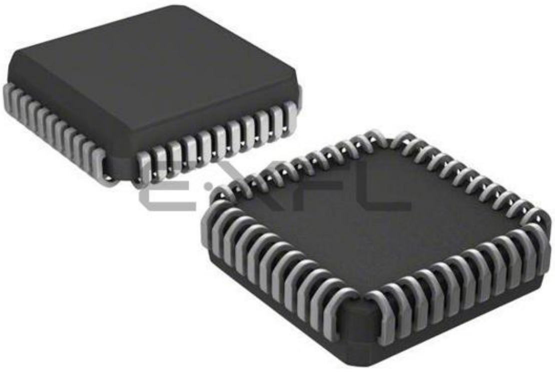
Microchip Technology - PIC16LC662-04/L Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, LED, POR, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc662-04-l

5.2 PORTB and TRISB Registers

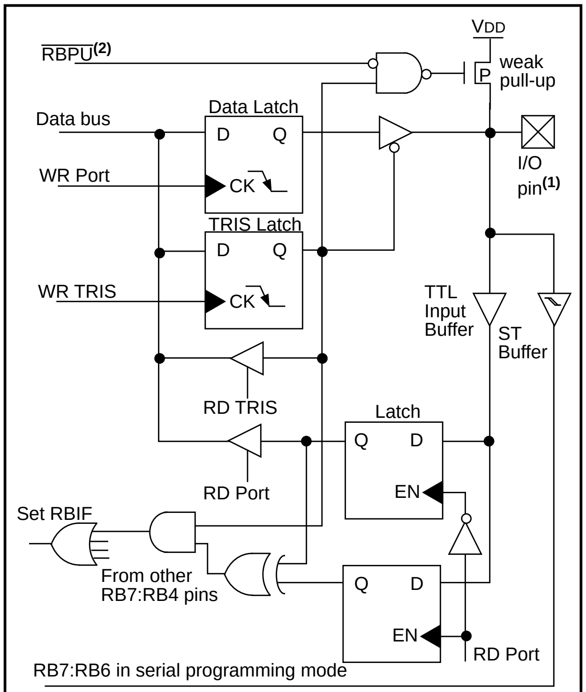
PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

Reading PORTB register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified, and then written to the port data latch.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of PORTB’s pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The “mismatch” outputs of RB7:RB4 are OR’ed together to generate the RBIF interrupt (flag latched in (INTCON<0>)).

FIGURE 5-5: BLOCK DIAGRAM OF RB7:RB4 PINS



Note 1: I/O pins have diode protection to VDD and VSS.
2: TRISB = '1' enables weak pull-up if $\overline{\text{RBPU}}$ = '0' (OPTION<7>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.

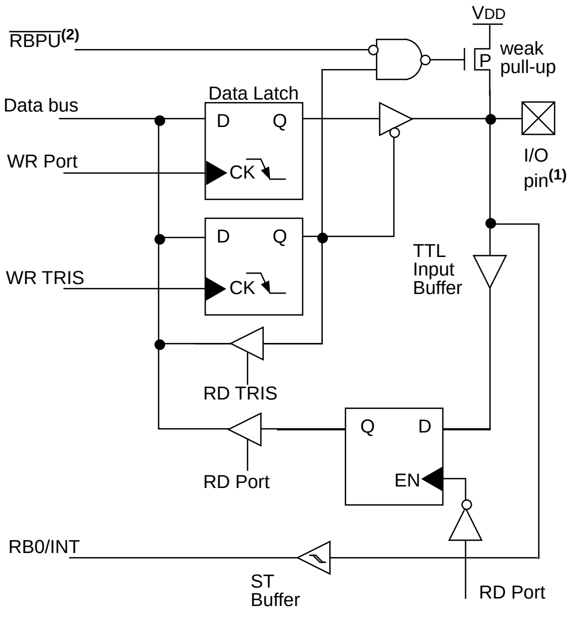
b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition, and allow flag bit RBIF to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a keypad and make it possible for wake-up on key-depression. (See AN552 in the Microchip *Embedded Control Handbook*.)

The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-6: BLOCK DIAGRAM OF RB3:RB0 PINS



Note 1: I/O pins have diode protection to VDD and VSS.
2: TRISB = '1' enables weak pull-up if $\overline{\text{RBPU}}$ = '0' (OPTION<7>).

5.5 PORTE and TRISE Register (PIC16C661 and PIC16C662 only)

PORTE has three pins RE0/$\overline{RD}$, RE1/$\overline{WR}$, and RE2/$\overline{CS}$, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). In this mode the input buffers are TTL.

Figure 5-9 shows the TRISE register, which also controls the parallel slave port operation.

FIGURE 5-9: TRISE REGISTER (ADDRESS 89h)

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
bit7							bit0

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

bit 7: **IBF:** Input Buffer Full Status bit
1 = A word has been received and waiting to be read by the CPU
0 = No word has been received

bit 6: **OBF**: Output Buffer Full Status bit
1 = The output buffer still holds a previously written word
0 = The output buffer has been read

bit 5: **IBOV**: Input Buffer Overflow Detect bit (in microprocessor mode)
1 = A write occurred when a previously input word has not been read (must be cleared in software)
0 = No overflow occurred

bit 4: **PSPMODE**: Parallel Slave Port Mode Select bit
1 = Parallel slave port mode
0 = General purpose I/O mode

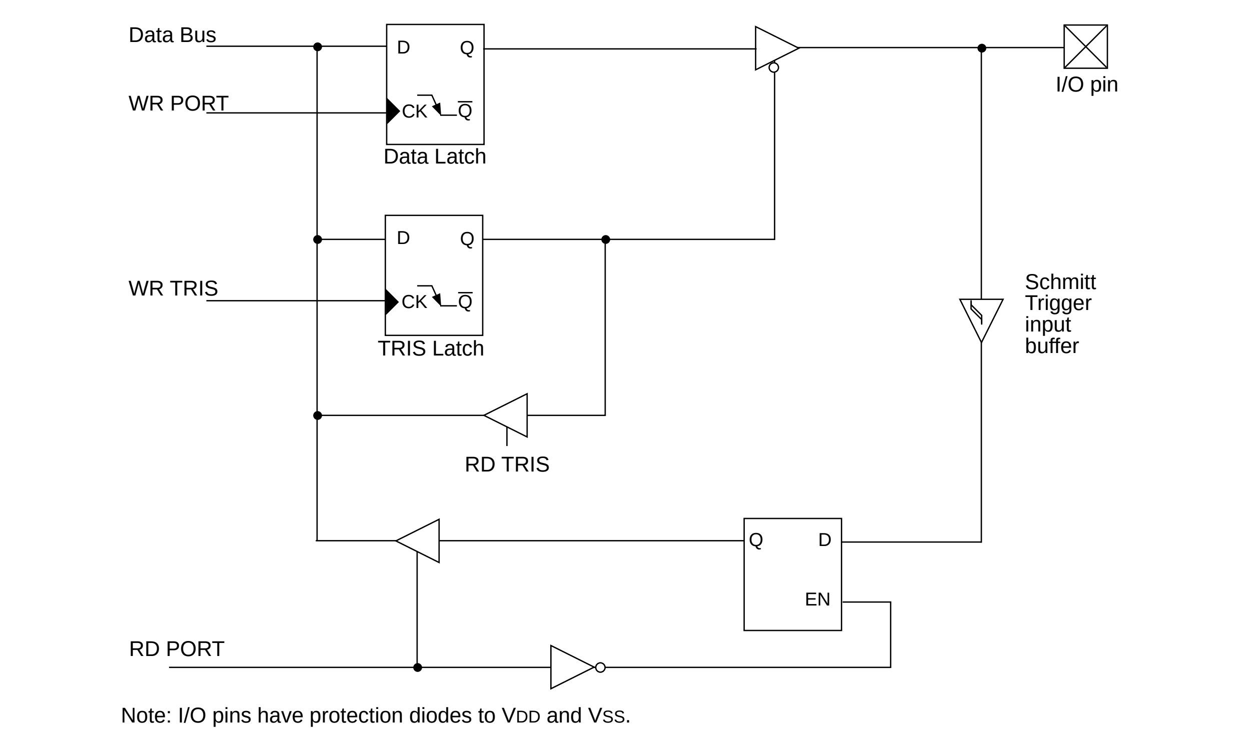
bit 3: **Unimplemented**: Read as '0'

bit 2: **TRISE2**: Direction control bit for pin RE2/$\overline{CS}$
1 = Input
0 = Output

bit 1: **TRISE1**: Direction control bit for pin RE1/$\overline{WR}$
1 = Input
0 = Output

bit 0: **TRISE0**: Direction control bit for pin RE0/$\overline{RD}$
1 = Input
0 = Output

FIGURE 5-10: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and VSS.

TABLE 5-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/$\overline{RD}$	bit0	ST/TTL(1)	Input/output port pin or read control input in parallel slave port mode: $\overline{RD}$ 1 = Not a read operation 0 = Read operation. Reads PORTD register (if chip selected)
RE1/$\overline{WR}$	bit1	ST/TTL(1)	Input/output port pin or write control input in parallel slave port mode: $\overline{WR}$ 1 = Not a write operation 0 = Write operation. Writes PORTD register (if chip selected)
RE2/$\overline{CS}$	bit2	ST/TTL(1)	Input/output port pin or chip select control input in parallel slave port mode: $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

TABLE 5-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

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
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	`---- -xxx`	`---- -uuu`
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	`0000 -111`	`0000 -111`

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

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 precaution 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'  ;Select new prescale
MOVWF  OPTION_REG   ;value & 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	PEIE	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	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

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

TABLE 7-1: REGISTERS ASSOCIATED WITH THE COMPARATOR MODULE

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
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	`00-- 0000`	`00-- 0000`
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	`000- 0000`	`000- 0000`
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000u`
0Ch	PIR1	PSPIF[(1)]	CMIF	—	—	—	—	—	—	`00-- ----`	`00-- ----`
8Ch	PIE1	PSPIE[(1)]	CMIE	—	—	—	—	—	—	`00-- ----`	`00-- ----`
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`--11 1111`	`--11 1111`

Note 1: These bits are reserved on the PIC16C641/642, always maintain these bits clear.

8.0 VOLTAGE REFERENCE MODULE

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference module is not being used.

The VRCON register, shown in Figure 8-1, controls the operation of the Voltage Reference Module. The block diagram is given in Figure 8-2.

FIGURE 8-1: VRCON REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
bit7							bit0

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

bit 7: **VREN:** VREF Enable
1 = VREF circuit powered up
0 = VREF circuit powered down, no IDD drain

bit 6: **VROE:** VREF Output Enable
1 = VREF is output on RA2 pin
0 = VREF is disconnected from RA2 pin

bit 5: **VRR:** VREF Range selection
1 = Low Range
0 = High Range

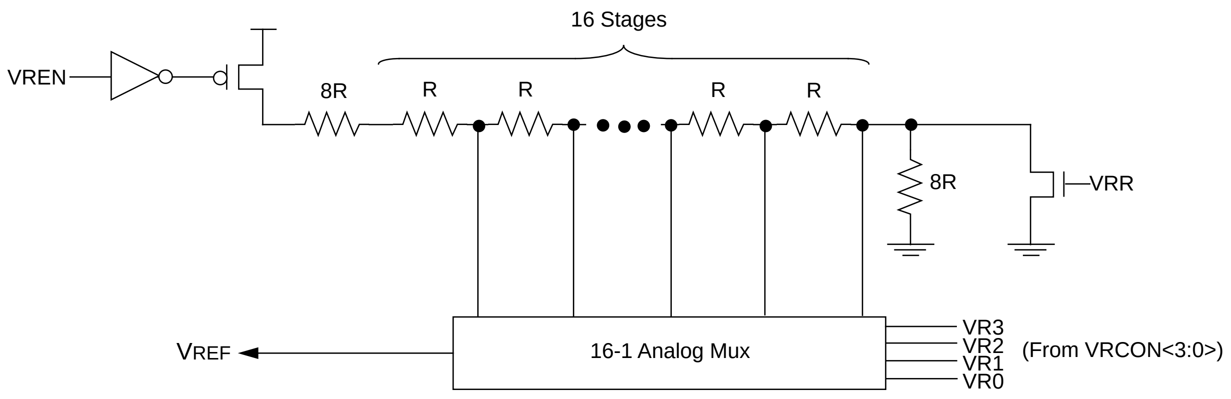
bit 4: **Unimplemented:** Read as '0'

bit 3-0: **VR3:VR0**: VREF value selection 0 ≤ VR3:VR0 ≤ 15

When: VRR = 1
Then: VREF = (VR3:VR0/ 24) • VDD

When: VRR = 0
Then: VREF = 1/4 • VDD + (VR3:VR0/ 32) • VDD

FIGURE 8-2: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: R is defined in Table 12-3.

9.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h–3FFFh), which can be accessed only during programming.

FIGURE 9-1: CONFIGURATION WORD

CP1	CP0	CP1	CP0	CP1	CP0	MPEEN	BODEN	CP1	CP0	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
bit13													bit0

CONFIG	Address
REGISTER:	2007h

bit 13-8 5-4: **CP1:CP0:** Code protection bits[2]
11 = Code protection off
10 = Upper half of program memory code protected
01 = Upper 3/4th of program memory code protected
00 = All memory is code protected

bit 7: **MPEEN**: Memory Parity Error Enable
1 = Memory Parity Checking is enabled
0 = Memory Parity Checking is disabled

bit 6: **BODEN**: Brown-out Reset Enable bit [1]
1 = BOR enabled
0 = BOR disabled

bit 3: **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit [1]
1 = PWRT disabled
0 = PWRT enabled

bit 2: **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Enabling Brown-out Reset automatically enables the Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.

2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

9.5 Interrupts

The PIC16C641 and PIC16C642 have four sources of interrupt, while the PIC16C661 and PIC16C662 have five sources:

- External interrupt RB0/INT
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB7:RB4)
- Comparator interrupt
- Parallel Slave Port interrupt (PIC16C661/662)

The interrupt control register, (INTCON), records individual core interrupt requests in flag bits. It also has various individual enable bits and the global interrupt enable bit.

The global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on reset.

The “return from interrupt” instruction, `RETFIE`, exits the interrupt routine as well as sets the GIE bit, which allows any pending interrupt to execute.

Those interrupts associated with the “core” have their flag and enable bits in the INTCON register. The core interrupts are: RB0/INT pin interrupt, the RB port change interrupt, and the TMR0 overflow interrupt. The INTCON register also contains the Peripheral Interrupt Enable bit, PEIE. Bit PEIE will enable/mask the peripheral interrupts (CM and PSP) from vectoring when bit PEIE is set/cleared.

Flag bits PSPIF and CMIF are contained in special function register PIR1. The corresponding interrupt enable bits (PSPIE and CMIE) are contained in special function register PIE1.

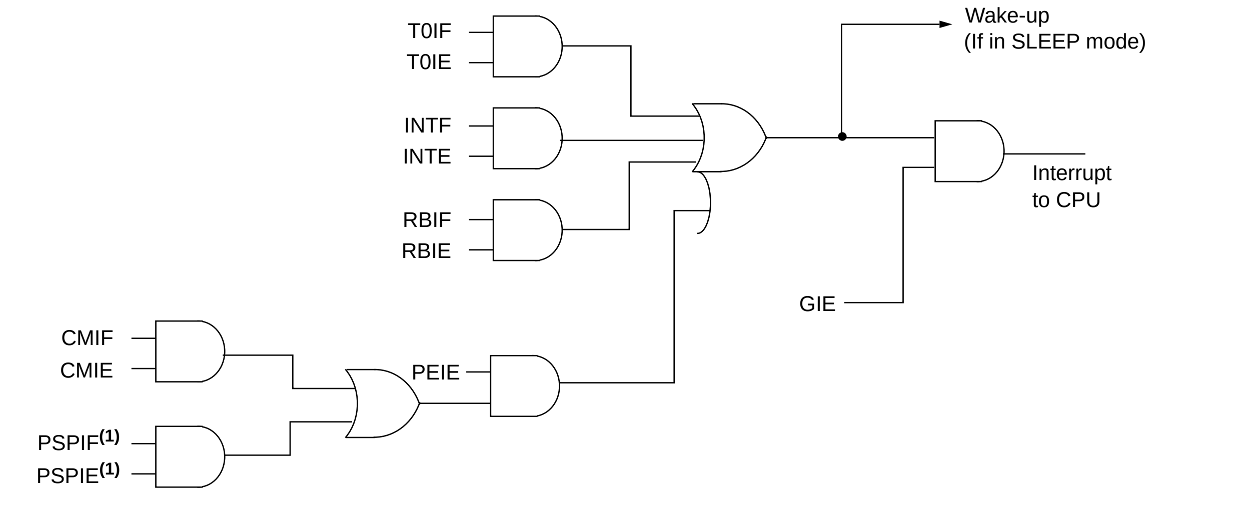
When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the RB0/INT or Port RB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 9-16). The latency is the same for one or two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The CPU will execute a NOP in the cycle immediately following the instruction which clears the GIE bit. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

FIGURE 9-15: INTERRUPT LOGIC



Note 1: The Parallel Slave Port is implemented on the PIC16C661 and PIC16C662 only.

TABLE 10-2: INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0000	0011	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

SUBLW Subtract W from Literal

Syntax: [*label*] SUBLW k

Operands: 0 ≤ k ≤ 255

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

`11`	`110x`	`kkkk`	`kkkk`

Description: The W register is subtracted (2’s complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1: `SUBLW 0x02`

Before Instruction

W = 1
C = ?

After Instruction

W = 1
C = 1; result is positive

Example 2: Before Instruction

W = 2
C = ?

After Instruction

W = 0
C = 1; result is zero

Example 3: Before Instruction

W = 3
C = ?

After Instruction

W = 0xFF
C = 0; result is negative

SUBWF Subtract W from f

Syntax: [*label*] SUBWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

`00`	`0010`	`dfff`	`ffff`

Description: Subtract (2’s complement method) W register from 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

Example 1: `SUBWF REG1,1`

Before Instruction

REG1 = 3
W = 2
C = ?

After Instruction

REG1 = 1
W = 2
C = 1; result is positive

Example 2: Before Instruction

REG1 = 2
W = 2
C = ?

After Instruction

REG1 = 0
W = 2
C = 1; result is zero

Example 3: Before Instruction

REG1 = 1
W = 2
C = ?

After Instruction

REG1 = 0xFF
W = 2
C = 0; result is negative

FIGURE 12-6: TIMER0 CLOCK TIMING

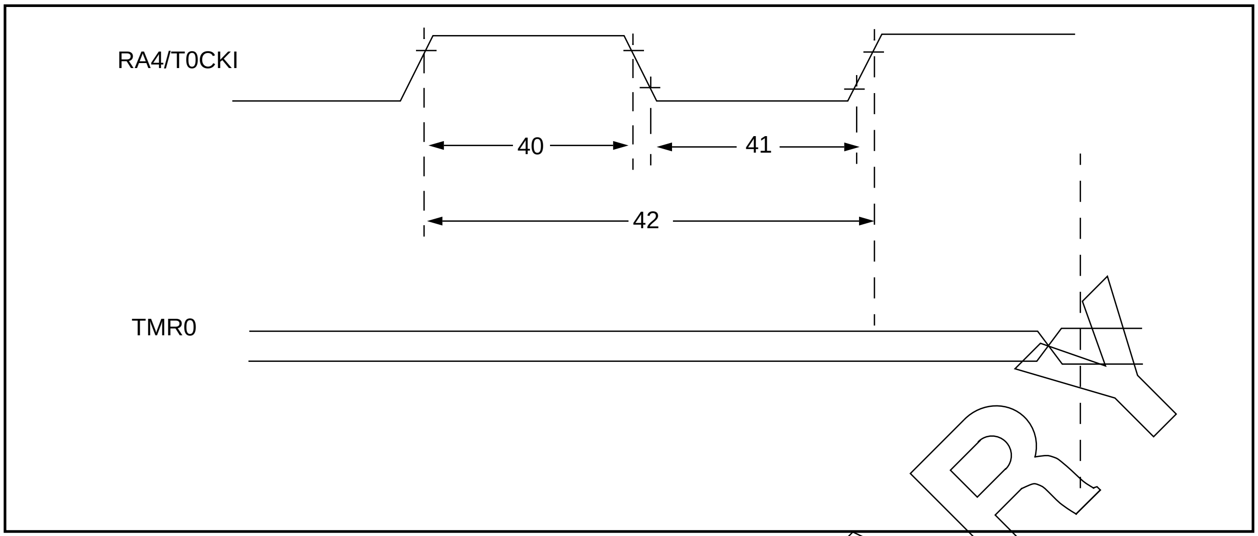


TABLE 12-7: TIMER0 CLOCK REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, …, 256)

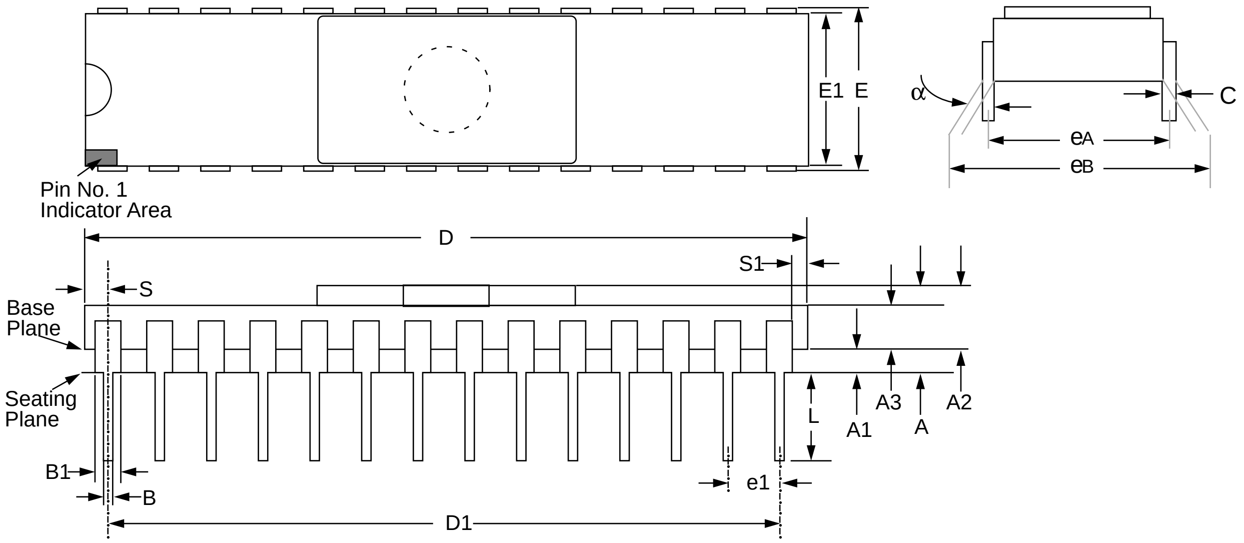
* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

13.0 DEVICE CHARACTERIZATION INFORMATION

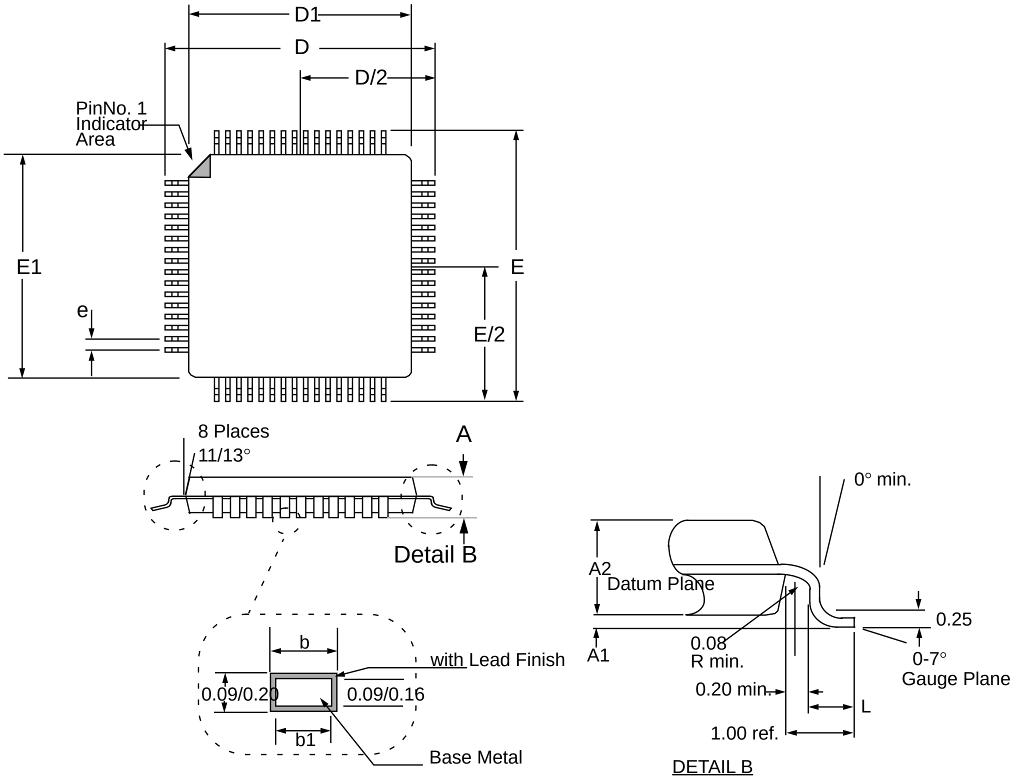
NOT AVAILABLE AT THIS TIME.

Package Type: 28-Lead Ceramic Side Brazed Dual In-Line with Window (JW) (300 mil)



Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	**Millimeters**			**Inches**		
	Min	**Max**	**Notes**	**Min**	**Max**	**Notes**
α	0°	10°		0°	10°	
A	3.937	5.030		0.155	0.198	
A1	1.016	1.524		0.040	0.060	
A2	2.921	3.506		0.115	0.138	
A3	1.930	2.388		0.076	0.094	
B	0.406	0.508		0.016	0.020	
B1	1.219	1.321	**Typical**	0.048	0.052	
C	0.228	0.305	**Typical**	0.009	0.012	
D	35.204	35.916		1.386	1.414	
D1	32.893	33.147	**BSC**	1.295	1.305	
E	7.620	8.128		0.300	0.320	
E1	7.366	7.620		0.290	0.300	
e1	2.413	2.667	**Typical**	0.095	0.105	
eA	7.366	7.874	**BSC**	0.290	0.310	
eB	7.594	8.179		0.299	0.322	
L	3.302	4.064		0.130	0.160	
S	1.143	1.397		0.045	0.055	
S1	0.533	0.737		0.021	0.029	

Package Type: 44-Lead Thin Plastic Quad Flatpack (PT/TQ) - 10x10x1 mm Body 1.0/0.10 mm Lead Form



Package Group: Plastic TQFP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	7°		0°	7°	
A	—	1.200		—	0.047	
A1	0.050	0.150		0.002	0.006	
A2	0.950	1.050		0.037	0.041	
b	0.300	0.450		0.012	0.018	
b1	0.300	0.400		0.012	0.016	
D	12.0	12.0	**BSC**	0.472	0.0472	**BSC**
D1	10.0	10.0	**BSC**	0.394	0.394	**BSC**
E	12.0	12.0	**BSC**	0.472	0.472	**BSC**
E1	10.0	10.0	**BSC**	0.394	0.394	**BSC**
e	0.8	0.8	**BSC**	0.031	0.031	**BSC**
L	0.450	0.750		0.018	0.030	

APPENDIX A: ENHANCEMENTS

The following are the list of enhancements over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bits. This allows larger page sizes both in program memory (4K now as opposed to 512 before) and register file (up to 176 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. PA2, PA1, PA0 bits are removed from STATUS register.
3. Data memory paging is slightly redefined. STATUS register is modified.
4. Four new instructions have been added: `RETURN`, `RETFIE`, `ADDLW`, and `SUBLW`.
   Two instructions `TRIS` and `OPTION` are being phased out although they are kept for compatibility with PIC16C5X.
5. `OPTION` and `TRIS` registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Six different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers, Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) are included for more reliable power-up. These timers can be invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt on change feature.
13. Timer0 clock input, T0CKI pin is also a port pin (RA4/T0CKI) and has a TRIS bit.
14. FSR is made a full 8-bit register.
15. "In-circuit programming" is made possible. The user can program PIC16CXX devices using only five pins: V_{DD}, V_{SS}, V_{PP}, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on Reset status bit ($\overline{\text{POR}}$), a Brown-out Reset status bit ($\overline{\text{BOR}}$), a Parity Error Reset ($\overline{\text{PER}}$), and a Memory Parity Enable (MPEEN) bit.
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.
18. PORTA inputs are now Schmitt Trigger inputs.
19. Brown-out Reset circuitry has been added.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for `CALL`, `GOTO`.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

E.4 PIC16C6X Family of Devices

	Clock	Memory			Peripherals				Features					
	Maximum Frequency of Operation (MHz)	Program Memory (x14 words) EPROM	Program Memory (x14 words) ROM	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I^{2}C, USART)	Parallel Slave Port	Interrupt Sources	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C62	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I^{2}C	—	7	22	3.0-6.0	Yes	—	28-pin SDIP, SOIC, SSOP
PIC16C62A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I^{2}C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16CR62(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I^{2}C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C63	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I^{2}C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16CR63(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I^{2}C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C64	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I^{2}C	Yes	8	33	3.0-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP
PIC16C64A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I^{2}C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR64(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I^{2}C	Yes	8	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16C65	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I^{2}C, USART	Yes	11	33	3.0-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP
PIC16C65A(1)	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I^{2}C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR65(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I^{2}C, USART	Yes	11	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
All PIC16C6X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

INDEX

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 (602) 786-7578.

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

To: Technical Publications Manager   Total Pages Sent ________________

RE: Reader Response

From: Name ____

Company ____

Address ____

City / State / ZIP / Country ____

Telephone: (_______) _________ - _________   FAX: (______) _________ - _________

Application (optional):

Would you like a reply?___Y ___N

Device: **PIC16C64X & PIC16C66X**   Literature Number: **DS30559A**

Questions:

1. What are the best features of this document?

____

____

2. How does this document meet your hardware and software development needs?

____

____

3. Do you find the organization of this data sheet easy to follow? If not, why?

____

____

4. What additions to the data sheet do you think would enhance the structure and subject?

____

____

5. What deletions from the data sheet could be made without affecting the overall usefulness?

____

____

6. Is there any incorrect or misleading information (what and where)?

____

____

7. How would you improve this document?

____

____

8. How would you improve our software, systems, and silicon products?

____

____

PIC16C64X & PIC16C66X PRODUCT IDENTIFICATION SYSTEM

PART NO. -XX X /XX XXX

Pattern: Special Requirements

Package:

SO	=	SOIC
L	=	PLCC
P	=	PDIP
TQ	=	TQFP
SP	=	Skinny DIP
JW	=	Windowed DIP

Temperature Range:

-	=	0°C to +70°C
I	=	-40°C to +85°C
E	=	-40°C to +125°C

Frequency Range:

04	=	4 MHz
10	=	10MHz
20	=	20 MHz

Device

Examples

a) PIC16C662-04/P Commercial Temp., PDIP Package, 4 MHz, normal VDD limits

b) PIC16C662-04I/SO Industrial Temp., SOIC package, 4 MHz, normal VDD limits

c) PIC16C662-04E/P Automotive Temp., PDIP package, 4 MHz, normal VDD limits

Please contact your local sales office for exact ordering procedures.

JW devices are UV erasable and can be programmed to any device configuration. JW devices meet the electrical requirements of each oscillator type (including LC devices).

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office (see below)
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.
For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

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

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.