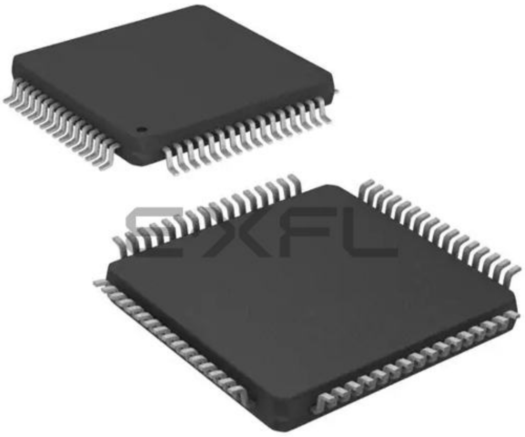
Microchip Technology - PIC17LC756A-08I/PT Datasheet



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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	902 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17lc756a-08i-pt

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.

1.0 OVERVIEW

This data sheet covers the PIC17C7XX group of the PIC17CXXX family of microcontrollers. The following devices are discussed in this data sheet:

- PIC17C752
- PIC17C756A
- PIC17C762
- PIC17C766

The PIC17C7XX devices are 68/84-pin, EPROM based members of the versatile PIC17CXXX family of low cost, high performance, CMOS, fully static, 8-bit microcontrollers.

All PIC® microcontrollers employ an advanced RISC architecture. The PIC17CXXX 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 path. 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 58 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance. For mathematical intensive applications, all devices have a single cycle 8 x 8 Hardware Multiplier.

PIC17CXXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC17C7XX devices have up to 902 bytes of RAM and 66 I/O pins. In addition, the PIC17C7XX adds several peripheral features, useful in many high performance applications, including:

- Four timer/counters
- Four capture inputs
- Three PWM outputs
- Two independent Universal Synchronous Asynchronous Receiver Transmitters (USARTs)
- An A/D converter (multi-channel, 10-bit resolution)
- A Synchronous Serial Port (SPI and I^{2}C w/ Master mode)

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. Wake-up from SLEEP can occur through several external and internal interrupts and device RESETS.

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

There are four configuration options for the device operational mode:

- Microprocessor
- Microcontroller
- Extended microcontroller
- Protected microcontroller

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

The device also has Brown-out Reset circuitry. This allows a device RESET to occur if the device VDD falls below the Brown-out voltage trip point (BVDD). The chip will remain in Brown-out Reset until VDD rises above BVDD.

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

The PIC17C7XX fits perfectly in applications that require extremely fast execution of complex software programs. These include applications ranging from precise motor control and industrial process control to automotive, instrumentation, and telecom applications.

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 (including die sales) make the PIC17C7XX 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 PIC17C7XX ideal for a wide range of embedded control applications.

1.1 Family and Upward Compatibility

The PIC17CXXX family of microcontrollers have architectural enhancements over the PIC16C5X and PIC16CXX families. These enhancements allow the device to be more efficient in software and hardware requirements. Refer to Appendix A for a detailed list of enhancements and modifications. Code written for PIC16C5X or PIC16CXX can be easily ported to PIC17CXXX devices (Appendix B).

1.2 Development Support

The PIC17CXXX 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. For additional information, see Section 19.0.

TABLE 3-1: PINOUT DESCRIPTIONS (CONTINUED)

Name	PIC17C75X			PIC17C76X		I/O/P Type	Buffer Type	Description
	DIP No.	PLCC No.	TQFP No.	PLCC No.	QFP No.			
								PORTG is a bi-directional I/O Port.
RG0/AN3	32	34	24	42	30	I/O	ST	RG0 can also be analog input 3.
RG1/AN2	31	33	23	41	29	I/O	ST	RG1 can also be analog input 2.
RG2/AN1/VREF-	30	32	22	40	28	I/O	ST	RG2 can also be analog input 1, or the ground reference voltage.
RG3/AN0/VREF+	29	31	21	39	27	I/O	ST	RG3 can also be analog input 0, or the positive reference voltage.
RG4/CAP3	35	38	27	46	33	I/O	ST	RG4 can also be the Capture3 input pin.
RG5/PWM3	36	39	28	47	34	I/O	ST	RG5 can also be the PWM3 output pin.
RG6/RX2/DT2	38	41	30	49	36	I/O	ST	RG6 can also be selected as the USART2 (SCI) Asynchronous Receive or USART2 (SCI) Synchronous Data.
RG7/TX2/CK2	37	40	29	48	35	I/O	ST	RG7 can also be selected as the USART2 (SCI) Asynchronous Transmit or USART2 (SCI) Synchronous Clock.
								PORTH is a bi-directional I/O Port. PORTH is only available on the PIC17C76X devices.
RH0	—	—	—	10	79	I/O	ST	
RH1	—	—	—	11	80	I/O	ST	
RH2	—	—	—	12	1	I/O	ST	
RH3	—	—	—	13	2	I/O	ST	
RH4/AN12	—	—	—	31	19	I/O	ST	RH4 can also be analog input 12.
RH5/AN13	—	—	—	32	20	I/O	ST	RH5 can also be analog input 13.
RH6/AN14	—	—	—	33	21	I/O	ST	RH6 can also be analog input 14.
RH7/AN15	—	—	—	34	22	I/O	ST	RH7 can also be analog input 15.
								PORTJ is a bi-directional I/O Port. PORTJ is only available on the PIC17C76X devices.
RJ0	—	—	—	52	39	I/O	ST	
RJ1	—	—	—	53	40	I/O	ST	
RJ2	—	—	—	54	41	I/O	ST	
RJ3	—	—	—	55	42	I/O	ST	
RJ4	—	—	—	73	59	I/O	ST	
RJ5	—	—	—	74	60	I/O	ST	
RJ6	—	—	—	75	61	I/O	ST	
RJ7	—	—	—	76	62	I/O	ST	
TEST	16	17	8	21	10	I	ST	Test mode selection control input. Always tie to VSS for normal operation.
VSS	17, 33, 49, 64	19, 36, 53, 68	9, 25, 41, 56	23, 44, 65, 84	11, 31, 51, 70	P		Ground reference for logic and I/O pins.
VDD	1, 18, 34, 46	2, 20, 37, 49,	10, 26, 38, 57	24, 45, 61, 2	12, 32, 48, 71	P		Positive supply for logic and I/O pins.
AVSS	28	30	20	38	26	P		Ground reference for A/D converter. This pin **MUST** be at the same potential as VSS.
AVDD	27	29	19	37	25	P		Positive supply for A/D converter. This pin **MUST** be at the same potential as VDD.
NC	—	1, 18, 35, 52	—	1, 22, 43, 64	—			No Connect. Leave these pins unconnected.

Legend: I = Input only; O = Output only; I/O = Input/Output;
P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input

Note 1: The output is only available by the peripheral operation.

2: Open drain input/output pin. Pin forced to input upon any device RESET.

7.7 Program Counter Module

The Program Counter (PC) is a 16-bit register. PCL, the low byte of the PC, is mapped in the data memory. PCL is readable and writable just as is any other register. PCH is the high byte of the PC and is not directly addressable. Since PCH is not mapped in data or program memory, an 8-bit register PCLATH (PC high latch) is used as a holding latch for the high byte of the PC. PCLATH is mapped into data memory. The user can read or write PCH through PCLATH.

The 16-bit wide PC is incremented after each instruction fetch during Q1 unless:

- Modified by a `GOTO`, `CALL`, `LCALL`, `RETURN`, `RETLW`, or `RETFIE` instruction
- Modified by an interrupt response
- Due to destination write to PCL by an instruction

"Skips" are equivalent to a forced `NOP` cycle at the skipped address.

Figure 7-7 and Figure 7-8 show the operation of the program counter for various situations.

FIGURE 7-7: PROGRAM COUNTER OPERATION

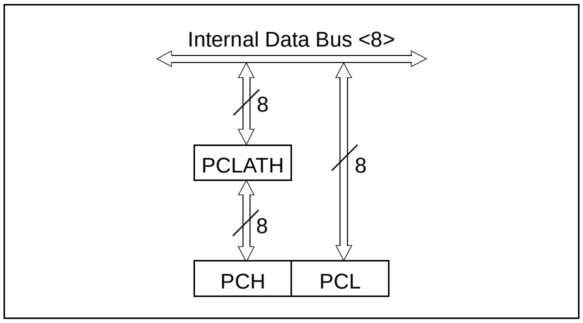
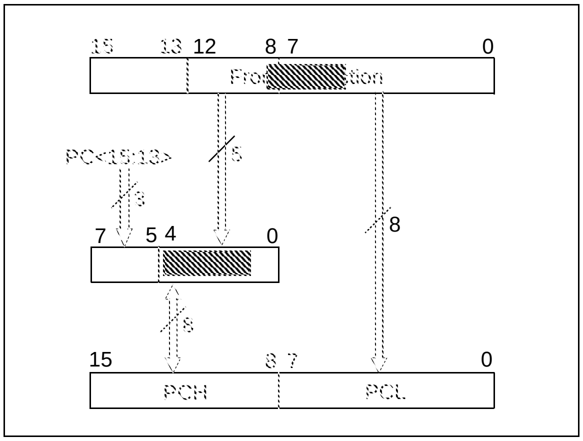


FIGURE 7-8: PROGRAM COUNTER USING THE `CALL` AND `GOTO` INSTRUCTIONS



Using Figure 7-7, the operations of the PC and PCLATH for different instructions are as follows:

a) `LCALL` instructions:

An 8-bit destination address is provided in the instruction (opcode). PCLATH is unchanged.

PCLATH → PCH

Opcode<7:0> → PCL

b) Read instructions on PCL:

Any instruction that reads PCL.

PCL → data bus → ALU or destination

PCH → PCLATH

c) Write instructions on PCL:

Any instruction that writes to PCL.

8-bit data → data bus → PCL

PCLATH → PCH

d) Read-Modify-Write instructions on PCL:

Any instruction that does a read-write-modify operation on PCL, such as `ADDWF PCL`.

Read: PCL → data bus → ALU

Write: 8-bit result → data bus → PCL

PCLATH → PCH

e) `RETURN` instruction:

Stack<MRU> → PC<15:0>

Using Figure 7-8, the operation of the PC and PCLATH for `GOTO` and `CALL` instructions is as follows:

`CALL, GOTO` instructions:

A 13-bit destination address is provided in the instruction (opcode).

Opcode<12:0> → PC<12:0>

PC<15:13> → PCLATH<7:5>

Opcode<12:8> → PCLATH<4:0>

The read-modify-write only affects the PCL with the result. PCH is loaded with the value in the PCLATH. For example, `ADDWF PCL` will result in a jump within the current page. If PC = 03F0h, WREG = 30h and PCLATH = 03h before instruction, PC = 0320h after the instruction. To accomplish a true 16-bit computed jump, the user needs to compute the 16-bit destination address, write the high byte to PCLATH and then write the low value to PCL.

The following PC related operations do not change PCLATH:

a) `LCALL`, `RETLW`, and `RETFIE` instructions.

b) Interrupt vector is forced onto the PC.

c) Read-modify-write instructions on PCL (e.g. `BSF PCL`).

NOTES:

TABLE 10-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	Function
RB0/CAP1	bit0	ST	Input/output or the Capture1 input pin. Software programmable weak pull-up and interrupt-on-change features.
RB1/CAP2	bit1	ST	Input/output or the Capture2 input pin. Software programmable weak pull-up and interrupt-on-change features.
RB2/PWM1	bit2	ST	Input/output or the PWM1 output pin. Software programmable weak pull-up and interrupt-on-change features.
RB3/PWM2	bit3	ST	Input/output or the PWM2 output pin. Software programmable weak pull-up and interrupt-on-change features.
RB4/TCLK12	bit4	ST	Input/output or the external clock input to Timer1 and Timer2. Software programmable weak pull-up and interrupt-on-change features.
RB5/TCLK3	bit5	ST	Input/output or the external clock input to Timer3. Software programmable weak pull-up and interrupt-on-change features.
RB6/SCK	bit6	ST	Input/output or the Master/Slave clock for the SPI. Software programmable weak pull-up and interrupt-on-change features.
RB7/SDO	bit7	ST	Input/output or data output for the SPI. Software programmable weak pull-up and interrupt-on-change features.

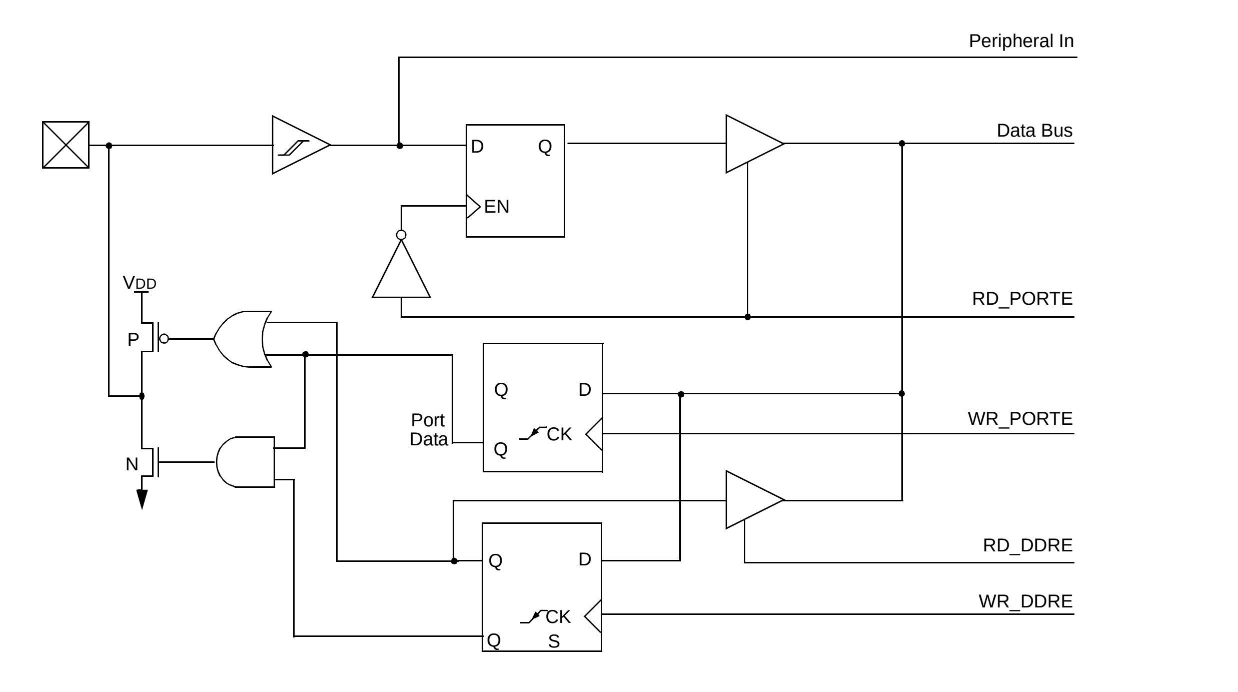
Legend: ST = Schmitt Trigger input

TABLE 10-4: REGISTERS/BITS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
12h, Bank 0	PORTB	RB7/ SDO	RB6/ SCK	RB5/ TCLK3	RB4/ TCLK12	RB3/ PWM2	RB2/ PWM1	RB1/ CAP2	RB0/ CAP1	xxxx xxxx	uuuu uuuu
11h, Bank 0	DDRB	Data Direction Register for PORTB								1111 1111	1111 1111
10h, Bank 0	PORTA	$\overline{\text{RBPU}}$	—	RA5/ TX1/CK1	RA4/ RX1/DT1	RA3/ SDI/SDA	RA2/ $\overline{\text{SS}}$/SCL	RA1/T0CKI	RA0/INT	0-xx 11xx	0-uu 11uu
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	TO	PD	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--11 11qq	--11 qquu
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
16h, Bank 1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank 1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{\text{PR3}}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends on condition. Shaded cells are not used by PORTB.

FIGURE 10-12: BLOCK DIAGRAM OF RE3/CAP4 PORT PIN



Note: I/O pin has protection diodes to VDD and VSS.

TABLE 10-9: PORTE FUNCTIONS

Name	Bit	Buffer Type	Function
RE0/$\overline{\text{ALE}}$	bit0	TTL	Input/output or system bus Address Latch Enable ($\overline{\text{ALE}}$) control pin.
RE1/$\overline{\text{OE}}$	bit1	TTL	Input/output or system bus Output Enable ($\overline{\text{OE}}$) control pin.
RE2/$\overline{\text{WR}}$	bit2	TTL	Input/output or system bus Write ($\overline{\text{WR}}$) control pin.
RE3/CAP4	bit3	ST	Input/output or Capture4 input pin.

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

TABLE 10-10: REGISTERS/BITS 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	$\overline{\text{MCLR}}$, WDT
15h, Bank 1	PORTE	—	—	—	—	RE3/CAP4	RE2/$\overline{\text{WR}}$	RE1/$\overline{\text{OE}}$	RE0/ALE	`---- xxxx`	`---- uuuu`
14h, Bank 1	DDRE	Data Direction Register for PORTE								`---- 1111`	`---- 1111`
14h, Bank 7	CA4L	Capture4 Low Byte								`xxxx xxxx`	`uuuu uuuu`
15h, Bank 7	CA4H	Capture4 High Byte								`xxxx xxxx`	`uuuu uuuu`
16h, Bank 7	TCON3	—	CA4OVF	CA3OVF	CA4ED1	CA4ED0	CA3ED1	CA3ED0	PWM3ON	`-000 0000`	`-000 0000`

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

13.1.3 USING PULSE WIDTH MODULATION (PWM) OUTPUTS WITH TIMER1 AND TIMER2

Three high speed pulse width modulation (PWM) outputs are provided. The PWM1 output uses Timer1 as its time base, while PWM2 and PWM3 may independently be software configured to use either Timer1 or Timer2 as the time base. The PWM outputs are on the RB2/PWM1, RB3/PWM2 and RG5/PWM3 pins.

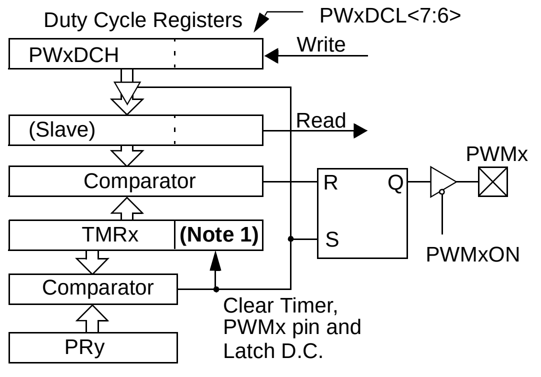
Each PWM output has a maximum resolution of 10-bits. At 10-bit resolution, the PWM output frequency is 32.2 kHz (@ 32 MHz clock) and at 8-bit resolution the PWM output frequency is 128.9 kHz. The duty cycle of the output can vary from 0% to 100%.

Figure 13-3 shows a simplified block diagram of a PWM module.

The duty cycle registers are double buffered for glitch free operation. Figure 13-4 shows how a glitch could occur if the duty cycle registers were not double buffered.

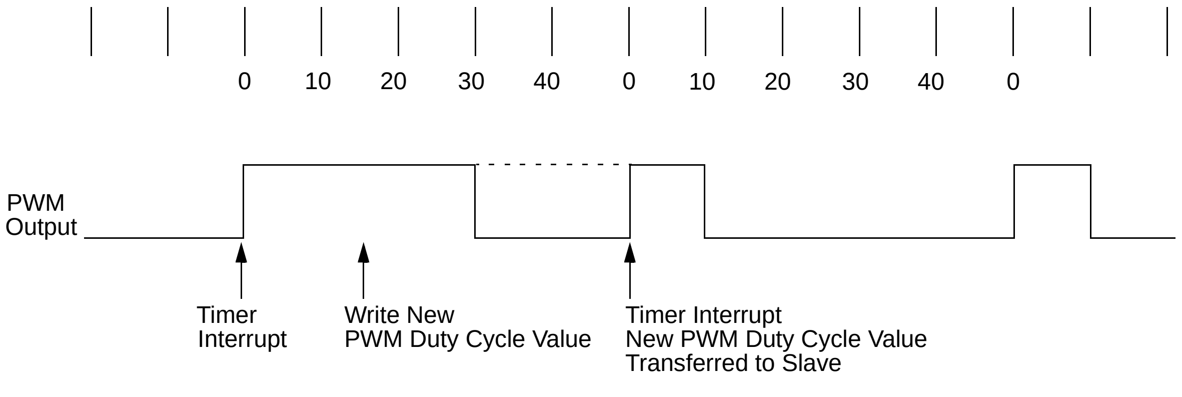
The user needs to set the PWM1ON bit (TCON2<4>) to enable the PWM1 output. When the PWM1ON bit is set, the RB2/PWM1 pin is configured as PWM1 output and forced as an output, irrespective of the data direction bit (DDRB<2>). When the PWM1ON bit is clear, the pin behaves as a port pin and its direction is controlled by its data direction bit (DDRB<2>). Similarly, the PWM2ON (TCON2<5>) bit controls the configuration of the RB3/PWM2 pin and the PWM3ON (TCON3<0>) bit controls the configuration of the RG5/PWM3 pin.

FIGURE 13-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

FIGURE 13-4: PWM OUTPUT (NOT BUFFERED)

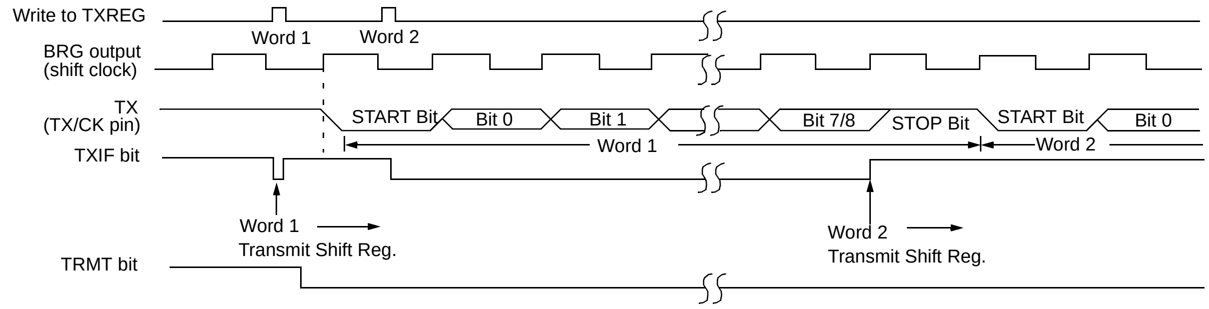

0
10
20
30
40
0
10
20
30
40
0
PWM Output
Timer Interrupt
Write New PWM Duty Cycle Value
Timer Interrupt New PWM Duty Cycle Value Transferred to Slave

Note: The dotted line shows PWM output if duty cycle registers were not double buffered. If the new duty cycle is written after the timer has passed that value, then the PWM does not reset at all during the current cycle, causing a "glitch".

In this example, PWM period = 50. Old duty cycle is 30. New duty cycle value is 10.

NOTES:

FIGURE 14-4: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 14-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank 1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank 1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
13h, Bank 0	RCSTA1	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
16h, Bank 0	TXREG1	Serial Port Transmit Register (USART1)								xxxx xxxx	uuuu uuuu
15h, Bank 0	TXSTA1	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG1	Baud Rate Generator Register (USART1)								0000 0000	0000 0000
10h, Bank 4	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h, Bank 4	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
13h, Bank 4	RCSTA2	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
16h, Bank 4	TXREG2	Serial Port Transmit Register (USART2)								xxxx xxxx	uuuu uuuu
15h, Bank 4	TXSTA2	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 4	SPBRG2	Baud Rate Generator Register (USART2)								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as a '0'. Shaded cells are not used for asynchronous transmission.

17.4.2 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.

17.5 Code Protection

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

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.

BSF — Bit Set f

Syntax: [*label*] BSF f,b

Operands: 0 ≤ f ≤ 255
0 ≤ b ≤ 7

Operation: 1 → (f<b>)

Status Affected: None

Encoding:

`1000`	`0bbb`	`ffff`	`ffff`

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:

```
BSF     FLAG_REG, 7
```

Before Instruction
FLAG_REG = 0x0A

After Instruction
FLAG_REG = 0x8A

BTFSC — Bit Test, skip if Clear

Syntax: [*label*] BTFSC f,b

Operands: 0 ≤ f ≤ 255
0 ≤ b ≤ 7

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

Status Affected: None

Encoding:

`1001`	`1bbb`	`ffff`	`ffff`

Description: If bit 'b' in register ’f' is 0, then the next instruction is skipped.

If bit 'b' is 0, then the next instruction fetched during the current instruction execution is discarded and a `NOP` is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

Example:

```
HERE    BTFSC   FLAG,1
FALSE   :
TRUE    :
```

Before Instruction
PC = address (`HERE`)

After Instruction
If FLAG<1> = 0;
PC = address (`TRUE`)
If FLAG<1> = 1;
PC = address (`FALSE`)

NEGW — Negate W

Syntax: [*label*] NEGW f,s

Operands: $0 \le f \le 255$
$s \in [0,1]$

Operation: $\overline{\text{WREG}} + 1 \rightarrow (f)$;
$\text{WREG} + 1 \rightarrow s$

Status Affected: OV, C, DC, Z

Encoding:

0010	110s	ffff	ffff

Description: WREG is negated using two's complement. If 's' is 0, the result is placed in WREG and data memory location 'f'. If 's' is 1, the result is placed only in data memory location 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f' and other specified register

Example: `NEGW REG,0`

Before Instruction
WREG = `0011 1010` [0x3A],
REG = `1010 1011` [0xAB]

After Instruction
WREG = `1100 0110` [0xC6]
REG = `1100 0110` [0xC6]

NOP — No Operation

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

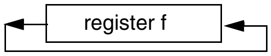
RLNCF — Rotate Left f (no carry)

Syntax:	[*label*] RLNCF f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	f<n> → d<n+1>; f<7> → d<0>
Status Affected:	None

Encoding:

`0010`	`001d`	`ffff`	`ffff`

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is 0, the result is placed 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'	Process Data	Write to destination

Example: `RLNCF REG, 1`

Before Instruction

C	=	`0`
REG	=	`1110 1011`

After Instruction

C	=	
REG	=	`1101 0111`

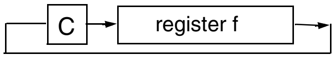
RRCF — Rotate Right f through Carry

Syntax:	[*label*] RRCF f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	f<n> → d<n-1>; f<0> → C; C → d<7>
Status Affected:	C

Encoding:

`0001`	`100d`	`ffff`	`ffff`

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `RRCF REG1,0`

Before Instruction

REG1	=	`1110 0110`
C	=	`0`

After Instruction

REG1	=	`1110 0110`
WREG	=	`0111 0011`
C	=	`0`

19.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - ICEPIC™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD for PIC16F87X
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM 2 Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 17 Demonstration Board
  - KEELOQ® Demonstration Board

19.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. The MPLAB IDE is a Windows®-based application that contains:

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor
- A project manager
- Customizable toolbar and key mapping
- A status bar
- On-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or ‘C’)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost-effective simulator to a full-featured emulator with minimal retraining.

19.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PIC MCU’s.

The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. The MPASM assembler generates relocatable object files for the MPLINK object linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file that contains source lines and generated machine code, and a COD file for debugging.

The MPASM assembler features include:

- Integration into MPLAB IDE projects.
- User-defined macros to streamline assembly code.
- Conditional assembly for multi-purpose source files.
- Directives that allow complete control over the assembly process.

19.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI ‘C’ compilers for Microchip’s PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

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

FIGURE 20-3: PIC17LC7XX-08 VOLTAGE-FREQUENCY GRAPH

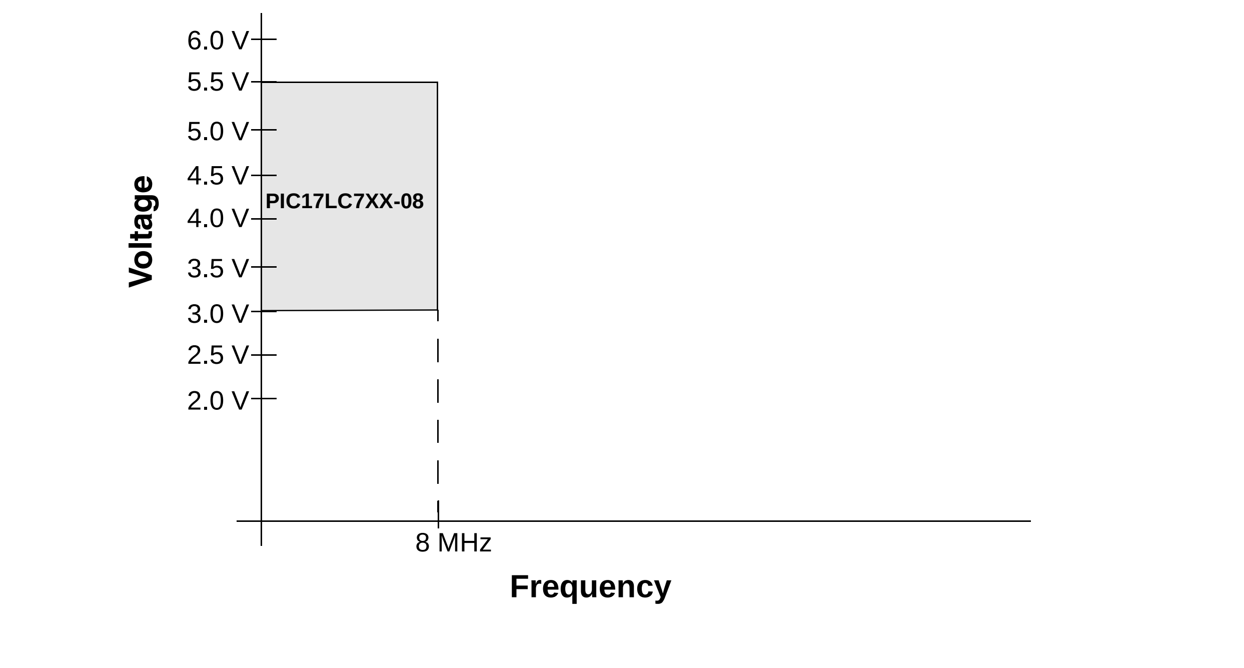
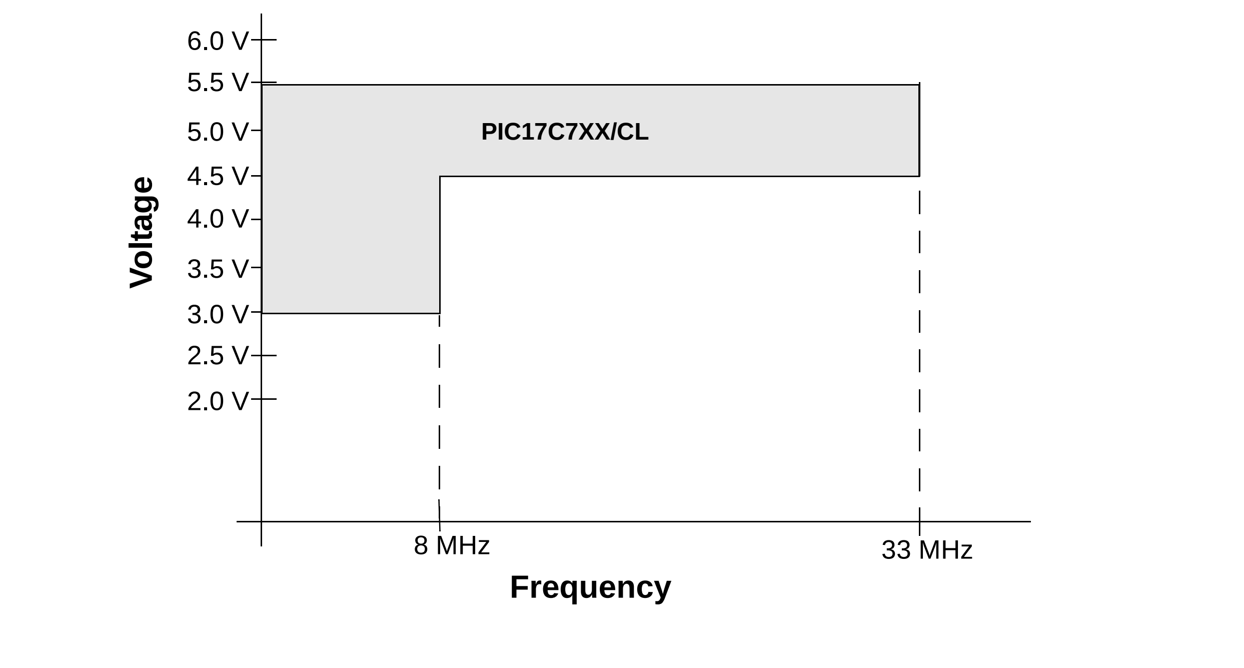


FIGURE 20-4: PIC17C7XX/CL VOLTAGE-FREQUENCY GRAPH



20.3 Timing Parameter Symbology

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

1. TppS2ppS		3. TCC:ST	(I^2C specifications only)
2. TppS		4. Ts	(I^2C specifications only)

T			
F	Frequency	T	Time

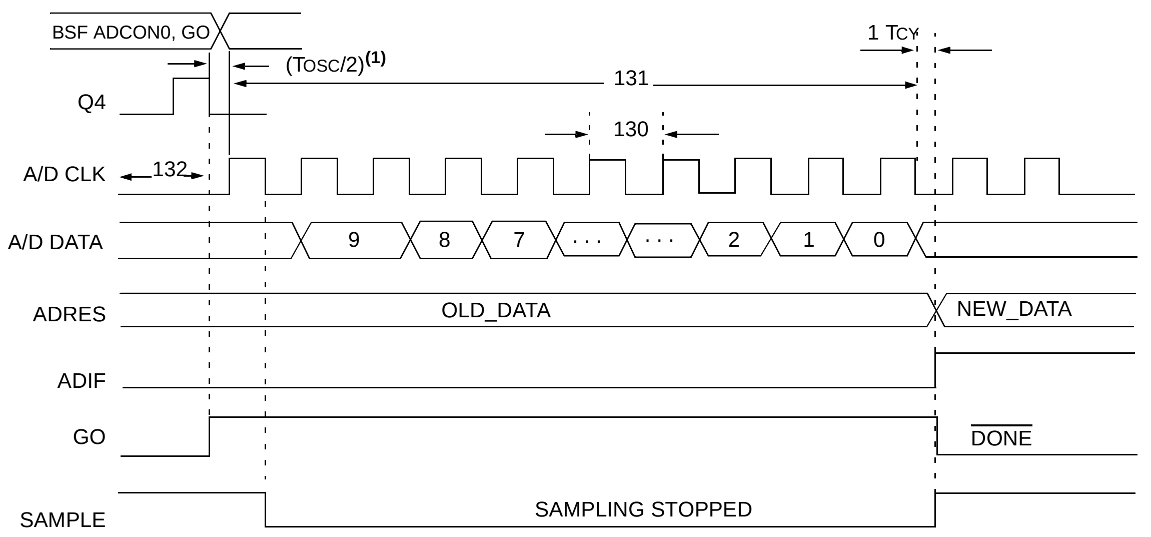
Lowercase symbols (pp) and their meanings:

pp			
ad	Address/Data	ost	Oscillator Start-Up Timer
al	ALE	pwrt	Power-Up Timer
cc	Capture1 and Capture2	rb	PORTB
ck	CLKOUT or clock	rd	$\overline{\text{RD}}$
dt	Data in	rw	$\overline{\text{RD}}$ or $\overline{\text{WR}}$
in	INT pin	t0	T0CKI
io	I/O port	t123	TCLK12 and TCLK3
mc	$\overline{\text{MCLR}}$	wdt	Watchdog Timer
oe	$\overline{\text{OE}}$	wr	$\overline{\text{WR}}$
os	OSC1		

Uppercase symbols and their meanings:

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

FIGURE 20-23: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

TABLE 20-19: A/D CONVERSION REQUIREMENTS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC17**C**XXX	1.6	—	—	µs	TOSC based, VREF ≥ 3.0V
			PIC17**LC**XXX	3.0	—	—	µs	TOSC based, VREF full range
			PIC17**C**XXX	2.0	4.0	6.0	µs	A/D RC mode
			PIC17**LC**XXX	3.0	6.0	9.0	µs	A/D RC mode
131	TCNV	Conversion time (not including acquisition time) **(Note 1)**		11	—	12	Tad	
132	TACQ	Acquisition time		**(Note 2)** 10	20 —	— —	µs µs	 The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1LSb (i.e., 5 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
134	TGO	Q4 to ADCLK start		—	Tosc/2	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated.

Note 1: ADRES register may be read on the following TCY cycle.

2: See Section 16.1 for minimum conditions when input voltage has changed more than 1 LSb.

FIGURE 21-9: TYPICAL IDD vs. FOSC OVER VDD (XT MODE)

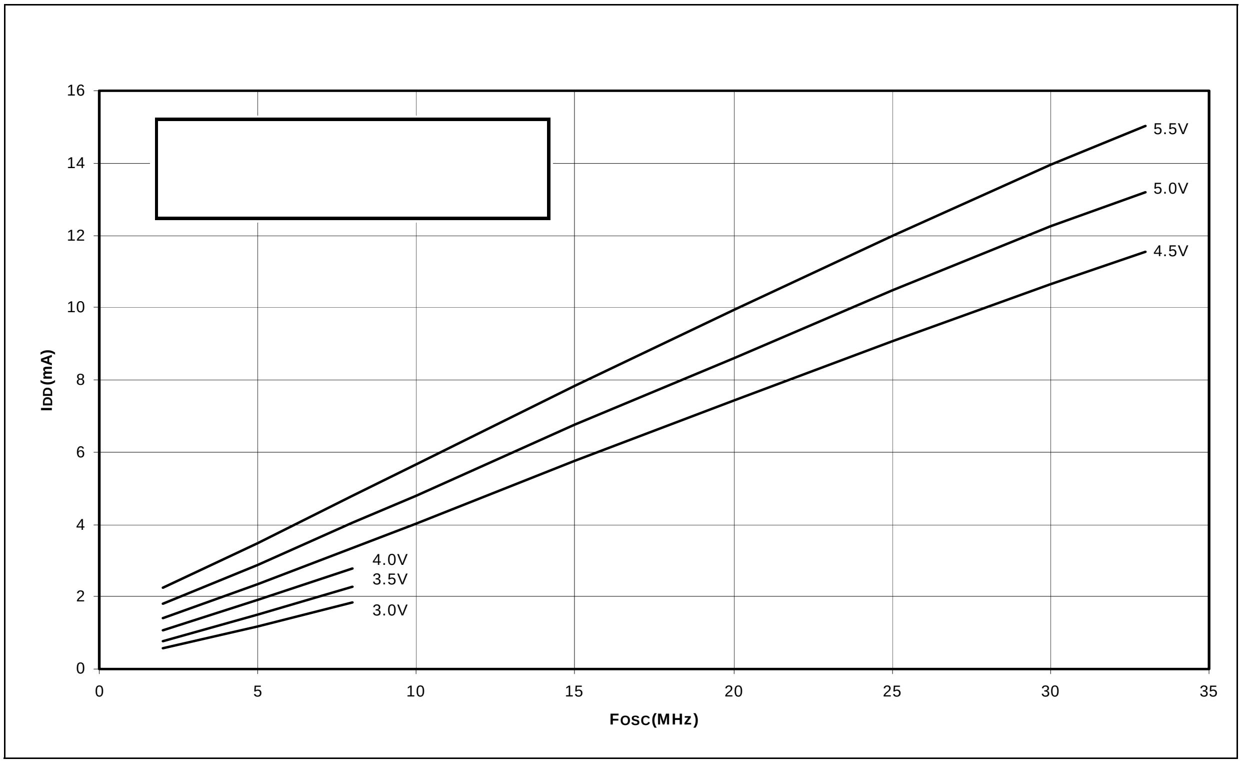
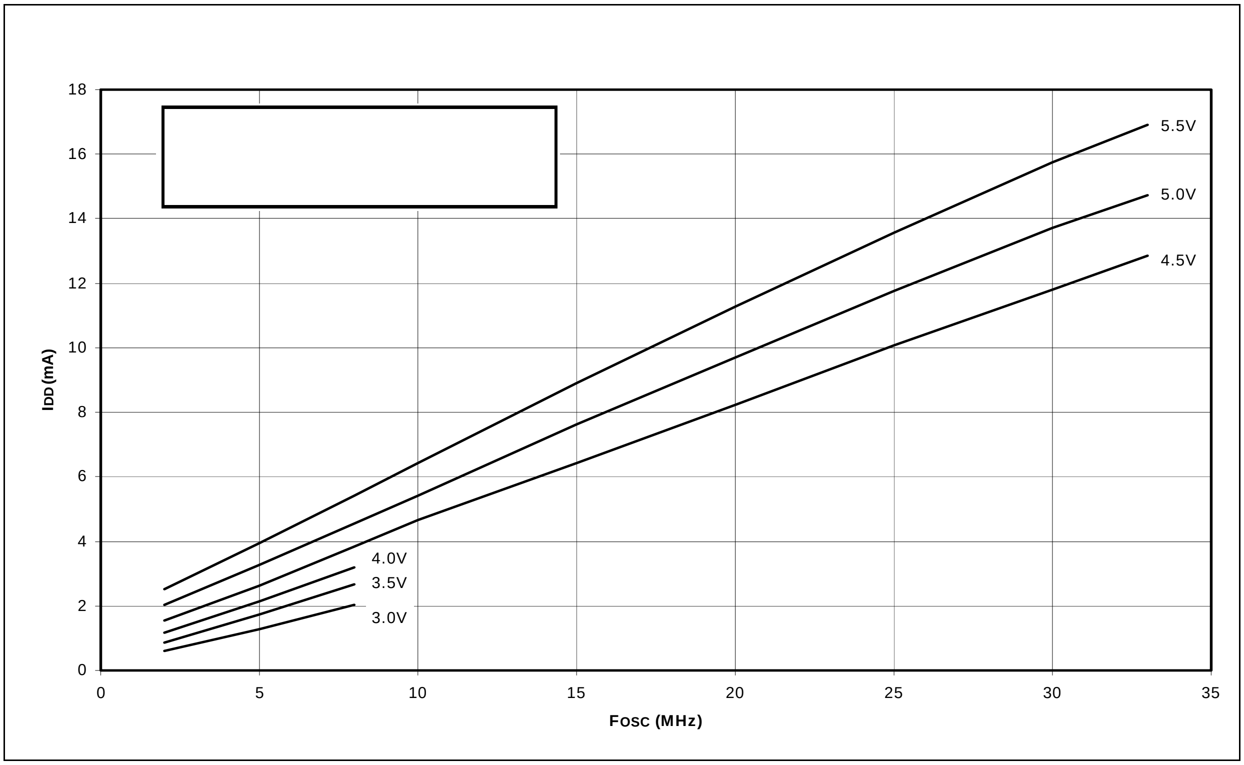


FIGURE 21-10: MAXIMUM IDD vs. FOSC OVER VDD (XT MODE)


IDD (mA)
FOSC (MHz)
5.5V
5.0V
4.5V
4.0V
3.5V
3.0V