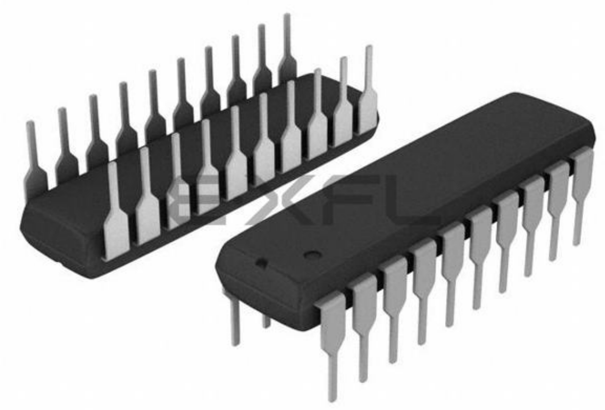


Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	15
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 6x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc771-e-p

2.2.2.4 PIE1 REGISTER

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

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (PIE1: 8Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **Unimplemented:** Read as ’0’

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

bit 5-4 **Unimplemented:** Read as ’0’

bit 3 **SSPIE:** Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

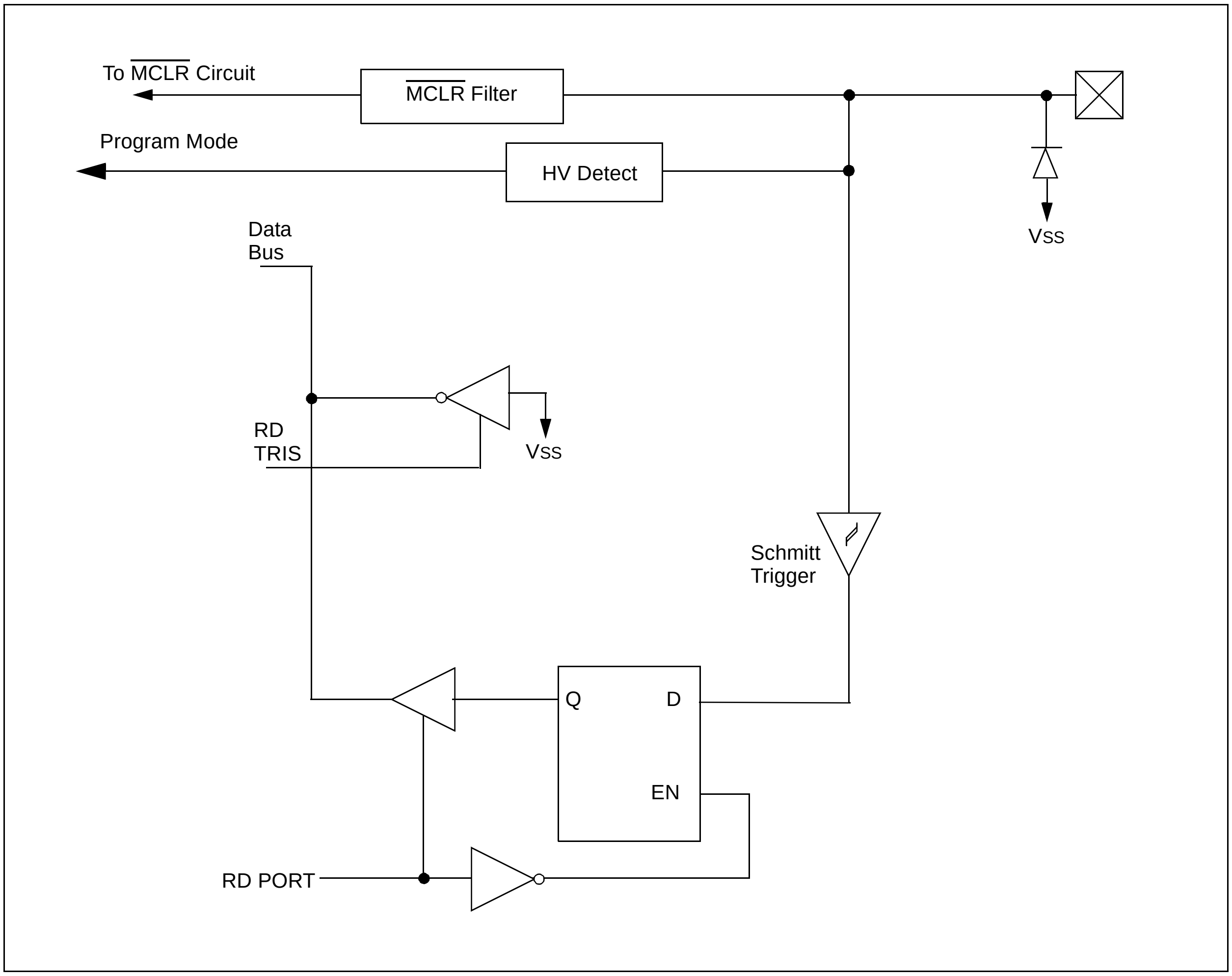
bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

FIGURE 3-4: BLOCK DIAGRAM OF RA5/$\overline{\text{MCLR}}$/VPP



4.3 READING THE EPROM PROGRAM MEMORY

To read a program memory location, the user must write 2 bytes of the address to the PMADRH and PMADRL registers, then set control bit RD (PMCON1<0>). Once the read control bit is set, the Program Memory Read (PMR) controller will use the second instruction cycle after to read the data. This causes the second instruction immediately following the "BSF PMCON1,RD" instruction to be ignored. The data is available, in the very next cycle, in the PMDATH and PMDATL registers; therefore it can be read as 2 bytes in the following instructions. PMDATH and PMDATL registers will hold this value until another Program Memory Read or until it is written to by the user.

Note:	The two instructions that follow setting the PMCON1 read bit must be NOPs.

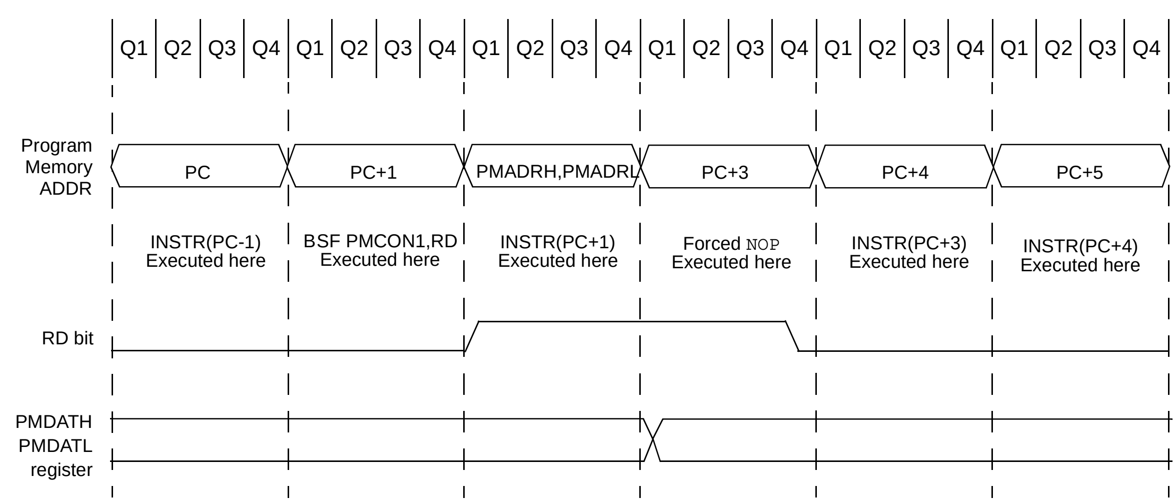
EXAMPLE 4-1: OTP PROGRAM MEMORY Read

```
    BSF     STATUS, RP1      ;
    BCF     STATUS, RP0      ; Bank 2
    MOVLW   MS_PROG_PM_ADDR  ;
    MOVWF   PMADRH           ; MS Byte of Program Memory Address to read
    MOVLW   LS_PROG_PM_ADDR  ;
    MOVWF   PMADRL           ; LS Byte of Program Memory Address to read
    BSF     STATUS, RP0      ; Bank 3
    BSF     PMCON1, RD       ; Program Memory Read
    NOP                      ; This instruction must be an NOP
    NOP                      ; This instruction must be an NOP
    next instruction         ; PMDATH:PMDATL now has the data
```

4.4 OPERATION DURING CODE PROTECT

When the device is code protected, the CPU can still perform the Program Memory Read function.

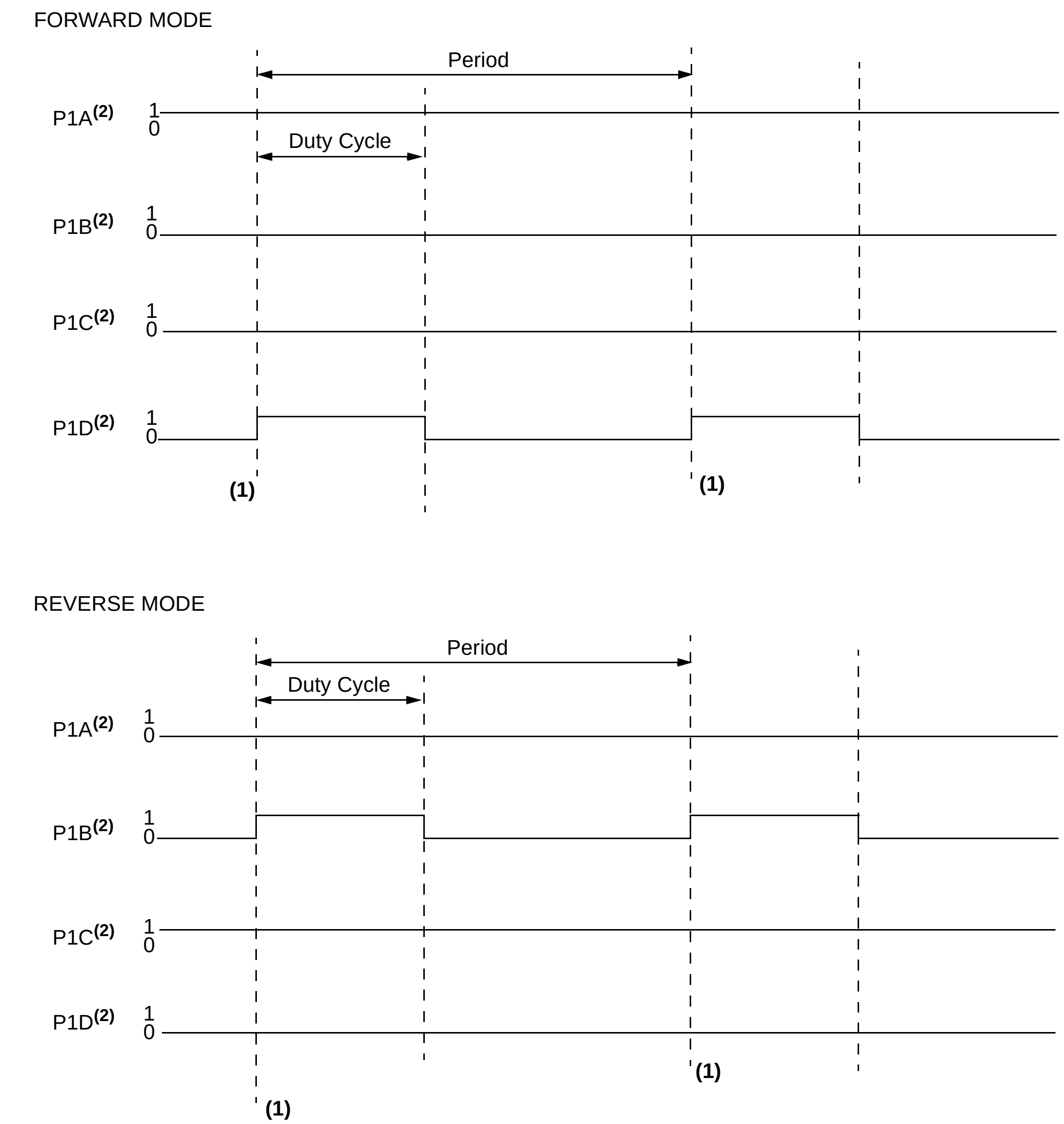
FIGURE 4-1: PROGRAM MEMORY READ CYCLE EXECUTION



In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, RB3/CCP1/P1A pin is continuously active, and RB7/T1OSI/P1D pin is modulated. In the Reverse mode, RB6/T1OSO/T1CKI/P1C pin is continuously active, and RB5/SDO/P1B pin is modulated.

P1A, P1B, P1C and P1D outputs are multiplexed with PORTB<3> and PORTB<5:7> data latches. TRISB<3> and TRISB<5:7> bits must be cleared to make the P1A, P1B, P1C, and P1D pins output.

FIGURE 8-8: FULL-BRIDGE PWM OUTPUT



Note 1: At this time, the TMR2 register is equal to the PR2 register.
2: Output signal is shown as asserted high.

9.2.11 I^{2}C MASTER MODE REPEATED START CONDITION TIMING

A Repeated START condition occurs when the RSEN bit (SSPCON2<1>) is set high while the I^{2}C module is in the idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<6:0> and begins counting. The SDA pin is released (brought high) for one baud rate generator count (TBRG). When the baud rate generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG period. This action is then followed by assertion of the SDA pin (SDA is low) for one TBRG period while SCL is high. As soon as a START condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. Following this, the baud rate generator is reloaded with the contents of SSPAD<6:0> and begins counting. When the BRG times out a third time, the RSEN bit in the SSPCON2 register is automatically cleared and SCL is pulled low. The SSPIF flag is set, which indicates the Restart sequence is complete.

> **Note 1:** If RSEN is set while another event is in progress, it will not take effect. Queuing of events is not allowed.
>
> **2:** A bus collision during the Repeated START condition occurs if either of the following is true:
>
> a) SDA is sampled low when SCL goes from low to high.
>
> b) SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data "1".

Immediately following the SSPIF bit transition to true, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an $\overline{\text{ACK}}$ is received, the user may then perform one of the following:

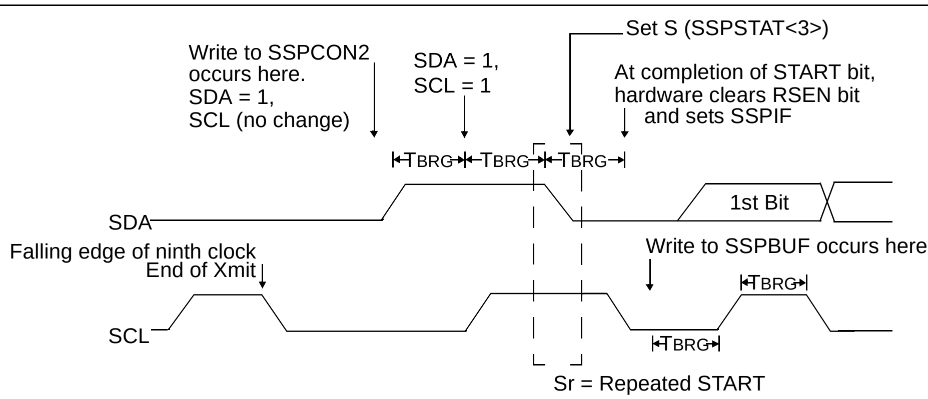
- Transmit an additional eight bits of address (if the user transmitted the first half of a 10-bit address with R/$\overline{\text{W}}$ = 0),
- Transmit eight bits of data (if the user transmitted a 7-bit address with R/$\overline{\text{W}}$ = 0), or
- Receive eight bits of data (if the user transmitted either the first half of a 10-bit address or a 7-bit address with R/$\overline{\text{W}}$ = 1).

9.2.11.1 WCOL STATUS FLAG

If the user writes the SSPBUF when a Repeated START sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

> **Note:** Because queueing of events is not allowed, writing of the lower five bits of SSPCON2 is disabled until the Repeated START condition is complete.

FIGURE 9-17: REPEAT START CONDITION WAVEFORM



9.2.13 I²C MASTER MODE RECEPTION

In Master-receive mode, the first byte transmitted contains seven bits of address data and the R/$\overline{\text{W}}$ bit. In this case, the R/$\overline{\text{W}}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. The START condition indicates the beginning of a transmission. The master-receiver terminates slave transmission by responding to the last byte with a NACK Acknowledge and follows this with a STOP condition to indicate to other masters that the bus is free.

Master mode reception is enabled by setting the receive enable bit, RCEN (SSPCON2<3>), immediately following the Acknowledge sequence.

Note:	The MSSP Module must be in an IDLE STATE before the RCEN bit is set or the RCEN bit will be disregarded.

The baud rate generator begins counting, and on each rollover, the state of the SCL pin changes (high to low/ low to high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the following events occur:

- The receive enable bit is automatically cleared.
- The contents of the SSPSR are loaded into the SSPBUF.
- The BF flag is set.
- The SSPIF is set.
- The baud rate generator is suspended from counting, holding SCL low.

The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an Acknowledge bit at the end of reception by clearing the ACKDT bit (SSPCON2<5>) and setting the Acknowledge sequence enable bit, ACKEN (SSPCON2<4>).

A typical receive sequence would go as follows:

a) The user generates a START Condition by setting the START enable bit (SEN) in SSPCON2.
b) SSPIF is set at the completion of the START sequence.
c) The user resets the SSPIF bit and loads the SSPBUF with seven bits of address in the MSbs and the LSb (R/$\overline{\text{W}}$ bit) set to '1' for receive.
d) Address and R/$\overline{\text{W}}$ is shifted out the SDA pin until all eight bits are transmitted.
e) The MSSP Module shifts in the $\overline{\text{ACK}}$ bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
f) The module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
g) The user resets the SSPIF bit and sets the RCEN bit to enable reception.
h) DATA is shifted into the SDA pin until all eight bits are received.
i) The MSSP module sets the SSPIF bit and clears the RCEN bit at the falling edge of the eighth clock.
j) The user resets the SSPIF bit and sets the ACKDT bit to '0' ($\overline{\text{ACK}}$), if another byte is anticipated. Otherwise, the ACKDT bit is set to '1' (NACK) to terminate reception. The user sets ADKEN to start the Acknowledge sequence.
k) The MSSP module sets the SSPIF bit at the completion of the Acknowledge.
l) If a NACK was sent in step (j), then the user proceeds with step (m). Otherwise, reception continues by repeating steps (g) through (j).
m) The user generates a STOP condition by setting the STOP enable bit PEN in SSPCON2.
n) SSPIF is set when the STOP condition is complete.

9.2.13.1 BF STATUS FLAG

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared by hardware when SSPBUF is read.

9.2.13.2 SSPOV STATUS FLAG

In receive operation, SSPOV is set when eight bits are received into the SSPSR and the BF flag is already set from a previous reception.

9.2.13.3 WCOL STATUS FLAG

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 9-25: BUS COLLISION DURING START CONDITION (SCL = 0)

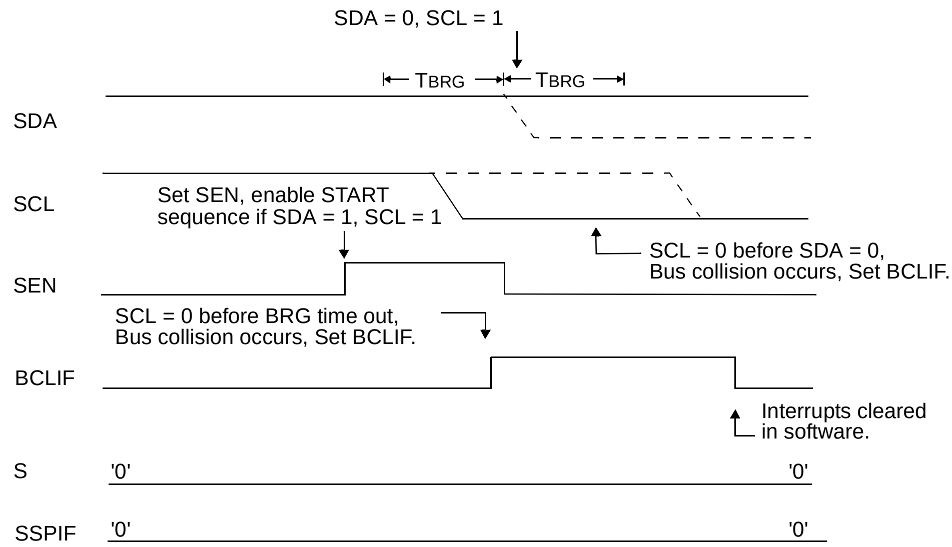
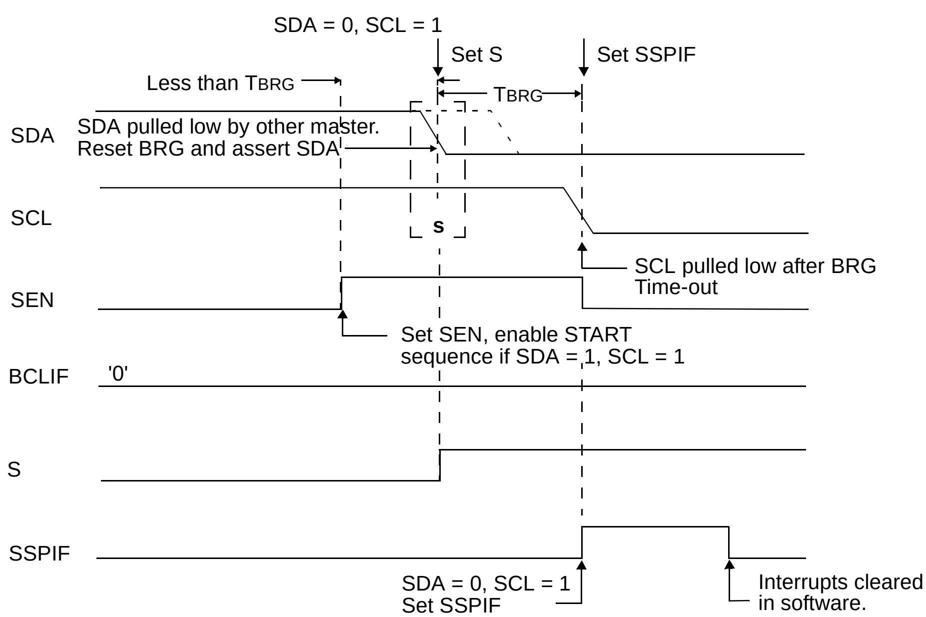


FIGURE 9-26: BRG RESET DUE TO SDA COLLISION DURING START CONDITION



 Advance Information

11.3 Selecting the A/D Conversion Clock

The A/D conversion cycle requires 13TAD: 1 TAD for settling time, and 12 TAD for conversion. The source of the A/D conversion clock is software selected. If neither the internal VRH nor VRL are used for the A/D converter, the four possible options for TAD are:

- 2 TOSC
- 8 TOSC
- 32 TOSC
- A/D RC oscillator

If the VRH or VRL are used for the A/D converter reference, then the TAD requirement is automatically increased by a factor of 8.

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 µs. Table 11-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

The ADIF bit is set on the rising edge of the 14th TAD. The GO/DONE bit is cleared on the falling edge of the 14th TAD.

TABLE 11-1: TAD VS. DEVICE OPERATING FREQUENCIES

A/D Reference Source	A/D Clock Source (TAD)		Device Frequency			
	Operation	**ADCS<1:0>**	**20 MHz**	**5 MHz**	**4 MHz**	**1.25 MHz**
External VREF or Analog Supply	2 TOSC	00	100 ns(2)	400 ns(2)	500 ns(2)	1.6 µs
	8 TOSC	01	400 ns(2)	1.6 µs	2.0 µs	6.4 µs
	32 TOSC	10	1.6 µs	6.4 µs(3)	8.0 µs(3)	25.6 µs(3)
	A/D RC	11	2 - 6 µs(1,4)	2 - 6 µs(1,4)	2 - 6 µs(1,4)	2 - 6 µs(1,4)
Internal VRH or VRL	16 TOSC	00	800 ns(2)	3.2 µs(2)	4 µs(2)	12.8 µs
	64 TOSC	01	3.2 µs(2)	12.8 µs	16 µs	51.2 µs(3)
	256 TOSC	10	12.8 µs	51.2 µs(3)	64 µs(3)	204.8 µs(3)
	A/D RC	11	16 - 48 µs(4,5)	16 - 48 µs(4,5)	16 - 48 µs(4,5)	16 - 48 µs(4,5)

Legend: Shaded cells are outside of recommended range.

Note 1: The A/D RC source has a typical TAD time of 4 µs for VDD > 3.0V.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

4: When the device frequency is greater than 1 MHz, the A/D RC clock source is only recommended if the conversion will be performed during SLEEP.

5: A/D RC clock source has a typical TAD time of 32 µs for VDD > 3.0V.

TABLE 12-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Power-on Reset or Brown-out Reset	$\overline{MCLR}$ Reset or WDT Reset	Wake-up via WDT or Interrupt
P1DEL	0000 0000	0000 0000	uuuu uuuu
REFCON	0000 ----	0000 ----	uuuu ----
LVDCON	--00 0101	--00 0101	--uu uuuu
ANSEL	--11 1111	--11 1111	--uu uuuu
ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	0000 0000	0000 0000	uuuu uuuu
PMDATL	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMADRL	xxxx xxxx	uuuu uuuu	uuuu uuuu
PMDATH	--xx xxxx	--uu uuuu	--uu uuuu
PMADRH	---- xxxx	---- uuuu	---- uuuu
PMCON1	1--- ---0	1--- ---0	1--- ---0

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

2: See Table 12-5 for RESET value for specific condition.

FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD})

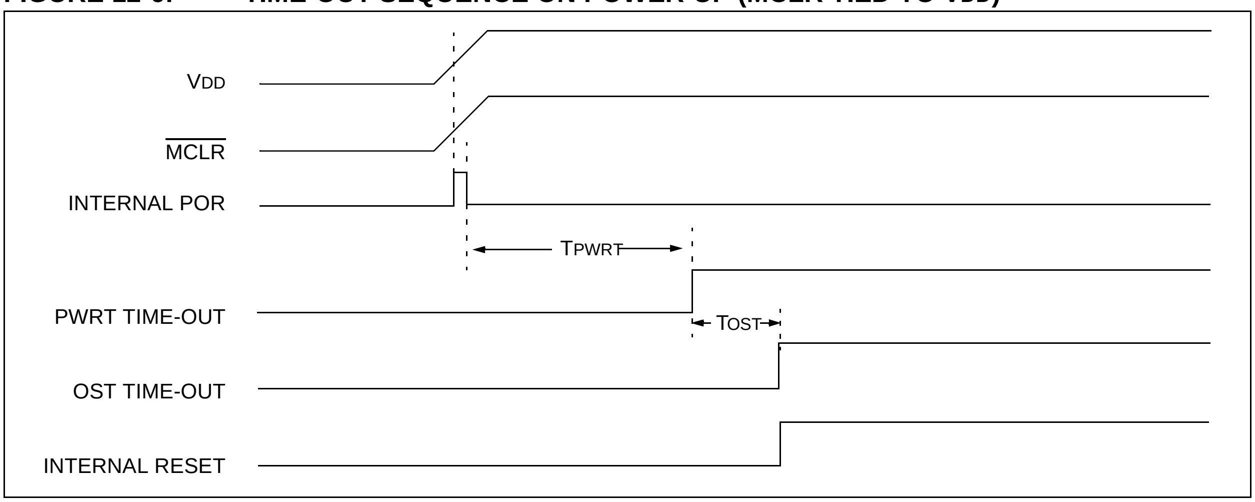
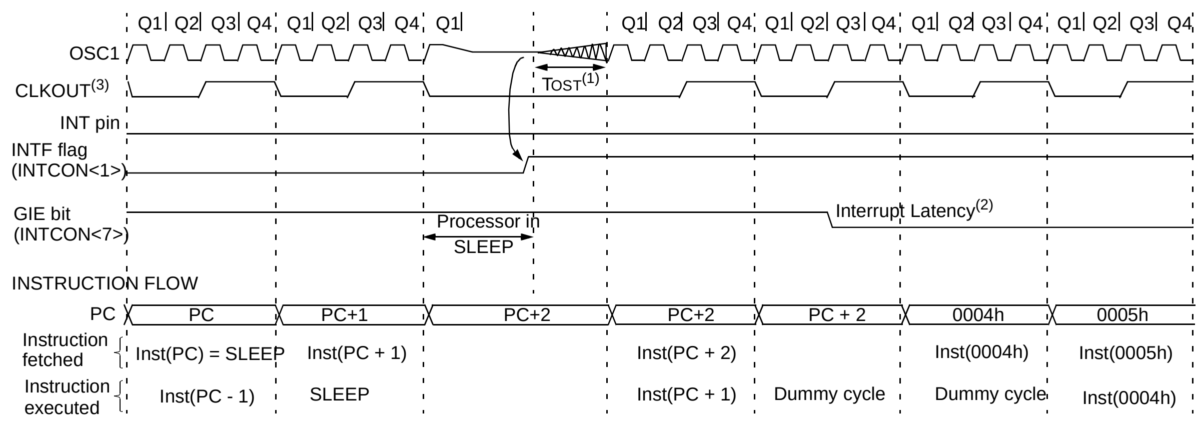


FIGURE 12-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: TOST = 1024TOSC (drawing not to scale) This delay applies to LP, XT and HS modes only.
2: GIE = '1' assumed. In this case after wake- up, the processor jumps to the interrupt routine. If GIE = '0', execution will continue in-line.
3: CLKOUT is not available in these osc modes, but shown here for timing reference.

12.14 Program Verification/Code Protection

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

Note:	Microchip does not recommend code protecting windowed devices. Code protected devices are not reprogrammable.

12.15 ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. It is recommended that only the 4 Least Significant bits of the ID location are used.

12.16 In-Circuit Serial Programming (ICSP™)

PIC16CXXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

For complete details of serial programming, please refer to the In-Circuit Serial Programming (ICSP™) Guide, (DS30277).

NOTES:

14.0 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
- 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

14.1 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.

14.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PIC MCUs.

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.

14.3 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.

14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

14.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

14.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows environment were chosen to best make these features available to you, the end user.

14.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

15.0 ELECTRICAL CHARACTERISTICS

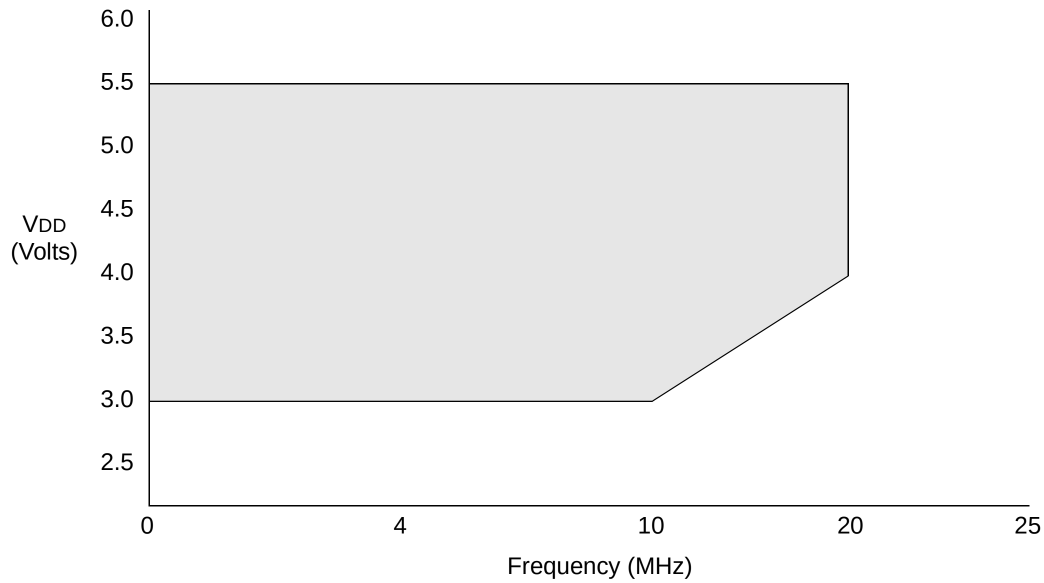
Absolute Maximum Ratings †

Ambient temperature under bias	-55 to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD}, $\overline{MCLR}$ and RA4)	-0.3V to (V_{DD} + 0.3V)
Voltage on V_{DD} with respect to V_{SS}	-0.3 to +7.5V
Maximum voltage between AV_{DD} and V_{DD} pins	± 0.3V
Maximum voltage between AV_{SS} and V_{SS} pins	± 0.3V
Voltage on $\overline{MCLR}$ with respect to V_{SS}	-0.3V to +8.5V
Voltage on RA4 with respect to Vss	-0.3V to +10.5V
Total power dissipation **(Note 1)**	1.0W
Maximum current out of V_{SS} pin	300 mA
Maximum current into V_{DD} pin	250 mA
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{DD}$)	± 20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{DD}$)	± 20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA and PORTB (combined)	200 mA
Maximum current sourced by PORTA and PORTB (combined)	200 mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$.

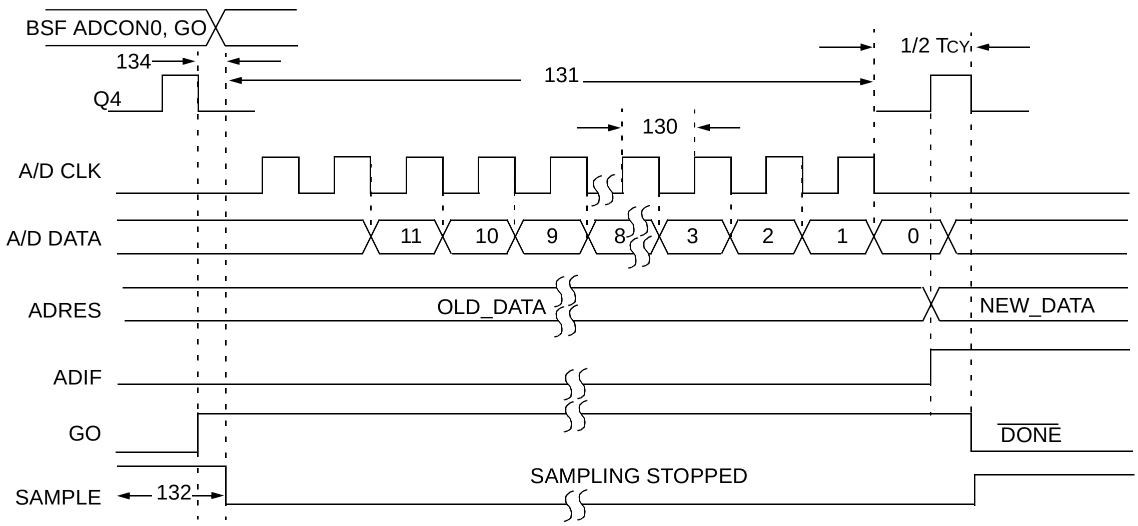
† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

FIGURE 15-3: PIC16LC717/770/771 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ 0°C, +70°C ≤ TA ≤ +85°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

FIGURE 15-14: PIC16C770/771 AND PIC16LC770/771 A/D CONVERSION TIMING (NORMAL MODE)



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

TABLE 15-22: MASTER SSP I²C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
101*	TLOW	Clock low time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
			1 MHz mode(1)	—	300	ns	
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
			1 MHz mode(1)	—	100	ns	
90*	TSU:STA	START condition setup time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	Only relevant for Repeated START condition
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
91*	THD:STA	START condition hold time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	After this period the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
			1 MHz mode(1)	TBD	—	ns	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**Note 2**
			400 kHz mode	100	—	ns	
			1 MHz mode(1)	TBD	—	ns	
92*	TSU:STO	STOP condition setup time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—	ms	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode(1)	—	—	ns	
110	TBUF	Bus free time	100 kHz mode	4.7 ‡	—	ms	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3 ‡	—	ms	
			1 MHz mode(1)	TBD‡	—	ms	
D102 ‡	Cb	Bus capacitive loading		—	400	pF	

* These parameters are characterized but not tested. For the value required by the I²C specification, please refer to the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

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

Note 1: Maximum pin capacitance = 10 pF for all I²C pins.

2: A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but (TSU:DAT) ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line. [(TR) + (TSU:DAT) = 1000 + 250 = 1250 ns], for 100 kHz mode, before the SCL line is released.

FIGURE 16-28: VOL VS. IOL (-40°C TO +125°C, VDD = 5.0V)

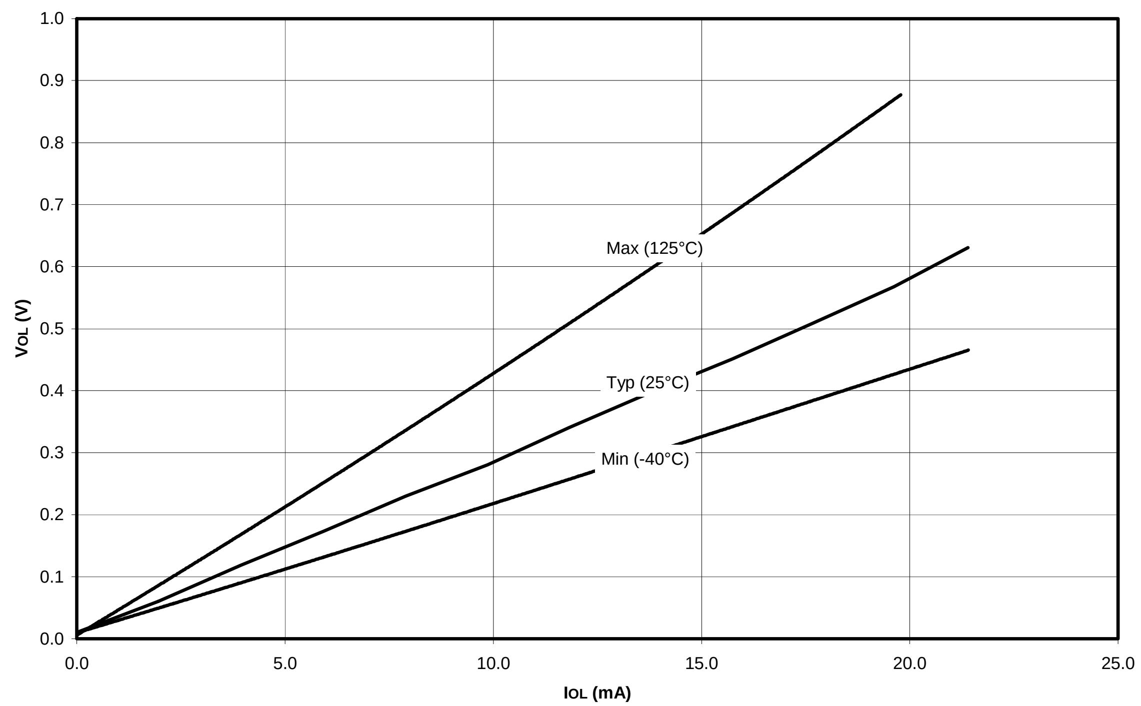
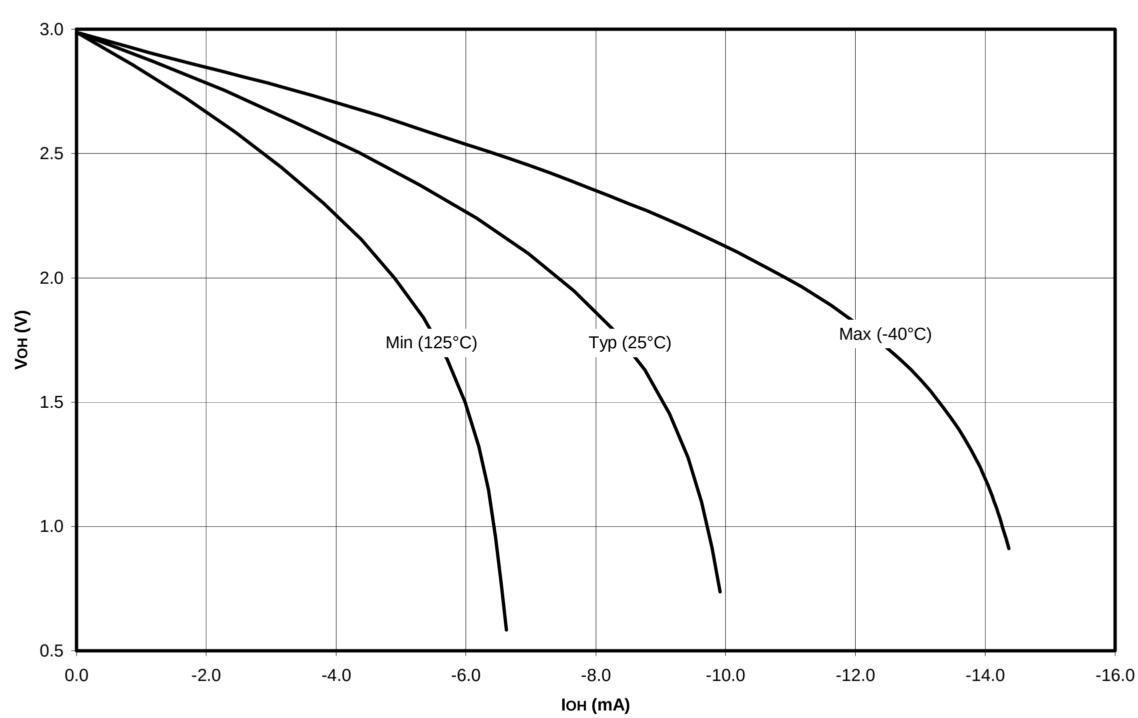


FIGURE 16-29: VOH VS. IOH (-40°C TO +125°C, VDD = 3.0V)



NOTES:

NOTES: