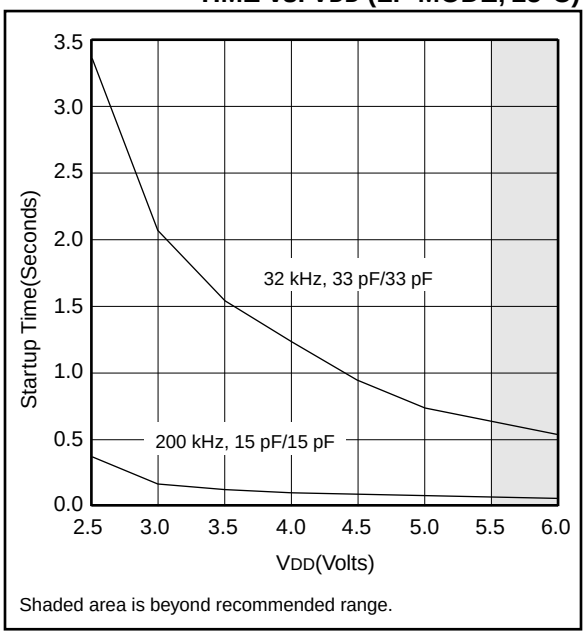


FIGURE 14-23: TYPICAL XTAL STARTUP TIME vs. VDD (HS MODE, 25°C)

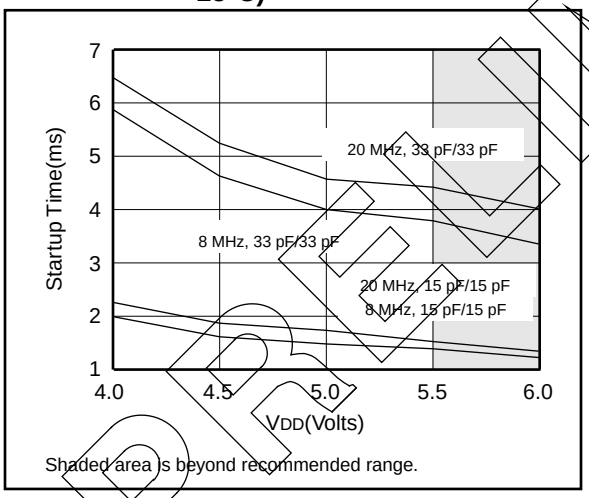


FIGURE 14-24: TYPICAL XTAL STARTUP TIME vs. VDD (XT MODE, 25°C)

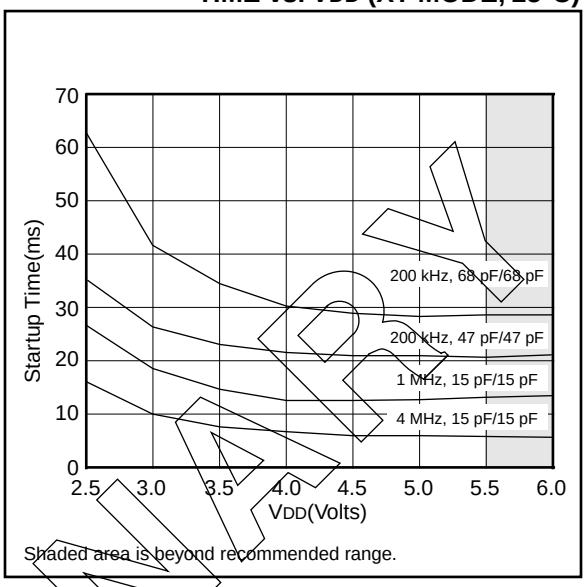


TABLE 14-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATORS

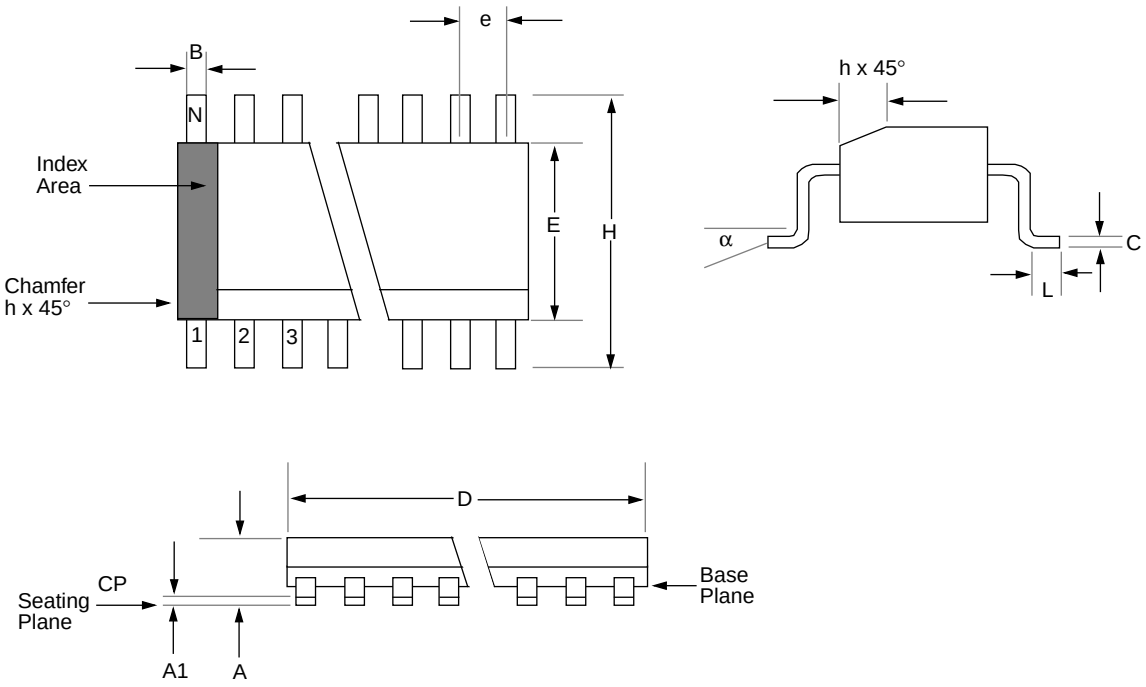
Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

Crystals Used		
32 kHz	Epson C-001R32.768K-A	± 20 PPM
200 kHz	STD XTL 200.000KHz	± 20 PPM
1 MHz	ECS ECS-10-13-1	± 50 PPM
4 MHz	ECS ECS-40-20-1	± 50 PPM
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM

Preliminary

© 1997 Microchip Technology Inc.

17.3 18-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body)(SO)



Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	11.353	11.735		0.447	0.462	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	**Reference**	0.050	0.050	**Reference**
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	18	18		18	18	
CP	–	0.102		–	0.004	

INDEX

A

A/D
- Accuracy/Error 44
- ADIF bit 39
- Analog Input Model Block Diagram 40
- Analog-to-Digital Converter 37
- Configuring Analog Port Pins 41
- Configuring the Interrupt 39
- Configuring the Module 39
- Connection Considerations 44
- Conversion Clock 41
- Conversion Time 43
- Conversions 42
- Converter Characteristics 99, 122, 145
- Delays 40
- Effects of a Reset 44
- Equations 40
- Faster Conversion - Lower Resolution Trade-off 43
- Flowchart of A/D Operation 45
- GO/$\overline{\text{DONE}}$ bit 39
- Internal Sampling Switch (Rss) Impedence 40
- Minimum Charging Time 40
- Operation During Sleep 44
- Sampling Requirements 40
- Source Impedence 40
- Time Delays 40
- Transfer Function 45

Absolute Maximum Ratings 89, 111, 135

AC Characteristics
- PIC16C710 101
- PIC16C711 101
- PIC16C715 125

ADCON0 Register 37

ADCON1 37

ADCON1 Register 14, 37

ADCS0 bit 37

ADCS1 bit 37

ADIE bit 19, 20

ADIF bit 21, 37

ADON bit 37

ADRES Register 15, 37, 39

ALU 7

Application Notes
- AN546 37
- AN552 27
- AN556 23
- AN607, Power-up Trouble Shooting 53

Architecture
- Harvard 7
- Overview 7
- von Neumann 7

Assembler
- MPASM Assembler 86

B

Block Diagrams
- Analog Input Model 40
- On-Chip Reset Circuit 52
- PIC16C71X 8
- RA3/RA0 Port Pins 25
- RA4/T0CKI Pin 25
- RB3:RB0 Port Pins 27
- RB7:RB4 Pins 28
- RB7:RB4 Port Pins 28
- Timer0 31
- Timer0/WDT Prescaler 34
- Watchdog Timer 65

BODEN bit 48

$\overline{\text{BOR}}$ bit 22, 54

Brown-out Reset (BOR) 53

C

C bit 17

C16C71 47

Carry bit 7

CHS0 bit 37

CHS1 bit 37

Clocking Scheme 10

Code Examples
- Call of a Subroutine in Page 1 from Page 0 24
- Changing Prescaler (Timer0 to WDT) 35
- Changing Prescaler (WDT to Timer0) 35
- Doing an A/D Conversion 42
- I/O Programming 30
- Indirect Addressing 24
- Initializing PORTA 25
- Initializing PORTB 27
- Saving STATUS and W Registers in RAM 64

Code Protection 47, 67

Computed GOTO 23

Configuration Bits 47

CP0 bit 47, 48

CP1 bit 48

D

DC bit 17

DC Characteristics 147
- PIC16C71 136
- PIC16C710 ..... 90, 101
- PIC16C711 ..... 90, 101
- PIC16C715 ..... 113, 125

Development Support 3, 85

Development Tools 85

Diagrams - See Block Diagrams

Digit Carry bit 7

Direct Addressing 24

E

Electrical Characteristics
- PIC16C71 135
- PIC16C710 89
- PIC16C711 89
- PIC16C715 111

External Brown-out Protection Circuit 60

External Power-on Reset Circuit 60

F

Family of Devices
- PIC16C71X 4

FOSC0 bit 47, 48

FOSC1 bit 47, 48

FSR Register 15, 16, 24

Fuzzy Logic Dev. System (*fuzzy*TECH®-MP) 87

G

General Description 3

GIE bit 19, 61

GO/$\overline{\text{DONE}}$ bit 37

© 1997 Microchip Technology Inc.

LIST OF TABLES

Table 1-1: PIC16C71X Family of Devices 4
Table 3-1: PIC16C710/71/711/715 Pinout Description 9
Table 4-1: PIC16C710/71/711 Special Function Register Summary 14
Table 4-2: PIC16C715 Special Function Register Summary 15
Table 5-1: PORTA Functions 26
Table 5-2: Summary of Registers Associated with PORTA 26
Table 5-3: PORTB Functions 28
Table 5-4: Summary of Registers Associated with PORTB 29
Table 6-1: Registers Associated with Timer0 35
Table 7-1: TAD vs. Device Operating Frequencies, PIC16C71 41
Table 7-2: TAD vs. Device Operating Frequencies, PIC16C710/711, PIC16C715 41
Table 7-3: Registers/Bits Associated with A/D, PIC16C710/71/711 46
Table 7-4: Registers/Bits Associated with A/D, PIC16C715 46
Table 8-1: Ceramic Resonators, PIC16C71 49
Table 8-2: Capacitor Selection For Crystal Oscillator, PIC16C71 49
Table 8-3: Ceramic Resonators, PIC16C710/711/715 50
Table 8-4: Capacitor Selection for Crystal Oscillator, PIC16C710/711/715 50
Table 8-5: Time-out in Various Situations, PIC16C71 54
Table 8-6: Time-out in Various Situations, PIC16C710/711/715 54
Table 8-7: Status Bits and Their Significance, PIC16C71 55
Table 8-8: Status Bits and Their Significance, PIC16C710/711 55
Table 8-9: Status Bits and Their Significance, PIC16C715 55
Table 8-10: Reset Condition for Special Registers, PIC16C710/71/711 56
Table 8-11: Reset Condition for Special Registers, PIC16C715 56
Table 8-12: Initialization Conditions For All Registers, PIC16C710/71/711 57
Table 8-13: Initialization Conditions for All Registers, PIC16C715 58
Table 9-1: Opcode Field Descriptions 69
Table 9-2: PIC16CXX Instruction Set 70
Table 10-1: Development Tools From Microchip 88
Table 11-1: Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) 89
Table 11-2: External Clock Timing Requirements 95
Table 11-3: CLKOUT and I/O Timing Requirements 96
Table 11-4: Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer, and Brown-out Reset Requirements 97
Table 11-5: Timer0 External Clock Requirements 98
Table 11-6: A/D Converter Characteristics: PIC16C710/711-04 (Commercial, Industrial, Extended) PIC16C710/711-10 (Commercial, Industrial, Extended) PIC16C710/711-20 (Commercial, Industrial, Extended) PIC16LC710/711-04 (Commercial, Industrial, Extended) 99
Table 11-7: A/D Conversion Requirements 100
Table 12-1: RC Oscillator Frequencies 107
Table 12-2: Capacitor Selection for Crystal Oscillators 108
Table 13-1: Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) 112
Table 13-2: Clock Timing Requirements 118
Table 13-3: CLKOUT and I/O Timing Requirements 119
Table 13-4: Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer, and Brown-out Reset Requirements 120
Table 13-5: Timer0 Clock Requirements 121
Table 13-6: A/D Converter Characteristics: PIC16C715-04 (Commercial, Industrial, Extended) PIC16C715-10 (Commercial, Industrial, Extended) PIC16C715-20 (Commercial, Industrial, Extended) 122
Table 13-7: A/D Converter Characteristics: PIC16LC715-04 (Commercial, Industrial) 123
Table 13-8: A/D Conversion Requirements 124
Table 14-1: RC Oscillator Frequencies 131
Table 14-2: Capacitor Selection for Crystal Oscillators 132
Table 15-1: Cross Reference of Device Specs for Oscillator Configurations and Frequencies of Operation (Commercial Devices) 135
Table 15-2: External Clock Timing Requirements 141
Table 15-3: CLKOUT and I/O Timing Requirements 142
Table 15-4: Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Requirements 143
Table 15-5: Timer0 External Clock Requirements 144
Table 15-6: A/D Converter Characteristics 145
Table 15-7: A/D Conversion Requirements 146
Table 16-1: RC Oscillator Frequencies 148

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rfPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV — DNV MSC The Netherlands Accredited by the RvA — ISO 9001 / QS-9000 REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 certified.