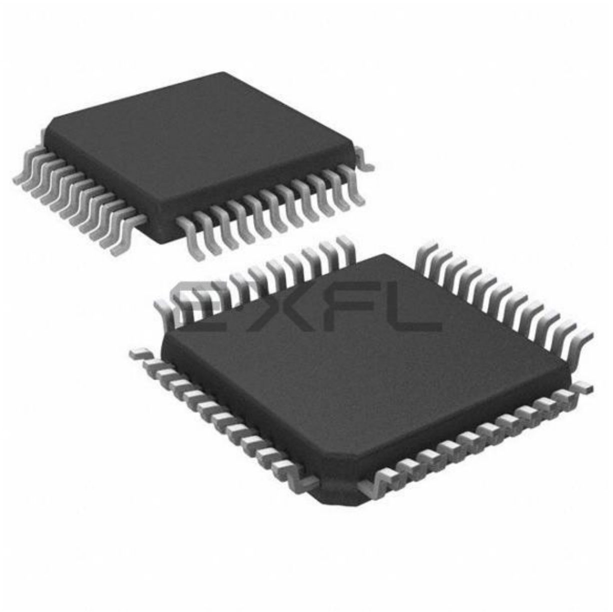
Microchip Technology - PIC17LC44-08I/PQ Datasheet



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	Active
Core Processor	PIC
Core Size	8-Bit
Speed	8MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	16KB (8K 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	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17lc44-08i-pq

1.0 OVERVIEW

This data sheet covers the PIC17C4X group of the PIC17CXX family of microcontrollers. The following devices are discussed in this data sheet:

- PIC17C42
- PIC17CR42
- PIC17C42A
- PIC17C43
- PIC17CR43
- PIC17C44

The PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, and PIC17C44 devices include architectural enhancements over the PIC17C42. These enhancements will be discussed throughout this data sheet.

The PIC17C4X devices are 40/44-Pin, EPROM/ROM-based members of the versatile PIC17CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC17CXX has enhanced core features, 16-level deep stack, and multiple internal and external interrupt sources.The separate instruction and data buses of the Harvard architecture allow a 16-bit wide instruction word with a separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 55 instructions (reduced instruction set) are available in the PIC17C42 and 58 instructions in all the other devices. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance. For mathematical intensive applications all devices, except the PIC17C42, have a single cycle 8 x 8 Hardware Multiplier.

PIC17CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC17C4X devices have up to 454 bytes of RAM and 33 I/O pins. In addition, the PIC17C4X adds several peripheral features useful in many high performance applications including:

- Four timer/counters
- Two capture inputs
- Two PWM outputs
- A Universal Synchronous Asynchronous Receiver Transmitter (USART)

These special features reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LF oscillator is for low frequency crystals and minimizes power consumption, XT is a standard crystal, and the EC is for external clock input. The SLEEP (power-down) mode offers additional power saving. The user can wake-up the chip from SLEEP through several external and internal interrupts and device resets.

There are four configuration options for the device operational modes:

- Microprocessor
- Microcontroller
- Extended microcontroller
- Protected microcontroller

The microprocessor and extended microcontroller modes allow up to 64K-words of external program memory.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software malfunction.

Table 1-1 lists the features of the PIC17C4X devices.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

The PIC17C4X fits perfectly in applications ranging from precise motor control and industrial process control to automotive, instrumentation, and telecom applications. Other applications that require extremely fast execution of complex software programs or the flexibility of programming the software code as one of the last steps of the manufacturing process would also be well suited. The EPROM technology makes customization of application programs (with unique security codes, combinations, model numbers, parameter storage, etc.) fast and convenient. Small footprint package options make the PIC17C4X ideal for applications with space limitations that require high performance. High speed execution, powerful peripheral features, flexible I/O, and low power consumption all at low cost make the PIC17C4X ideal for a wide range of embedded control applications.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X and PIC16CXX families of microcontrollers will see the architectural enhancements that have been implemented. These enhancements allow the device to be more efficient in software and hardware requirements. Please refer to Appendix A for a detailed list of enhancements and modifications. Code written for PIC16C5X or PIC16CXX can be easily ported to PIC17CXX family of devices (Appendix B).

1.2 Development Support

The PIC17CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a universal programmer, a “C” compiler, and fuzzy logic support tools.

5.4 Interrupt Operation

Global Interrupt Disable bit, GLINTD (CPUSTA<4>), enables all unmasked interrupts (if clear) or disables all interrupts (if set). Individual interrupts can be disabled through their corresponding enable bits in the INTSTA register. Peripheral interrupts need either the global peripheral enable PEIE bit disabled, or the specific peripheral enable bit disabled. Disabling the peripherals via the global peripheral enable bit, disables all peripheral interrupts. GLINTD is set on reset (interrupts disabled).

The `RETFIE` instruction allows returning from interrupt and re-enable interrupts at the same time.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with interrupt vector. There are four interrupt vectors to reduce interrupt latency.

The peripheral interrupt vector has multiple interrupt sources. Once in the peripheral interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The peripheral interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid continuous interrupts.

The PIC17C4X devices have four interrupt vectors. These vectors and their hardware priority are shown in Table 5-1. If two enabled interrupts occur "at the same time", the interrupt of the highest priority will be serviced first. This means that the vector address of that interrupt will be loaded into the program counter (PC).

TABLE 5-1: INTERRUPT VECTORS/ PRIORITIES

Address	Vector	Priority
0008h	External Interrupt on RA0/ INT pin (INTF)	1 (Highest)
0010h	TMR0 overflow interrupt (T0IF)	2
0018h	External Interrupt on T0CKI (T0CKIF)	3
0020h	Peripherals (PEIF)	4 (Lowest)

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

Note 2: When disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).

Note 3: For the PIC17C42 only:
If an interrupt occurs while the Global Interrupt Disable (GLINTD) bit is being set, the GLINTD bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the `RETFIE` instruction). The events that would cause this to occur are:

1. An interrupt occurs simultaneously with an instruction that sets the GLINTD bit.
2. The program branches to the Interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the `RETFIE` instruction. This causes the GLINTD bit to be cleared (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

1. Ensure that the GLINTD bit was set by the instruction, as shown in the following code:

```
LOOP   BSF    CPUSTA, GLINTD ; Disable Global
                             ; Interrupt
       BTFSS  CPUSTA, GLINTD ; Global Interrupt
                             ; Disabled?
       GOTO   LOOP           ; NO, try again
                             ; YES, continue
                             ; with program
                             ; low
```

7.3 Table Reads

The table read allows the program memory to be read. This allows constant data to be stored in the program memory space, and retrieved into data memory when needed. Example 7-2 reads the 16-bit value at program memory address TBLPTR. After the dummy byte has been read from the TABLATH, the TABLATH is loaded with the 16-bit data from program memory address TBLPTR + 1. The first read loads the data into the latch, and can be considered a dummy read (unknown data loaded into 'f'). INDF0 should be configured for either auto-increment or auto-decrement.

EXAMPLE 7-2: TABLE READ

```
MOVLW   HIGH (TBL_ADDR) ; Load the Table
MOVWF   TBLPTRH         ;   address
MOVLW   LOW (TBL_ADDR)  ;
MOVWF   TBLPTRL         ;
TABLRD  0,0,DUMMY       ; Dummy read,
                        ;  Updates TABLATCH
TLRD    1, INDF0        ; Read HI byte
                        ;   of TABLATCH
TABLRD  0,1,INDF0       ; Read LO byte
                        ;   of TABLATCH and
                        ;   Update TABLATCH
```

FIGURE 7-7: TABLRD TIMING

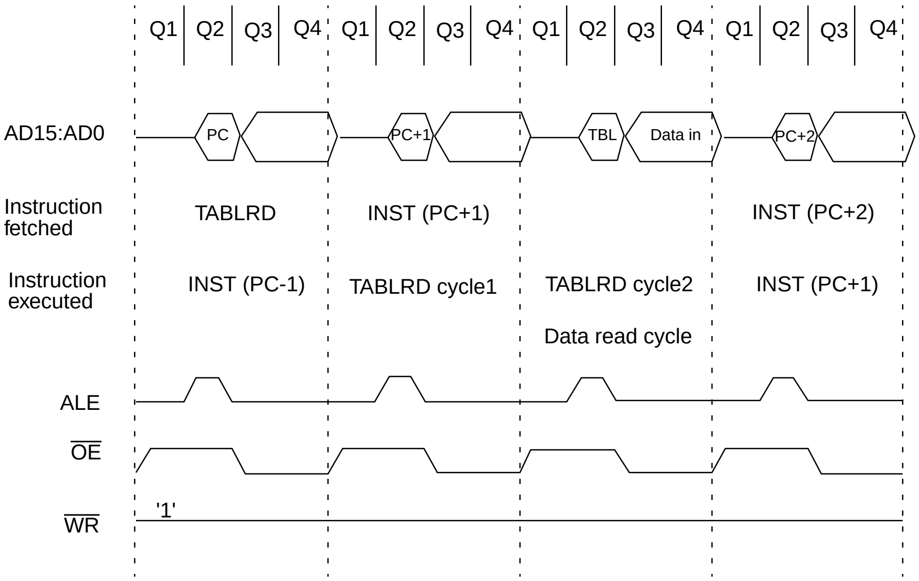
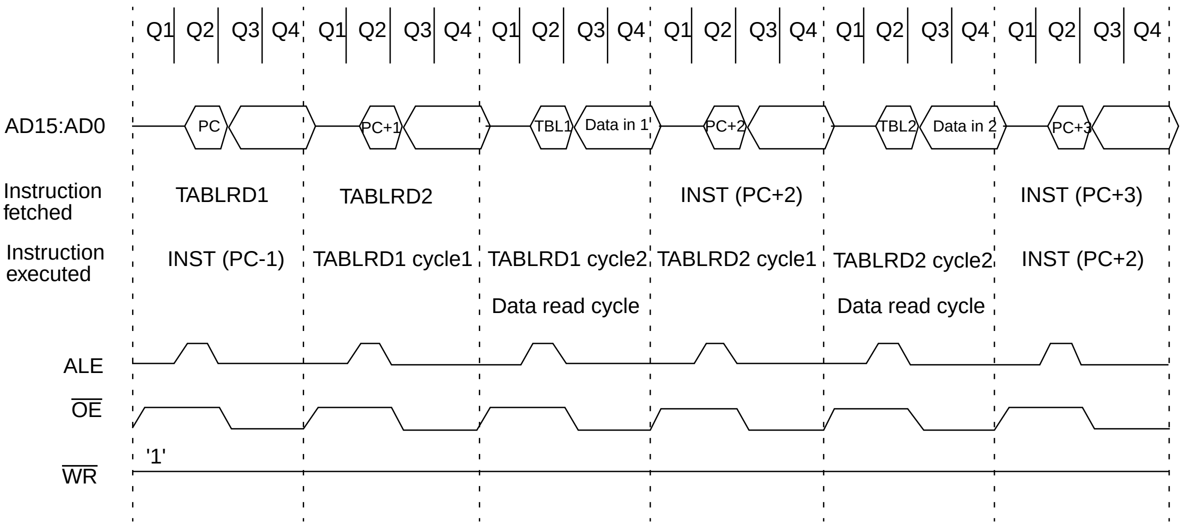


FIGURE 7-8: TABLRD TIMING (CONSECUTIVE TABLRD INSTRUCTIONS)



12.2.1 ONE CAPTURE AND ONE PERIOD REGISTER MODE

In this mode registers PR3H/CA1H and PR3L/CA1L constitute a 16-bit period register. A block diagram is shown in Figure 12-7. The timer increments until it equals the period register and then resets to 0000h. TMR3 Interrupt Flag bit (TMR3IF) is set at this point. This interrupt can be disabled by clearing the TMR3 Interrupt Enable bit (TMR3IE). TMR3IF must be cleared in software.

This mode is selected if control bit CA1/$\overline{\text{PR3}}$ is clear. In this mode, the Capture1 register, consisting of high byte (PR3H/CA1H) and low byte (PR3L/CA1L), is configured as the period control register for TMR3. Capture1 is disabled in this mode, and the corresponding Interrupt bit CA1IF is never set. TMR3 increments until it equals the value in the period register and then resets to 0000h.

Capture2 is active in this mode. The CA2ED1 and CA2ED0 bits determine the event on which capture will occur. The possible events are:

- Capture on every falling edge
- Capture on every rising edge
- Capture every 4th rising edge
- Capture every 16th rising edge

When a capture takes place, an interrupt flag is latched into the CA2IF bit. This interrupt can be enabled by setting the corresponding mask bit CA2IE. The Peripheral Interrupt Enable bit (PEIE) must be set and the Global Interrupt Disable bit (GLINTD) must be cleared for the interrupt to be acknowledged. The CA2IF interrupt flag bit must be cleared in software.

When the capture prescale select is changed, the prescaler is not reset and an event may be generated. Therefore, the first capture after such a change will be ambiguous. However, it sets the time-base for the next capture. The prescaler is reset upon chip reset.

Capture pin RB1/CAP2 is a multiplexed pin. When used as a port pin, Capture2 is not disabled. However, the user can simply disable the Capture2 interrupt by clearing CA2IE. If RB1/CAP2 is used as an output pin, the user can activate a capture by writing to the port pin. This may be useful during development phase to emulate a capture interrupt.

The input on capture pin RB1/CAP2 is synchronized internally to internal phase clocks. This imposes certain restrictions on the input waveform (see the Electrical Specification section for timing).

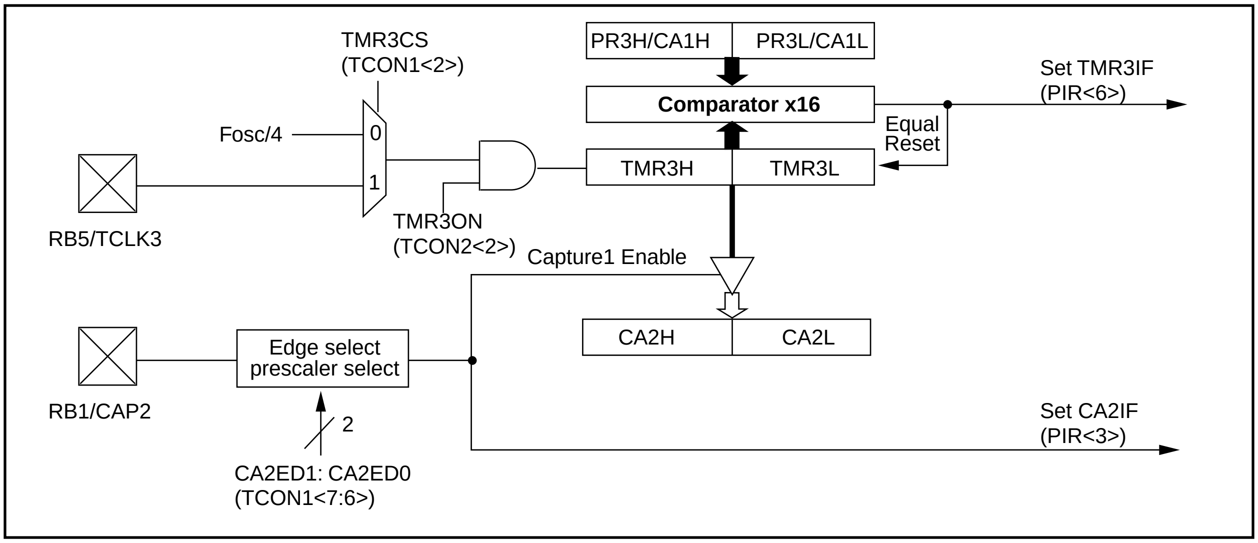
The Capture2 overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture2 register and another “event” has occurred on the RB1/CA2 pin. The new event will not transfer the Timer3 value to the capture register, protecting the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture2 register, the master overflow bit is transferred to the slave overflow bit (CA2OVF) and then the master bit is reset. The user can then read TCON2 to determine the value of CA2OVF.

The recommended sequence to read capture registers and capture overflow flag bits is shown in Example 12-1.

EXAMPLE 12-1: SEQUENCE TO READ CAPTURE REGISTERS

```
MOVLB 3              ;Select Bank 3
MOVPF CA2L,LO_BYTE   ;Read Capture2 low
                     ;byte, store in LO_BYTE
MOVPF CA2H,HI_BYTE   ;Read Capture2 high
                     ;byte, store in HI_BYTE
MOVPF TCON2,STAT_VAL ;Read TCON2 into file
                     ;STAT_VAL
```

FIGURE 12-7: TIMER3 WITH ONE CAPTURE AND ONE PERIOD REGISTER BLOCK DIAGRAM



13.4 USART Synchronous Slave Mode

The synchronous slave mode differs from the master mode in the fact that the shift clock is supplied externally at the RA5/TX/CK pin (instead of being supplied internally in the master mode). This allows the device to transfer or receive data in the SLEEP mode. The slave mode is entered by clearing the CSRC (TXSTA<7>) bit.

13.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the sync master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to TXREG and then the SLEEP instruction executes, the following will occur. The first word will immediately transfer to the TSR and will transmit as the shift clock is supplied. The second word will remain in TXREG. TXIF will not be set. When the first word has been shifted out of TSR, TXREG will transfer the second word to the TSR and the TXIF flag will now be set. If TXIE is enabled, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, then the program will branch to interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. Clear the CREN bit.
3. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Start transmission by loading data to TXREG.
6. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
7. Enable the transmission by setting TXEN.

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN) allows transmission to start sooner then doing these two events in the reverse order.

Note:	To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

13.4.2 USART SYNCHRONOUS SLAVE RECEPTION

Operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode. Also, SREN is a don't care in slave mode.

If receive is enabled (CREN) prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR will transfer the data to RCREG (setting RCIF) and if the RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. If interrupts are desired, then set the RCIE bit.
3. If 9-bit reception is desired, then set the RX9 bit.
4. To enable reception, set the CREN bit.
5. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
6. Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading RCREG.
8. If any error occurred, clear the error by clearing the CREN bit.

Note:	To abort reception, either clear the SPEN bit, the SREN bit (when in single receive mode), or the CREN bit (when in continuous receive mode). This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

14.4.2 MINIMIZING CURRENT CONSUMPTION

To minimize current consumption, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should be at VDD or VSS. The contributions from on-chip pull-ups on PORTB should also be considered, and disabled when possible.

14.5 Code Protection

The code in the program memory can be protected by selecting the microcontroller in code protected mode (PM2:PM0 = '000').

Note:	PM2 does not exist on the PIC17C42. To select code protected microcontroller mode, PM1:PM0 = '00'.

In this mode, instructions that are in the on-chip program memory space, can continue to read or write the program memory. An instruction that is executed outside of the internal program memory range will be inhibited from writing to or reading from program memory.

Note:	Microchip does not recommend code protecting windowed devices.

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

ADDLW ADD Literal to WREG

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(WREG) + k → (WREG)
Status Affected:	OV, C, DC, Z
Encoding:	`1011` `0001` `kkkk` `kkkk`
Description:	The contents of WREG are added to the 8-bit literal 'k' and the result is placed in WREG.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Execute	Write to WREG

Example: `ADDLW 0x15`

Before Instruction
WREG = 0x10

After Instruction
WREG = 0x25

ADDWF ADD WREG to f

Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	(WREG) + (f) → (dest)
Status Affected:	OV, C, DC, Z
Encoding:	`0000` `111d` `ffff` `ffff`
Description:	Add WREG to register 'f'. 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: `ADDWF REG, 0`

Before Instruction

WREG	=	0x17
REG	=	0xC2

After Instruction

WREG	=	0xD9
REG	=	0xC2

RETFIE Return from Interrupt

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → (PC);
0 → GLINTD;
PCLATH is unchanged.

Status Affected: GLINTD

Encoding:

`0000`	`0000`	`0000`	`0101`

Description: Return from Interrupt. Stack is POP'ed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by clearing the GLINTD bit. GLINTD is the global interrupt disable bit (CPUSTA<4>).

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register T0STA	Execute	NOP
Forced NOP	NOP	Execute	NOP

Example: `RETFIE`

After Interrupt
PC = TOS
GLINTD = 0

RETLW Return Literal to WREG

Syntax: [*label*] RETLW k

Operands: $0 \le k \le 255$

Operation: k → (WREG); TOS → (PC);
PCLATH is unchanged

Status Affected: None

Encoding:

`1011`	`0110`	`kkkk`	`kkkk`

Description: WREG is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Execute	Write to WREG
Forced NOP	NOP	Execute	NOP

Example:

```
    CALL TABLE ; WREG contains table
               ;  offset value
               ;  WREG now has
               ;  table value
    :
TABLE
    ADDWF PC   ; WREG = offset
    RETLW k0   ; Begin table
    RETLW k1   ;
    :
    :
    RETLW kn   ; End of table
```

Before Instruction
WREG = 0x07

After Instruction
WREG = value of k7

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

DC CHARACTERISTICS		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40˚C ≤ T_A ≤ +40˚C Operating voltage V_{DD} range as described in Section 17.1					
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Internal Program Memory Programming Specs (Note 4)					
D110	V_{PP}	Voltage on $\overline{MCLR}/V_{PP}$ pin	12.75	–	13.25	V	Note 5
D111	V_{DDP}	Supply voltage during programming	4.75	5.0	5.25	V	
D112	I_{PP}	Current into $\overline{MCLR}/V_{PP}$ pin	–	25 ‡	50 ‡	mA	
D113	I_{DDP}	Supply current during programming	–	–	30 ‡	mA	
D114	T_{PROG}	Programming pulse width	10	100	1000	µs	Terminated via internal/external interrupt or a reset

* 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.

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The $\overline{MCLR}/V_{PP}$ pin may be kept in this range at times other than programming, but is not recommended.

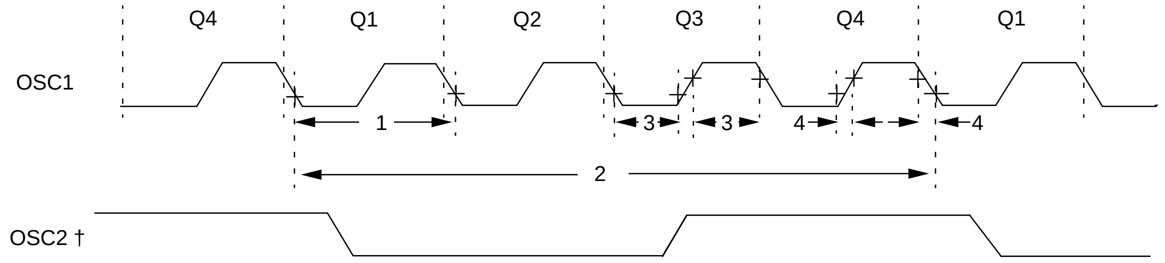
6: For TTL buffers, the better of the two specifications may be used.

Note:	When using the Table Write for internal programming, the device temperature must be less than 40˚C.

Applicable Devices	42	R42	42A	43	R43	44

17.4 Timing Diagrams and Specifications

FIGURE 17-2: EXTERNAL CLOCK TIMING



† In EC and RC modes only.

TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	**External CLKIN Frequency (Note 1)**	DC DC	— —	16 25	MHz MHz	EC osc mode - PIC17C42-16 - PIC17C42-25
		Oscillator Frequency (Note 1)	DC 1 1 DC	— — — —	4 16 25 2	MHz MHz MHz MHz	RC osc mode XT osc mode - PIC17C42-16 - PIC17C42-25 LF osc mode
1	Tosc	**External CLKIN Period (Note 1)**	62.5 40	— —	— —	ns ns	EC osc mode - PIC17C42-16 - PIC17C42-25
		Oscillator Period (Note 1)	250 62.5 40 500	— — — —	— 1,000 1,000 —	ns ns ns ns	RC osc mode XT osc mode - PIC17C42-16 - PIC17C42-25 LF osc mode
2	TCY	**Instruction Cycle Time (Note 1)**	160	4/Fosc	DC	ns	
3	TosL, TosH	**Clock in (OSC1) High or Low Time**	10 ‡	—	—	ns	EC oscillator
4	TosR, TosF	**Clock in (OSC1) Rise or Fall Time**	—	—	5 ‡	ns	EC oscillator

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

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1 pin.
When an external clock input is used, the “Max.” cycle time limit is “DC” (no clock) for all devices.

FIGURE 17-11: MEMORY INTERFACE WRITE TIMING

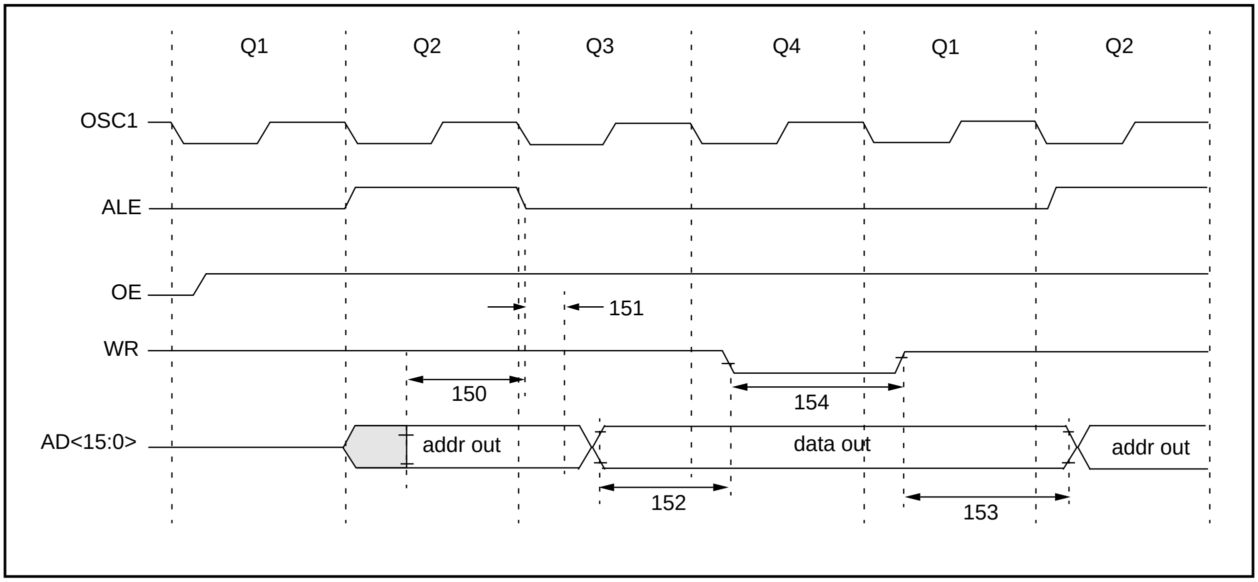


TABLE 17-11: MEMORY INTERFACE WRITE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25Tcy - 30	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	0	—	—	ns	
152	TadV2wrL	Data out valid to $\overline{WR}$↓ (data setup time)	0.25Tcy - 40	—	—	ns	
153	TwrH2adI	$\overline{WR}$↑ to data out invalid (data hold time)	—	0.25TCY §	—	ns	
154	TwrL	$\overline{WR}$ pulse width	—	0.25TCY §	—	ns	

* 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 is guaranteed by design.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 18-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

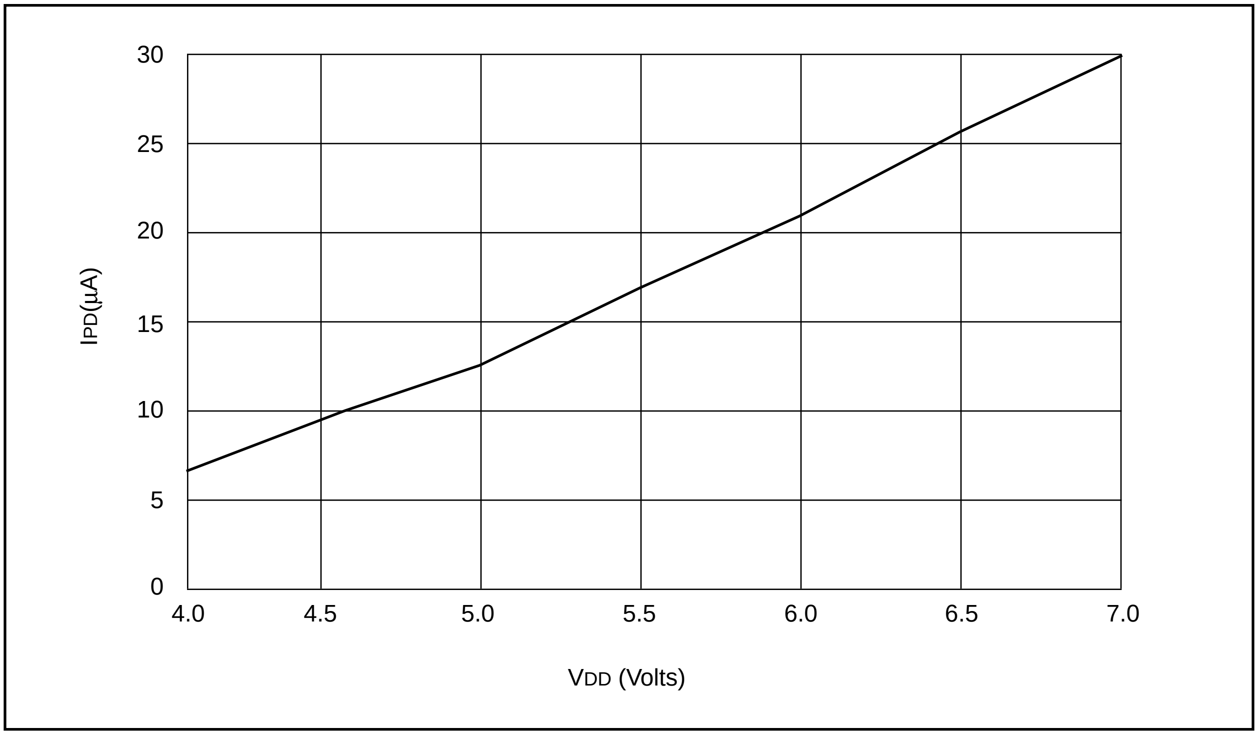
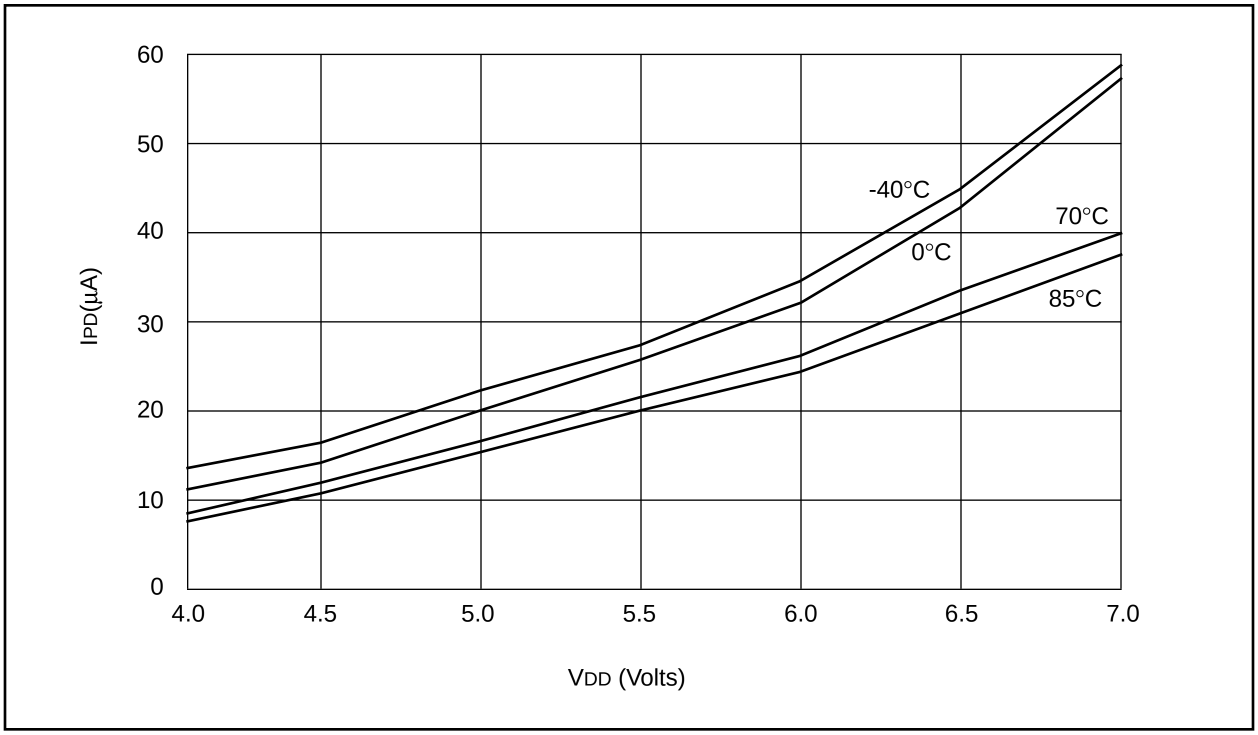


FIGURE 18-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



Applicable Devices	42	R42	42A	43	R43	44

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

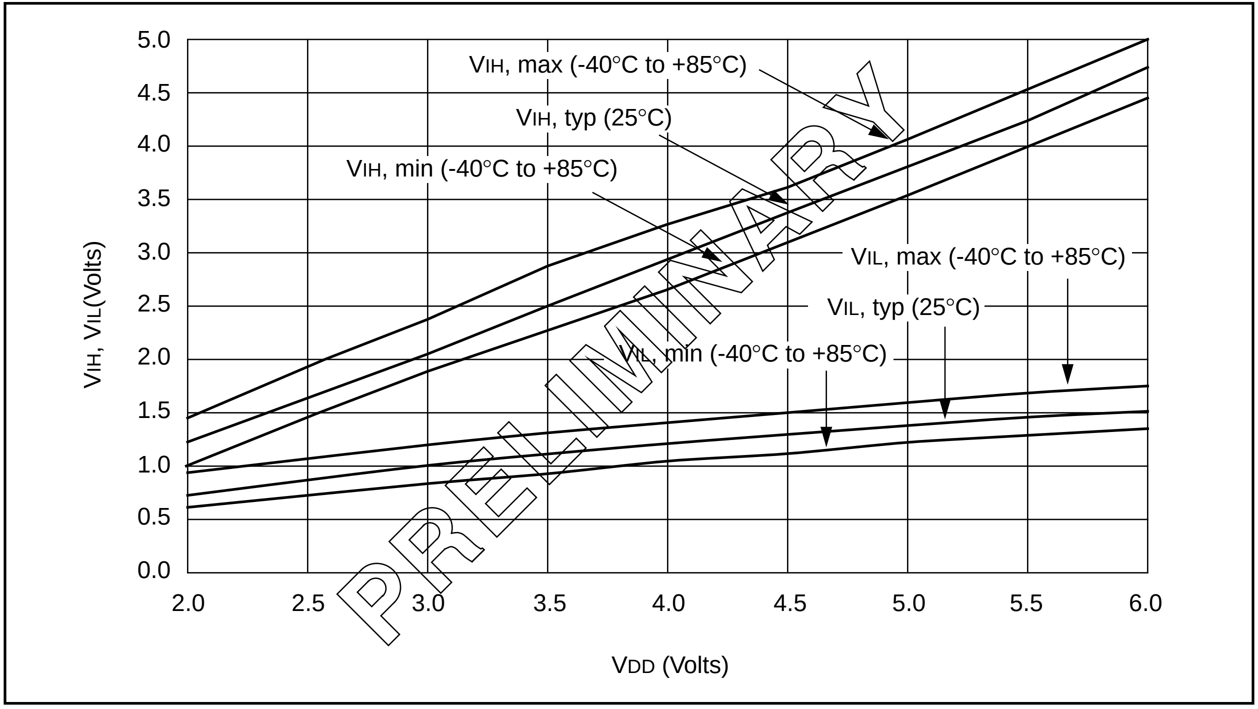
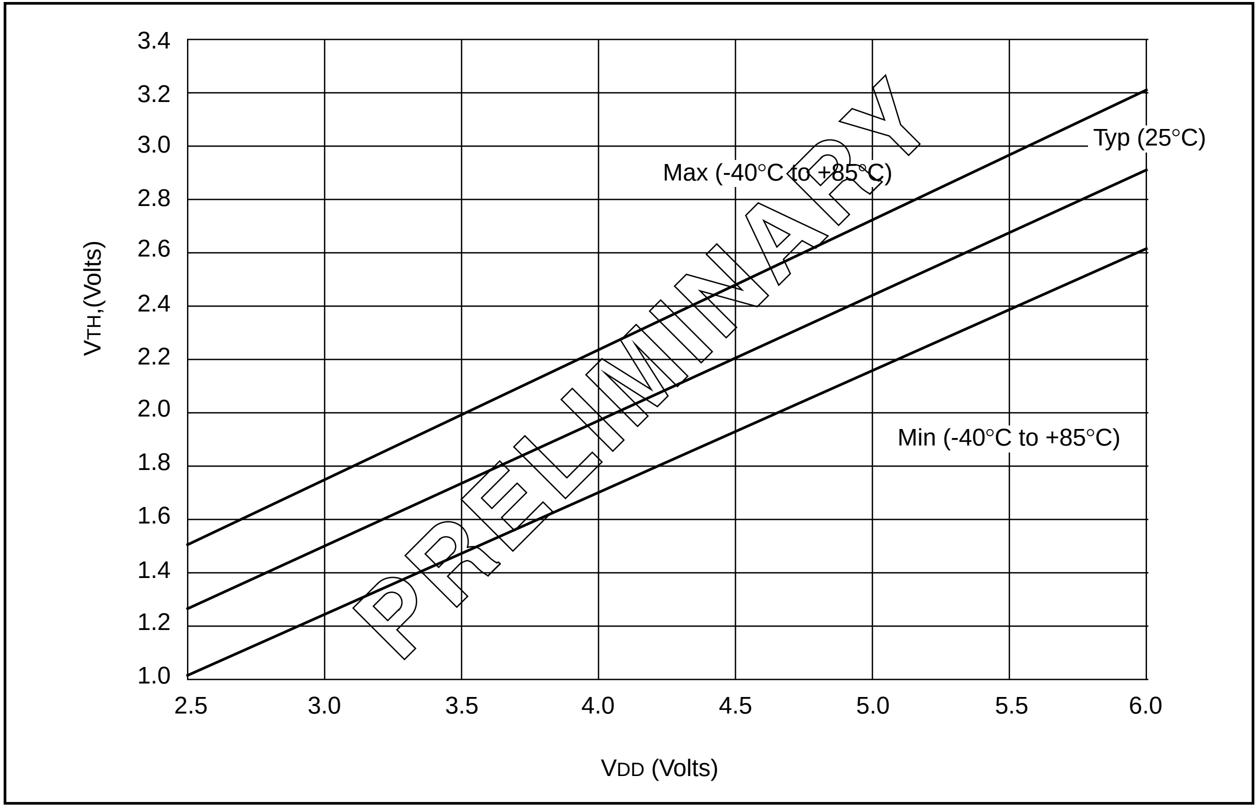


FIGURE 18-20: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs. VDD



Applicable Devices	42	R42	42A	43	R43	44

FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, AND POWER-UP TIMER TIMING

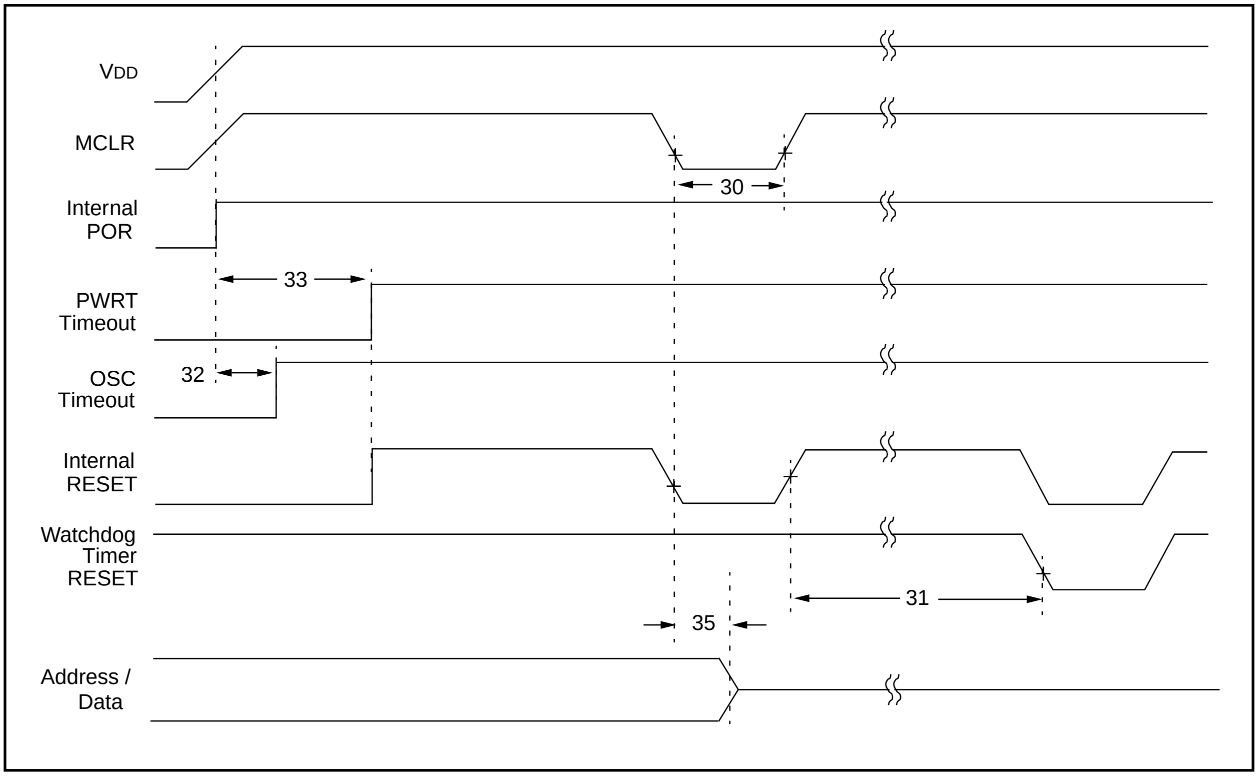


TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)		100 *	—	—	ns	VDD = 5V
31	Twdt	Watchdog Timer Time-out Period (Prescale = 1)		5 *	12	25 *	ms	VDD = 5V
32	Tost	Oscillation Start-up Timer Period		—	1024TOSC§	—	ms	TOSC = OSC1 period
33	Tpwrt	Power-up Timer Period		40 *	96	200 *	ms	VDD = 5V
35	TmcL2adl	$\overline{\text{MCLR}}$ to System Interface bus (AD15:AD0>) invalid	PIC17CR42/42A/43/R43/44	—	—	100 *	ns	
			PIC17LCR42/42A/43/R43/44	—	—	120 *	ns	

* 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.

‡ These parameters are for design guidance only and are not tested, nor characterized.

§ This specification ensured by design.

Applicable Devices	42	R42	42A	43	R43	44

FIGURE 20-5: TRANSCONDUCTANCE (gm) OF LF OSCILLATOR vs. VDD

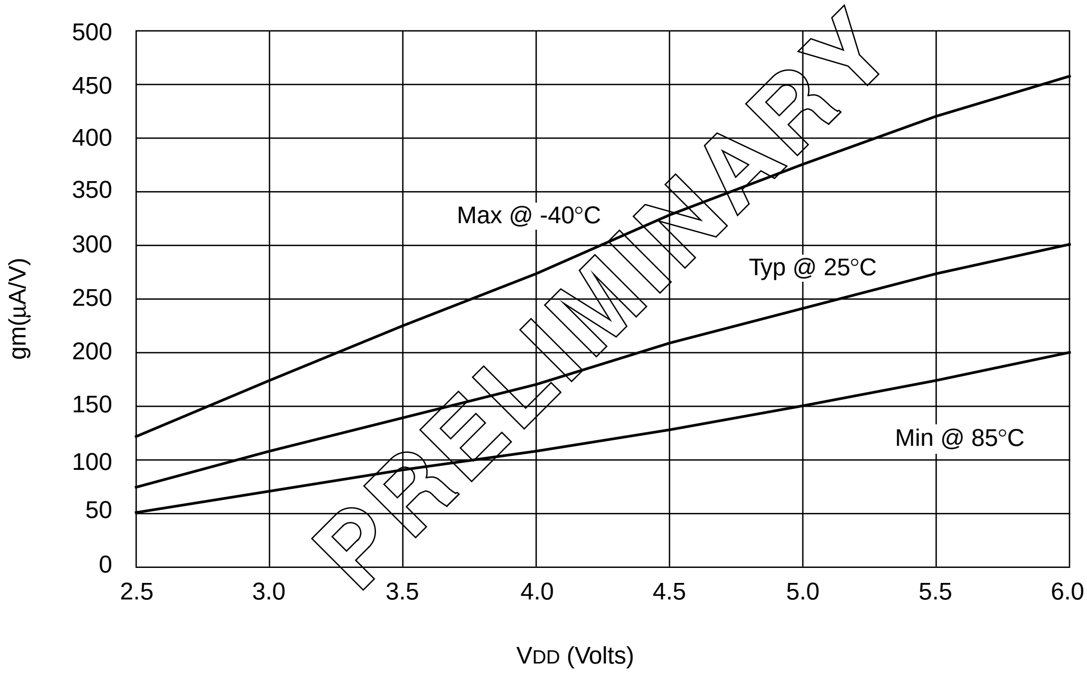
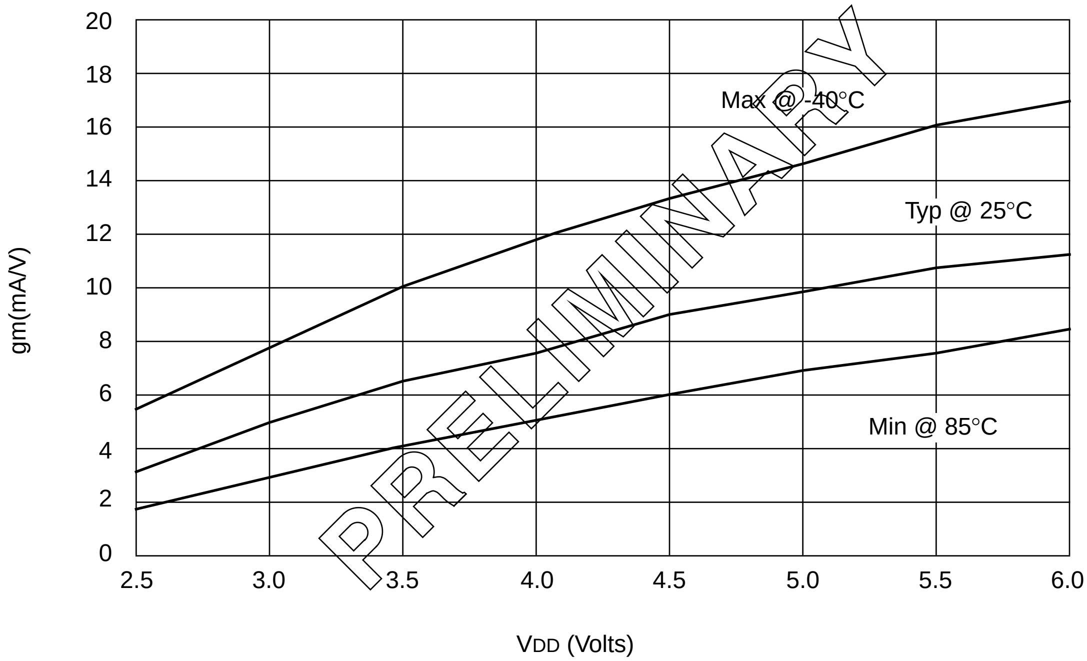


FIGURE 20-6: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD



E.6 PIC16C8X Family of Devices

	Clock	Memory					Peripherals	Features			
		Program Memory									
	Maximum Frequency of Operation (MHz)	Flash	EEPROM	ROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Packages
PIC16C84	10	—	1K	—	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F84[1]	10	1K	—	—	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR84[1]	10	—	—	1K	68	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F83[1]	10	512	—	—	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR83[1]	10	—	—	512	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC

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

All PIC16C8X 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: **PIC17C4X**　　Literature Number: **DS30412C**

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?

____

____