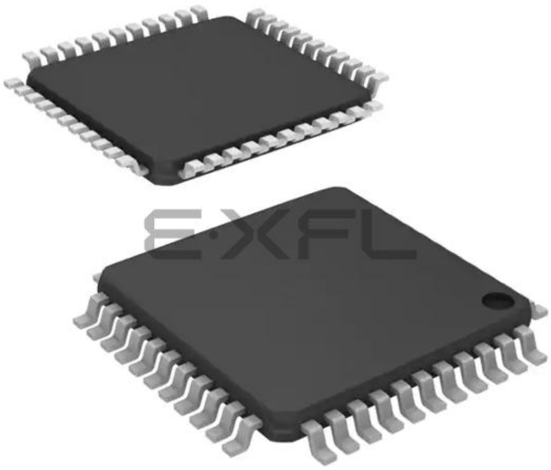


Welcome to E-XFL.COM

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	8MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	8KB (4K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17lc43-08-pt

5.3 Peripheral Interrupt Request Register (PIR)

This register contains the individual flag bits for the peripheral interrupts.

Note:	These bits will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled), or the GLINTD bit is set (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the interrupt flag to ensure that the program does not immediately branch to the peripheral interrupt service routine.

FIGURE 5-4: PIR REGISTER (ADDRESS: 16h, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R - 1	R - 0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **RBIF**: PORTB Interrupt on Change Flag bit
1 = One of the PORTB inputs changed (Software must end the mismatch condition)
0 = None of the PORTB inputs have changed

bit 6: **TMR3IF**: Timer3 Interrupt Flag bit
If Capture1 is enabled (CA1/$\overline{PR3}$ = 1)
1 = Timer3 overflowed
0 = Timer3 did not overflow

If Capture1 is disabled (CA1/$\overline{PR3}$ = 0)
1 = Timer3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value
0 = Timer3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value

bit 5: **TMR2IF**: Timer2 Interrupt Flag bit
1 = Timer2 value has rolled over to 0000h from equalling the period register (PR2) value
0 = Timer2 value has not rolled over to 0000h from equalling the period register (PR2) value

bit 4: **TMR1IF**: Timer1 Interrupt Flag bit
If Timer1 is in 8-bit mode (T16 = 0)
1 = Timer1 value has rolled over to 0000h from equalling the period register (PR) value
0 = Timer1 value has not rolled over to 0000h from equalling the period register (PR2) value

If Timer1 is in 16-bit mode (T16 = 1)
1 = TMR1:TMR2 value has rolled over to 0000h from equalling the period register (PR1:PR2) value
0 = TMR1:TMR2 value has not rolled over to 0000h from equalling the period register (PR1:PR2) value

bit 3: **CA2IF**: Capture2 Interrupt Flag bit
1 = Capture event occurred on RB1/CAP2 pin
0 = Capture event did not occur on RB1/CAP2 pin

bit 2: **CA1IF**: Capture1 Interrupt Flag bit
1 = Capture event occurred on RB0/CAP1 pin
0 = Capture event did not occur on RB0/CAP1 pin

bit 1: **TXIF**: USART Transmit Interrupt Flag bit
1 = Transmit buffer is empty
0 = Transmit buffer is full

bit 0: **RCIF**: USART Receive Interrupt Flag bit
1 = Receive buffer is full
0 = Receive buffer is empty

NOTES:

TABLE 6-3: SPECIAL FUNCTION REGISTERS (Cont.'d)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (3)
Bank 2											
10h	TMR1	Timer1								xxxx xxxx	uuuu uuuu
11h	TMR2	Timer2								xxxx xxxx	uuuu uuuu
12h	TMR3L	TMR3 register; low byte								xxxx xxxx	uuuu uuuu
13h	TMR3H	TMR3 register; high byte								xxxx xxxx	uuuu uuuu
14h	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
16h	PR3L/CA1L	Timer3 period register, low byte/capture1 register; low byte								xxxx xxxx	uuuu uuuu
17h	PR3H/CA1H	Timer3 period register, high byte/capture1 register; high byte								xxxx xxxx	uuuu uuuu
Bank 3											
10h	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
14h	CA2L	Capture2 low byte								xxxx xxxx	uuuu uuuu
15h	CA2H	Capture2 high byte								xxxx xxxx	uuuu uuuu
16h	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
Unbanked											
18h (5)	PRODL	Low Byte of 16-bit Product (8 x 8 Hardware Multiply)								xxxx xxxx	uuuu uuuu
19h (5)	PRODH	High Byte of 16-bit Product (8 x 8 Hardware Multiply)								xxxx xxxx	uuuu uuuu

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

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<15:8> whose contents are updated from or transferred to the upper byte of the program counter.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in CPUSTA are not affected by a MCLR reset.

3: Other (non power-up) resets include: external reset through MCLR and the Watchdog Timer Reset.

4: The following values are for both TBLPTRL and TBLPTRH:
All PIC17C4X devices (Power-on Reset 0000 0000) and (All other resets 0000 0000)
except the PIC17C42 (Power-on Reset xxxx xxxx) and (All other resets uuuu uuuu)

5: The PRODL and PRODH registers are not implemented on the PIC17C42.

6.3 Stack Operation

The PIC17C4X devices have a 16 x 16-bit wide hardware stack (Figure 6-1). The stack is not part of either the program or data memory space, and the stack pointer is neither readable nor writable. The PC is "PUSHed" onto the stack when a `CALL` instruction is executed or an interrupt is acknowledged. The stack is "POPed" in the event of a `RETURN`, `RETLW`, or a `RETFIE` instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer, with the stack pointer initialized to '0' after all resets. There is a stack available bit (STKAV) to allow software to ensure that the stack has not overflowed. The STKAV bit is set after a device reset. When the stack pointer equals Fh, STKAV is cleared. When the stack pointer rolls over from Fh to 0h, the STKAV bit will be held clear until a device reset.

Note 1: There is not a status bit for stack underflow. The STKAV bit can be used to detect the underflow which results in the stack pointer being at the top of stack.

Note 2: There are no instruction mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW`, and `RETFIE` instructions, or the vectoring to an interrupt vector.

Note 3: After a reset, if a "POP" operation occurs before a "PUSH" operation, the STKAV bit will be cleared. This will appear as if the stack is full (underflow has occurred). If a "PUSH" operation occurs next (before another "POP"), the STKAV bit will be locked clear. Only a device reset will cause this bit to set.

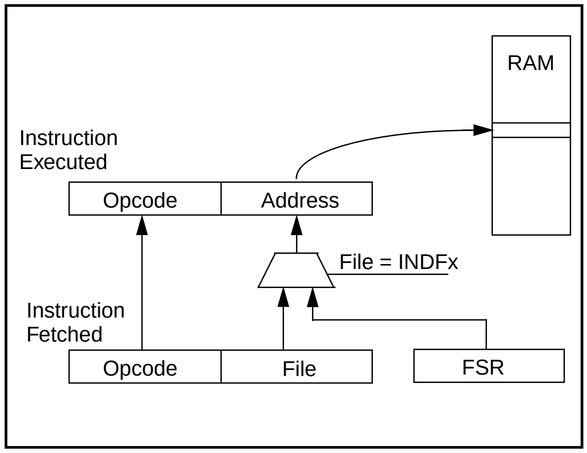
After the device is "PUSHed" sixteen times (without a "POP"), the seventeenth push overwrites the value from the first push. The eighteenth push overwrites the second push (and so on).

6.4 Indirect Addressing

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. That is, the register that is to be read or written can be modified by the program. This can be useful for data tables in the data memory. Figure 6-10 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register.

Example 6-1 shows the use of indirect addressing to clear RAM in a minimum number of instructions. A similar concept could be used to move a defined number of bytes (block) of data to the USART transmit register (TXREG). The starting address of the block of data to be transmitted could easily be modified by the program.

FIGURE 6-10: INDIRECT ADDRESSING



7.2 Table Writes to External Memory

Table writes to external memory are always two-cycle instructions. The second cycle writes the data to the external memory location. The sequence of events for an external memory write are the same for an internal write.

> **Note:** If an interrupt is pending or occurs during the TABLWT, the two cycle table write completes. The RA0/INT, TMR0, or T0CKI interrupt flag is automatically cleared or the pending peripheral interrupt is acknowledged.

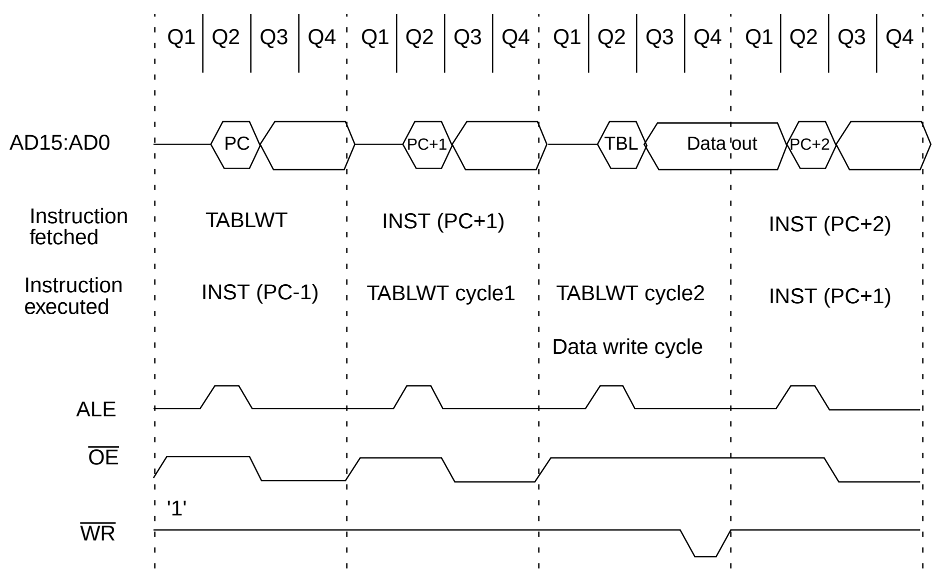
7.2.2 TABLE WRITE CODE

The "i" operand of the TABLWT instruction can specify that the value in the 16-bit TBLPTR register is automatically incremented for the next write. In Example 7-1, the TBLPTR register is not automatically incremented.

EXAMPLE 7-1: TABLE WRITE

```
CLRWDT                     ; Clear WDT
MOVLW   HIGH (TBL_ADDR) ; Load the Table
MOVWF   TBLPTRH          ;   address
MOVLW   LOW (TBL_ADDR)  ;
MOVWF   TBLPTRL          ;
MOVLW   HIGH (DATA)      ; Load HI byte
TLWT    1, WREG          ;   in TABLATCH
MOVLW   LOW (DATA)       ; Load LO byte
TABLWT  0,0,WREG         ;   in TABLATCH
                         ;   and write to
                         ;   program memory
                         ;   (Ext. SRAM)
```

FIGURE 7-5: TABLWT WRITE TIMING (EXTERNAL MEMORY)



Note: If external write GLINTD = '1', Enable bit = '1', '1' → Flag bit, Do table write. The highest pending interrupt is cleared.

12.2.3 EXTERNAL CLOCK INPUT FOR TIMER3

When TMR3CS is set, the 16-bit TMR3 increments on the falling edge of clock input TCLK3. The input on the RB5/TCLK3 pin is sampled and synchronized by the internal phase clocks twice every instruction cycle. This causes a delay from the time a falling edge appears on TCLK3 to the time TMR3 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section. Figure 12-9 shows the timing diagram when operating from an external clock.

12.2.4 READING/WRITING TIMER3

Since Timer3 is a 16-bit timer and only 8-bits at a time can be read or written, care should be taken when reading or writing while the timer is running. The best method to read or write the timer is to stop the timer, perform any read or write operation, and then restart Timer3 (using the TMR3ON bit). However, if it is necessary to keep Timer3 free-running, care must be taken. For writing to the 16-bit TMR3, Example 12-2 may be used. For reading the 16-bit TMR3, Example 12-3 may be used. Interrupts must be disabled during this routine.

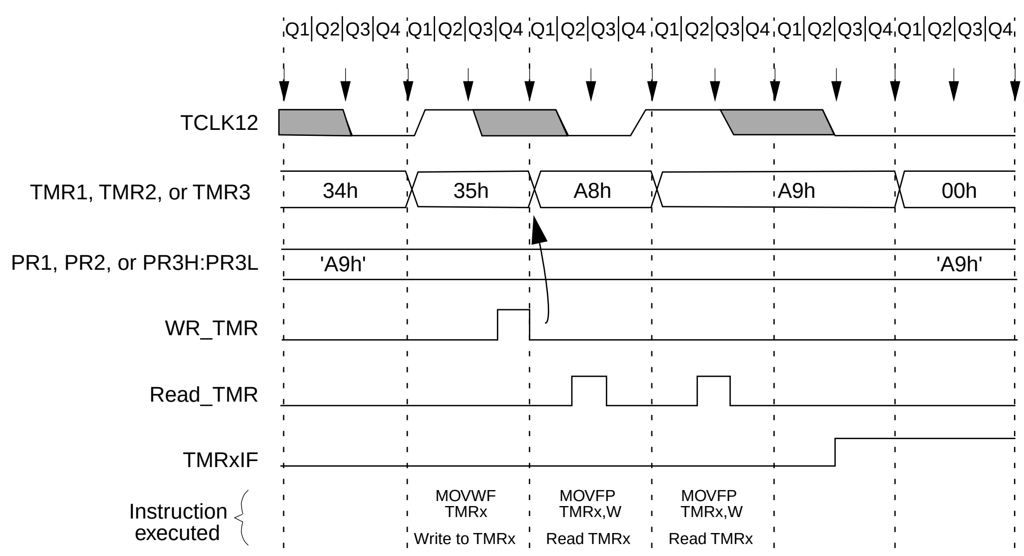
EXAMPLE 12-2: WRITING TO TMR3

```
BSF   CPUSTA, GLINTD ;Disable interrupt
MOVFP RAM_L,  TMR3L  ;
MOVFP RAM_H,  TMR3H  ;
BCF   CPUSTA, GLINTD ;Done,enable interrupt
```

EXAMPLE 12-3: READING FROM TMR3

```
MOVPF   TMR3L, TMPLO    ;read low tmr0
MOVPF   TMR3H, TMPHI    ;read high tmr0
MOVFP   TMPLO, WREG     ;tmplo —> wreg
CPFSLT  TMR3L, WREG     ;tmr0l < wreg?
RETURN                  ;no then return
MOVPF   TMR3L, TMPLO    ;read low tmr0
MOVPF   TMR3H, TMPHI    ;read high tmr0
RETURN                  ;return
```

FIGURE 12-9: TMR1, TMR2, AND TMR3 OPERATION IN EXTERNAL CLOCK MODE



Note 1: TCLK12 is sampled in Q2 and Q4.
2: ↓ indicates a sampling point.
3: The latency from TCLK12 ↓ to timer increment is between 2Tosc and 6Tosc.

FIGURE 13-3: USART TRANSMIT

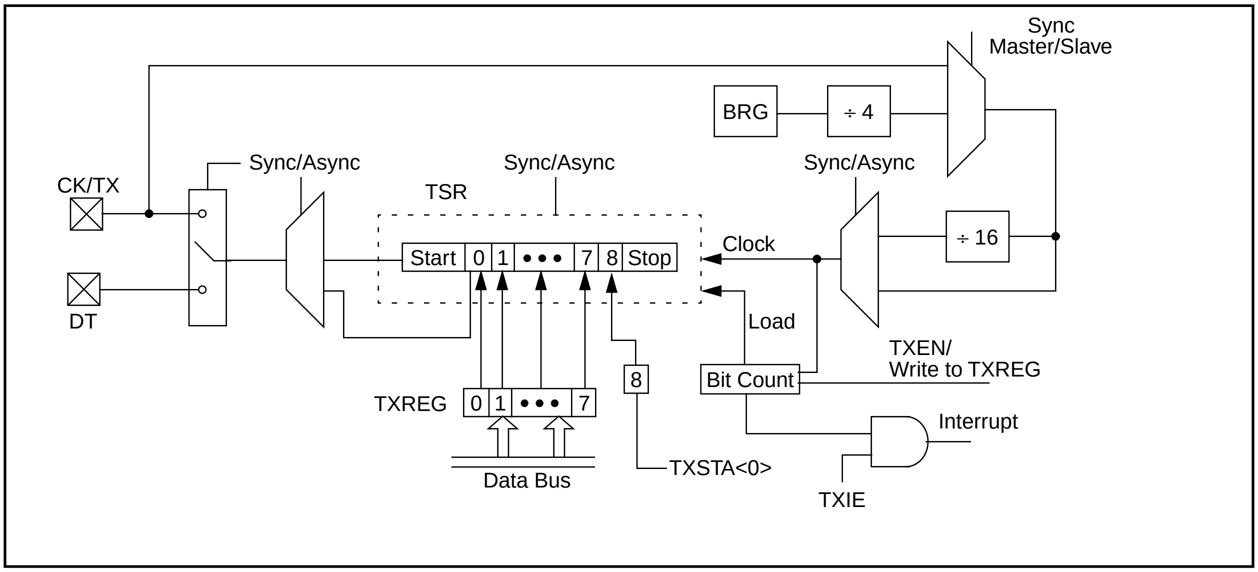
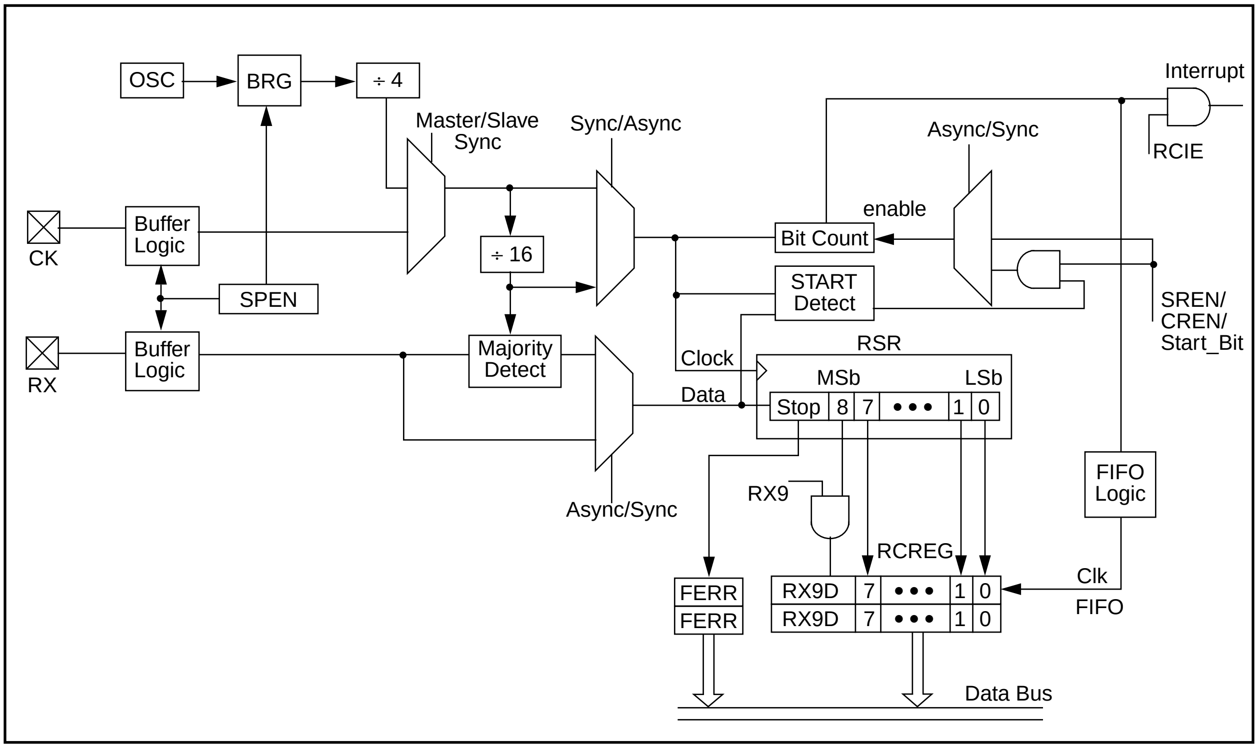


FIGURE 13-4: USART RECEIVE



SUBWF — Subtract WREG from f

Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 255$ $d \in [0,1]$
Operation:	(f) – (W) → (dest)
Status Affected:	OV, C, DC, Z
Encoding:	`0000` `010d` `ffff` `ffff`
Description:	Subtract WREG from register 'f' (2's complement method). If 'd' is 0 the result is stored in WREG. 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'	Execute	Write to destination

Example 1: `SUBWF REG1, 1`

Before Instruction
REG1 = 3
WREG = 2
C = ?

After Instruction
REG1 = 1
WREG = 2
C = 1 ; result is positive
Z = 0

Example 2:

Before Instruction
REG1 = 2
WREG = 2
C = ?

After Instruction
REG1 = 0
WREG = 2
C = 1 ; result is zero
Z = 1

Example 3:

Before Instruction
REG1 = 1
WREG = 2
C = ?

After Instruction
REG1 = FF
WREG = 2
C = 0 ; result is negative
Z = 0

SUBWFB — Subtract WREG from f with Borrow

Syntax:	[*label*] SUBWFB f,d
Operands:	$0 \le f \le 255$ $d \in [0,1]$
Operation:	(f) – (W) – $\overline{C}$ → (dest)
Status Affected:	OV, C, DC, Z
Encoding:	`0000` `001d` `ffff` `ffff`
Description:	Subtract WREG and the carry flag (borrow) from register 'f' (2's complement method). If 'd' is 0 the result is stored in WREG. 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'	Execute	Write to destination

Example 1: `SUBWFB REG1, 1`

Before Instruction
REG1 = 0x19 (`0001 1001`)
WREG = 0x0D (`0000 1101`)
C = 1

After Instruction
REG1 = 0x0C (`0000 1011`)
WREG = 0x0D (`0000 1101`)
C = 1 ; result is positive
Z = 0

Example2: `SUBWFB REG1,0`

Before Instruction
REG1 = 0x1B (`0001 1011`)
WREG = 0x1A (`0001 1010`)
C = 0

After Instruction
REG1 = 0x1B (`0001 1011`)
WREG = 0x00
C = 1 ; result is zero
Z = 1

Example3: `SUBWFB REG1,1`

Before Instruction
REG1 = 0x03 (`0000 0011`)
WREG = 0x0E (`0000 1101`)
C = 1

After Instruction
REG1 = 0xF5 (`1111 0100`) [2's comp]
WREG = 0x0E (`0000 1101`)
C = 0 ; result is negative
Z = 0

Applicable Devices	42	R42	42A	43	R43	44

17.1 DC CHARACTERISTICS: PIC17C42-16 (Commercial, Industrial) PIC17C42-25 (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	Supply Voltage	4.5	–	5.5	V	
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure internal Power-on Reset signal	–	VSS	–	V	See section on Power-on Reset for details
D004	SVDD	VDD rise rate to ensure internal Power-on Reset signal	0.060*	–	–	mV/ms	See section on Power-on Reset for details
D010 D011 D012 D013 D014	IDD	Supply Current (Note 2)	– – – – –	3 6 11 19 95	6 12 * 24 * 38 150	mA mA mA mA µA	FOSC = 4 MHz (Note 4) FOSC = 8 MHz FOSC = 16 MHz FOSC = 25 MHz FOSC = 32 kHz WDT enabled (EC osc configuration)
D020 D021	IPD	Power-down Current (Note 3)	– –	10 < 1	40 5	µA µA	VDD = 5.5V, WDT enabled VDD = 5.5V, WDT disabled

* 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: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
$\overline{\text{OSC1}}$ = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD or VSS, T0CKI = VDD, $\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.
Current consumed from the oscillator and I/O's driving external capacitive or resistive loads need to be considered.
For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as: VDD / (2 • R).
For capacitive loads, The current can be estimated (for an individual I/O pin) as (CL • VDD) • f
CL = Total capacitive load on the I/O pin; f = average frequency on the I/O pin switches.
The capacitive currents are most significant when the device is configured for external execution (includes extended microcontroller mode).

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-13: WDT TIMER TIME-OUT PERIOD vs. VDD

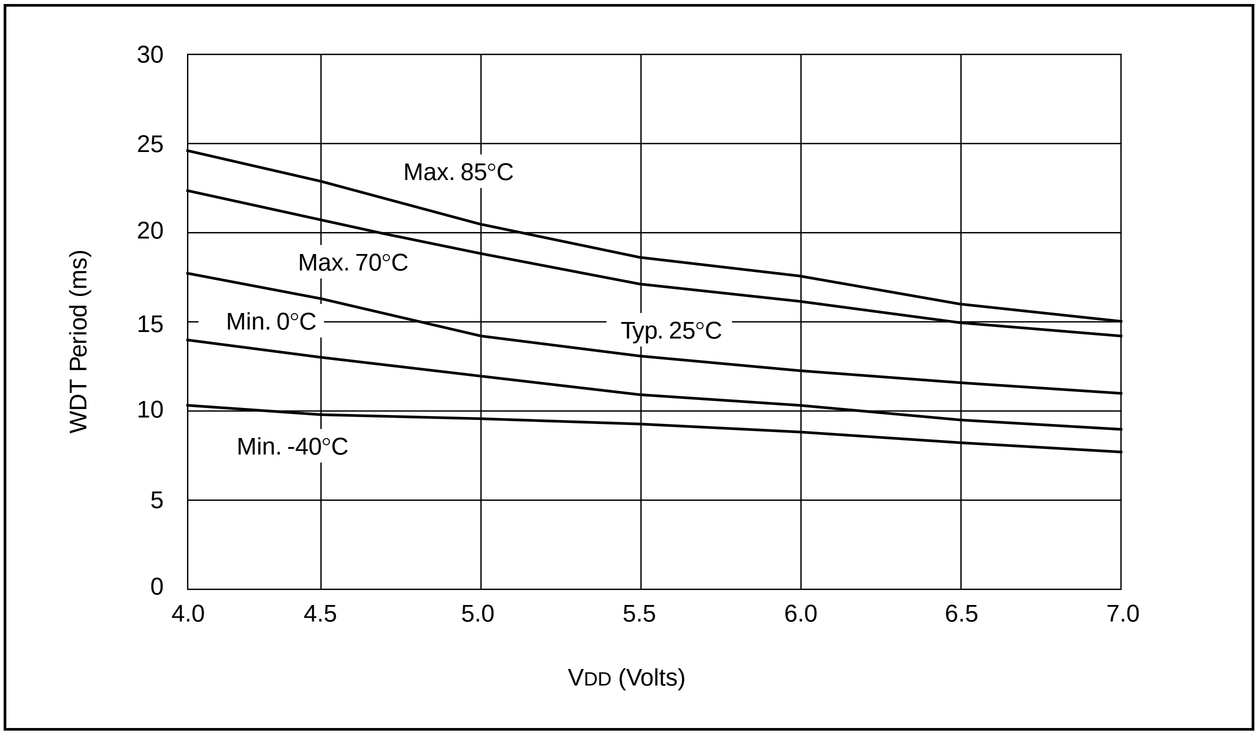
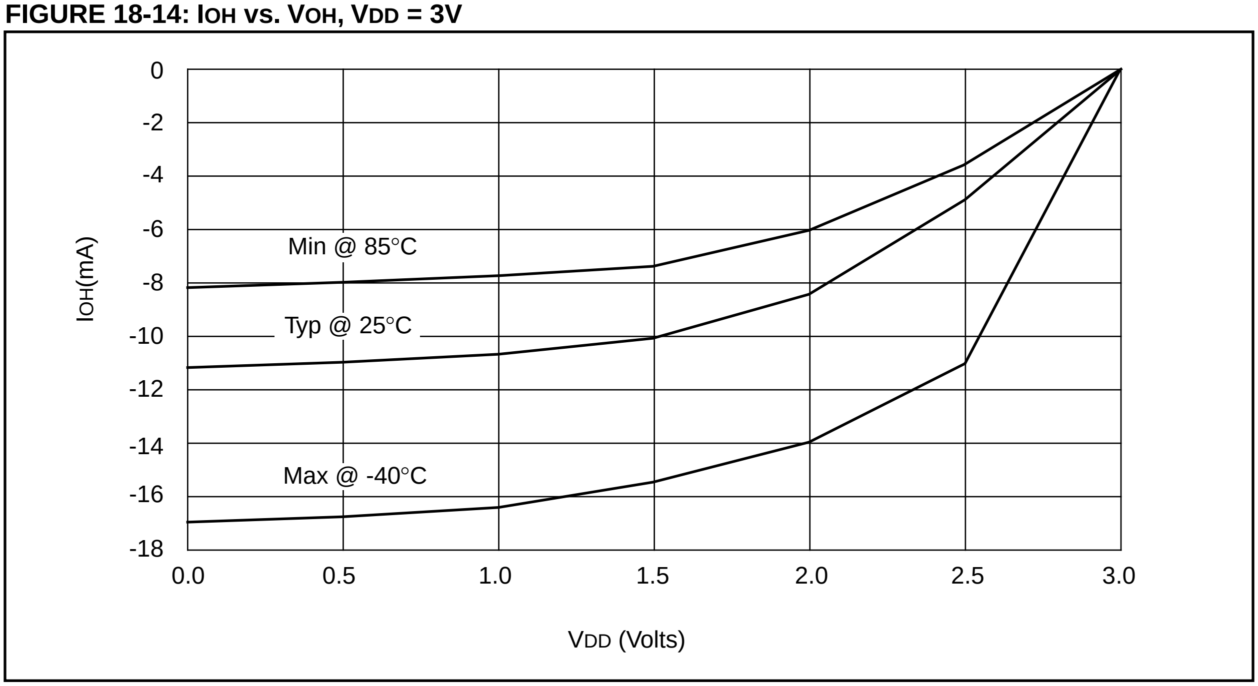


FIGURE 18-14: IOH vs. VOH, VDD = 3V



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-15: IOH vs. VOH, VDD = 5V

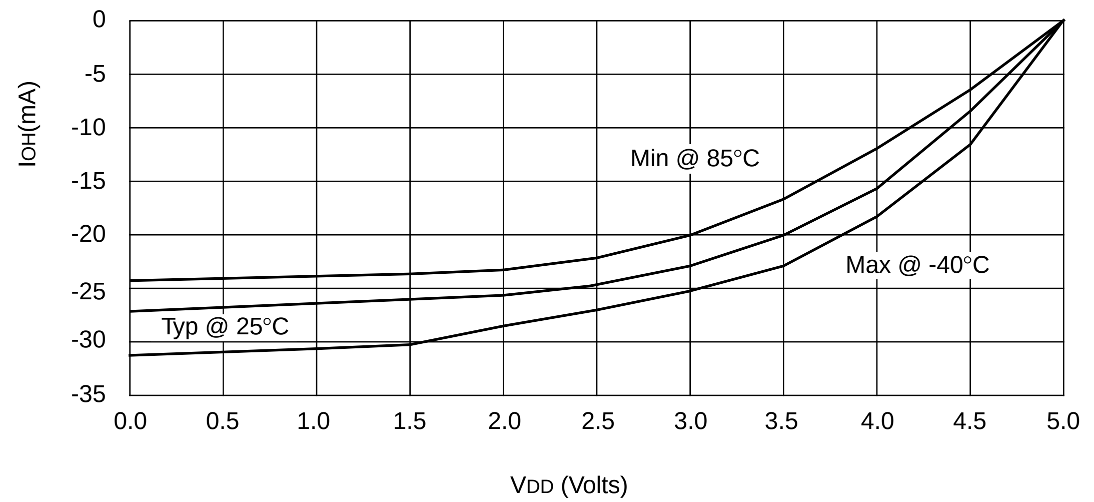
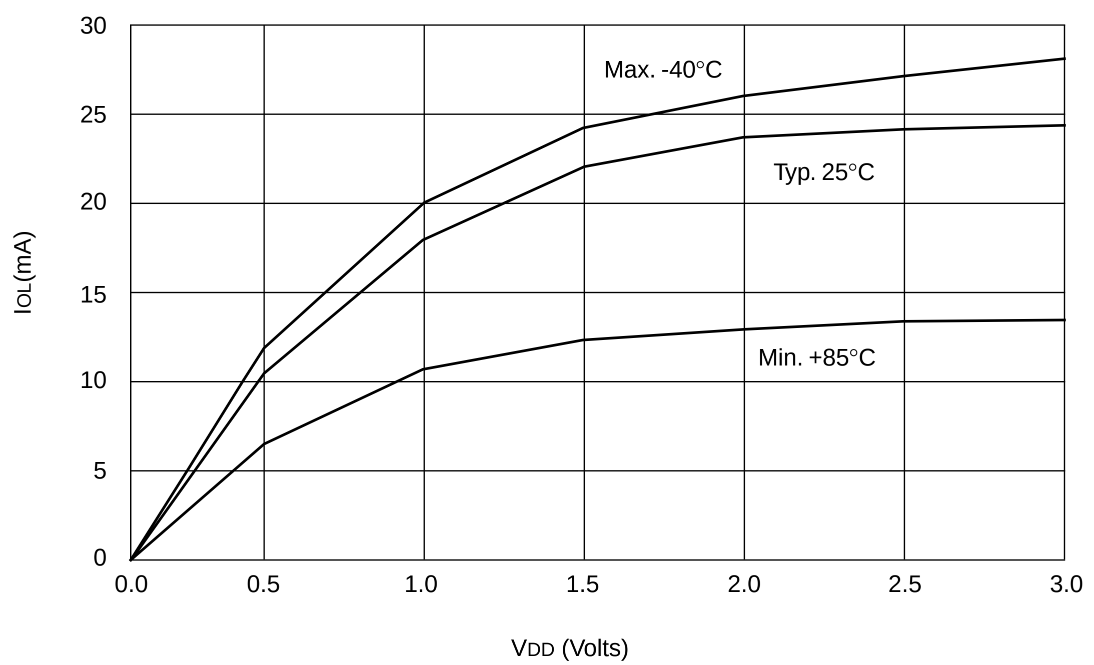


FIGURE 18-16: IOL vs. VOL, VDD = 3V



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-19: VTH, VIL of I/O PINS (SCHMITT TRIGGER) VS. VDD

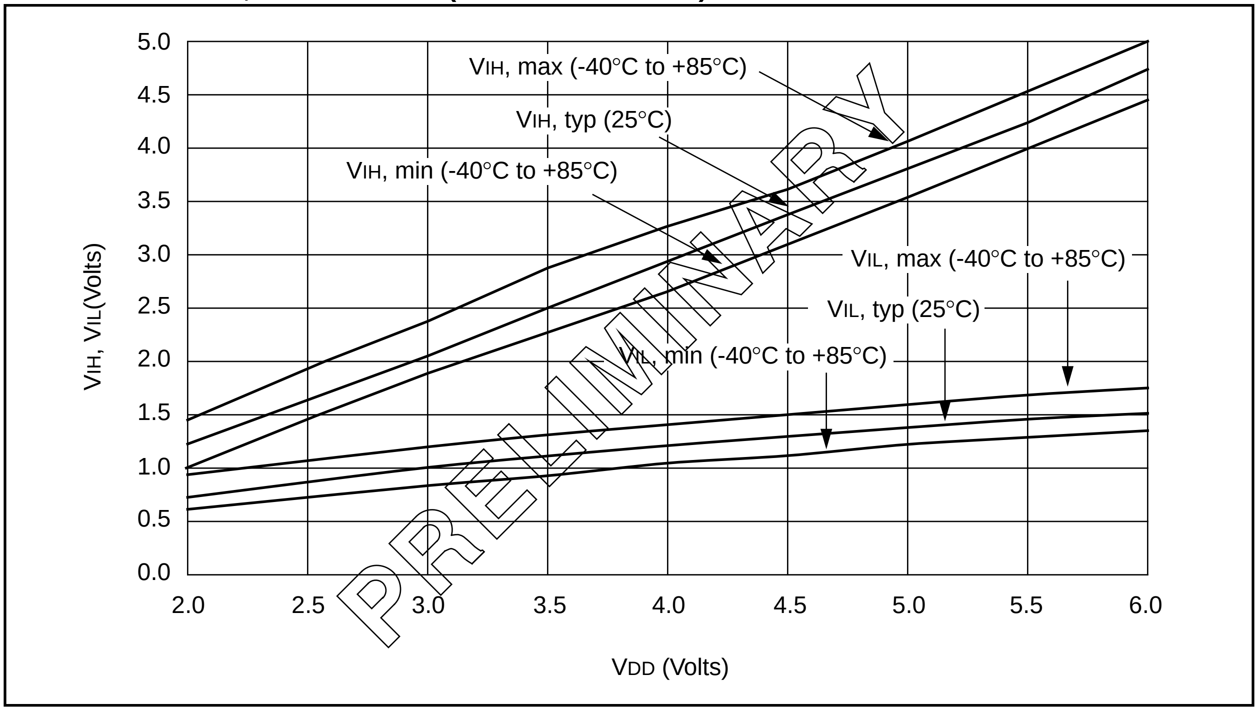
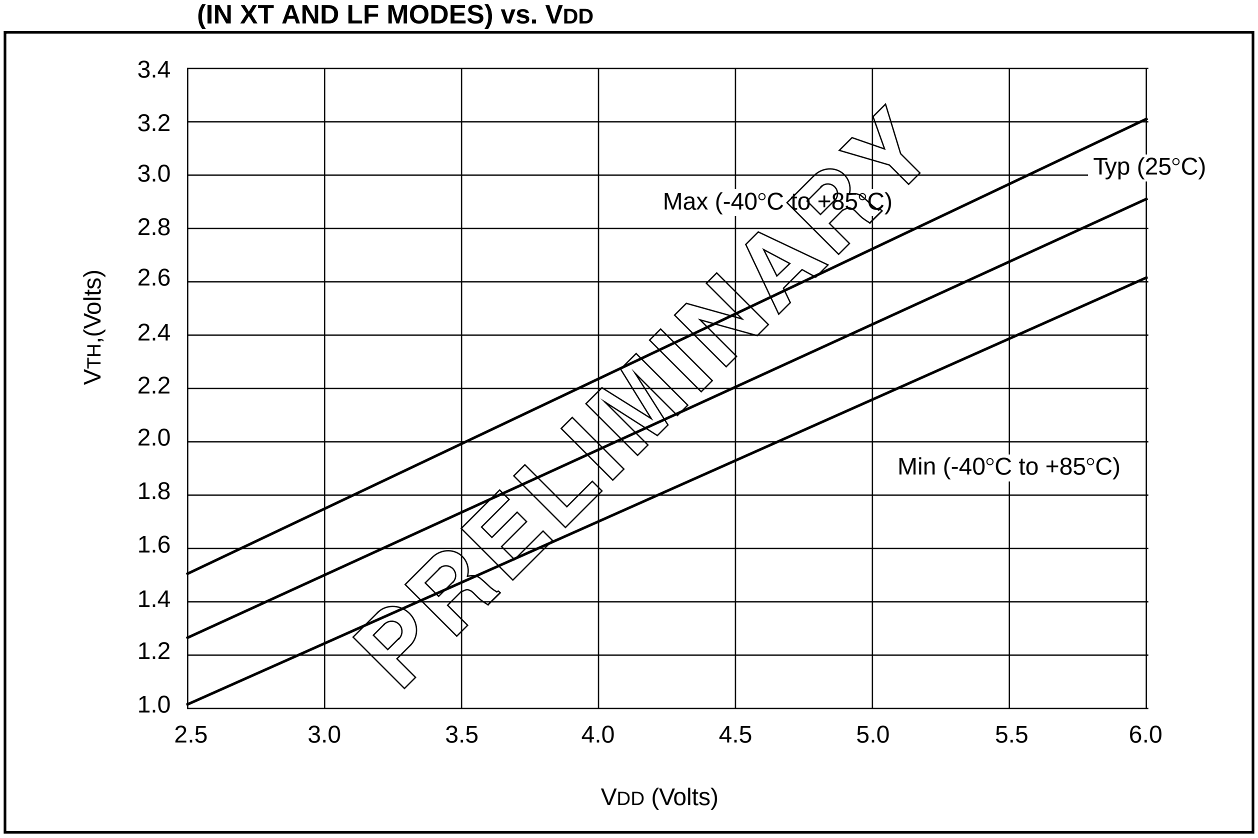


FIGURE 18-20: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs. VDD



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 19-5: TIMER0 CLOCK TIMINGS

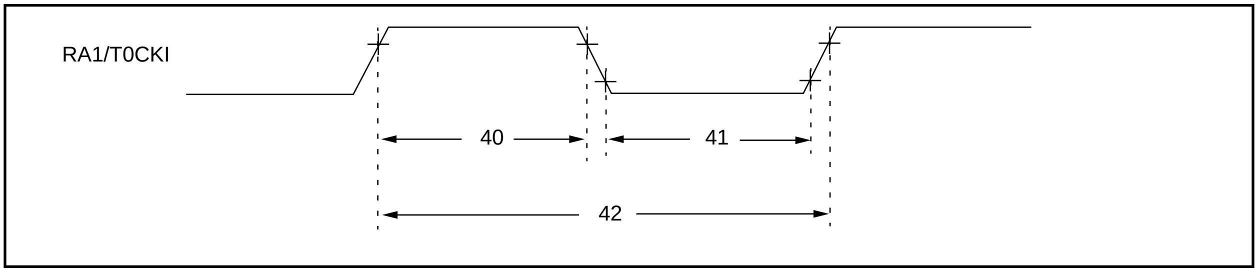


TABLE 19-5: TIMER0 CLOCK REQUIREMENTS

Parameter 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		Greater of: 20 ns or $\frac{Tcy + 40}{N}$ §	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* 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.
§ This specification ensured by design.

FIGURE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS

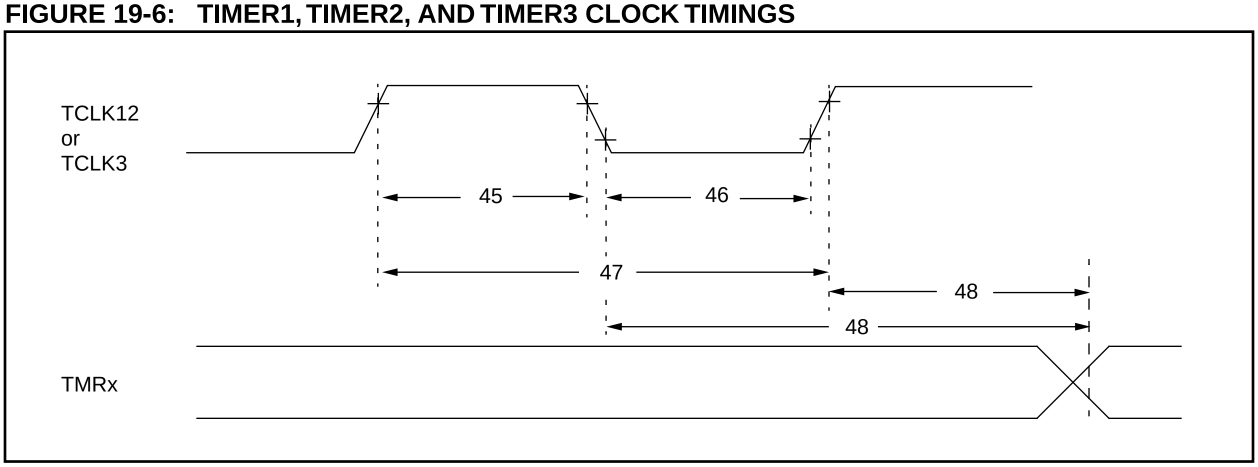


TABLE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ †	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5TCY + 20 §	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5TCY + 20 §	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	$\frac{TCY + 40}{N}$ §	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmrI	Delay from selected External Clock Edge to Timer increment	2TOSC §		6Tosc §		

* 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.
§ This specification ensured by design.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 20-17: IOL vs. VOL, VDD = 5V

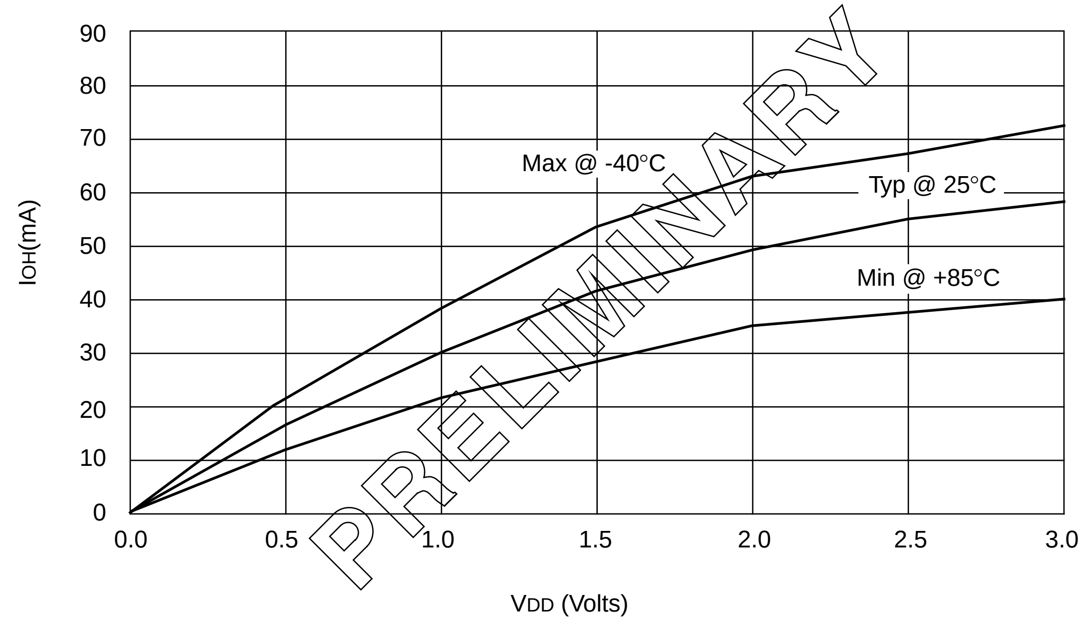
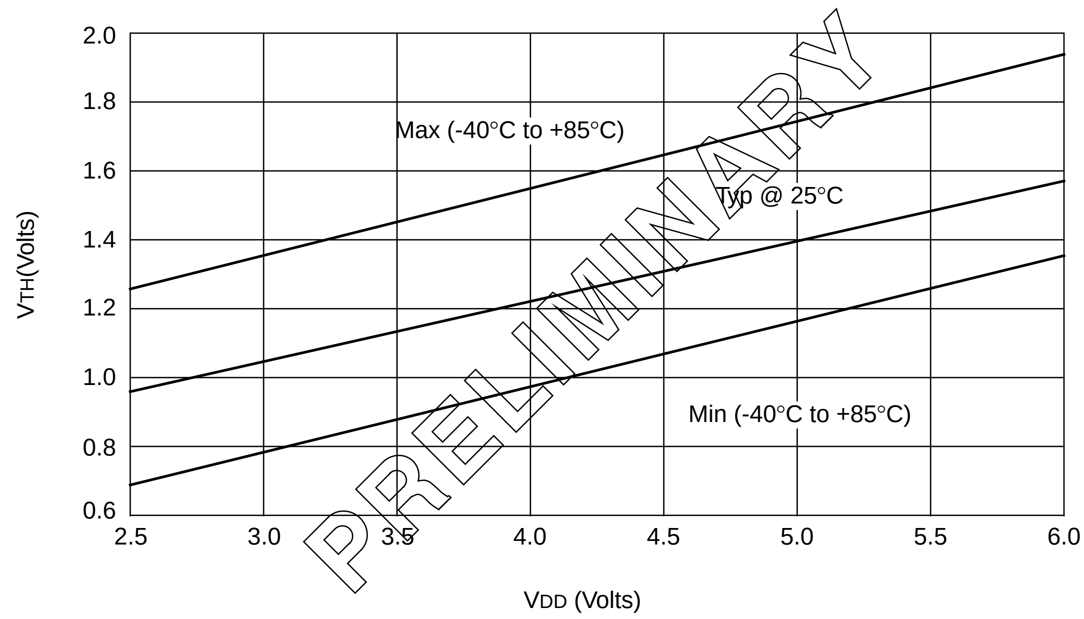


FIGURE 20-18: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) vs. VDD



APPENDIX A: MODIFICATIONS

The following is the list of modifications over the PIC16CXX microcontroller family:

1. Instruction word length is increased to 16-bit. This allows larger page sizes both in program memory (8 Kwords verses 2 Kwords) and register file (256 bytes versus 128 bytes).
2. Four modes of operation: microcontroller, protected microcontroller, extended microcontroller, and microprocessor.
3. 22 new instructions.
   The `MOVF`, `TRIS` and `OPTION` instructions have been removed.
4. 4 new instructions for transferring data between data memory and program memory. This can be used to “self program” the EPROM program memory.
5. Single cycle data memory to data memory transfers possible (`MOVPF` and `MOVFP` instructions). These instructions do not affect the Working register (WREG).
6. W register (WREG) is now directly addressable.
7. A PC high latch register (PCLATH) is extended to 8-bits. The PCLATCH register is now both readable and writable.
8. Data memory paging is redefined slightly.
9. DDR registers replaces function of TRIS registers.
10. Multiple Interrupt vectors added. This can decrease the latency for servicing the interrupt.
11. Stack size is increased to 16 deep.
12. BSR register for data memory paging.
13. Wake up from SLEEP operates slightly differently.
14. The Oscillator Start-Up Timer (OST) and Power-Up Timer (PWRT) operate in parallel and not in series.
15. PORTB interrupt on change feature works on all eight port pins.
16. TMR0 is 16-bit plus 8-bit prescaler.
17. Second indirect addressing register added (FSR1 and FSR2). Configuration bits can select the FSR registers to auto-increment, auto-decrement, remain unchanged after an indirect address.
18. Hardware multiplier added (8 x 8 $\rightarrow$ 16-bit) (PIC17C43 and PIC17C44 only).
19. Peripheral modules operate slightly differently.
20. Oscillator modes slightly redefined.
21. Control/Status bits and registers have been placed in different registers and the control bit for globally enabling interrupts has inverse polarity.
22. Addition of a test mode pin.
23. In-circuit serial programming is not implemented.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16CXX to PIC17CXX, the user should take the following steps:

1. Remove any `TRIS` and `OPTION` instructions, and implement the equivalent code.
2. Separate the interrupt service routine into its four vectors.
3. Replace:
   ```
   MOVF    REG1, W
   ```
   with:
   ```
   MOVFP   REG1, WREG
   ```
4. Replace:
   ```
   MOVF    REG1, W
   MOVWF   REG2
   ```
   with:
   ```
   MOVPF   REG1, REG2 ; Addr(REG1)<20h
   ```
   or
   ```
   MOVFP   REG1, REG2 ; Addr(REG2)<20h
   ```

> **Note:** If REG1 and REG2 are both at addresses greater then 20h, two instructions are required.
> ```
> MOVFP   REG1, WREG ;
> MOVPF   WREG, REG2 ;
> ```

5. Ensure that all bit names and register names are updated to new data memory map location.
6. Verify data memory banking.
7. Verify mode of operation for indirect addressing.
8. Verify peripheral routines for compatibility.
9. Weak pull-ups are enabled on reset.

To convert code from the PIC17C42 to all the other PIC17C4X devices, the user should take the following steps.

1. If the hardware multiply is to be used, ensure that any variables at address 18h and 19h are moved to another address.
2. Ensure that the upper nibble of the BSR was not written with a non-zero value. This may cause unexpected operation since the RAM bank is no longer 0.
3. The disabling of global interrupts has been enhanced so there is no additional testing of the GLINTD bit after a `BSF  CPUSTA,  GLINTD` instruction.

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/I²C, 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/I²C	—	7	22	3.0-6.0	Yes	—	28-pin SDIP, SOIC, SSOP
PIC16C62A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16CR62(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C63	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16CR63(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	10	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C64	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	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/I²C	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/I²C	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/I²C, 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/I²C, 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/I²C, 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.

E.8 PIC17CXX Family of Devices

	Clock: Maximum Frequency of Operation (MHz)	Memory: Program Memory (Words) EPROM	Memory: Program Memory (Words) ROM	Memory: RAM Data Memory (bytes)	Peripherals: Timer Module(s)	Peripherals: Captures	Peripherals: PWMs	Peripherals: Serial Port(s) (USART)	Features: Hardware Multiply	Features: External Interrupts	Features: Interrupt Sources	Features: I/O Pins	Features: Voltage Range (Volts)	Features: Number of Instructions	Features: Packages
PIC17C42	25	2K	—	232	TMR0,TMR1, TMR2,TMR3	2	2	Yes	—	Yes	11	33	4.5-5.5	55	40-pin DIP; 44-pin PLCC, MQFP
PIC17C42A	25	2K	—	232	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17CR42	25	—	2K	232	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17C43	25	4K	—	454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17CR43	25	—	4K	454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17C44	25	8K		454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

E

F

G

H

I

LIST OF TABLES