MTA85XXX

PICSEE™ 20-Pin MCU with Serial EEPROM Multi-Chip Module

FEATURES

- Multi-chip module
- PIC16C54A or PIC16C58A Microcontrollers with 24LC01B or 24LC02B Serial EEPROMs (SEEs) in a single package
- Wide operating voltage range: VDD = 3.0V to 6.25V
- Microcontroller control of SEE power for low standby current: MTA85X1X series
- Industrial grade **only**

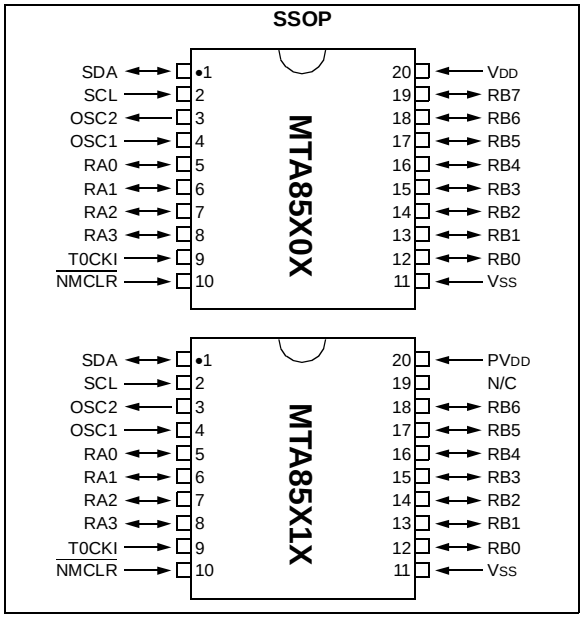
High Performance RISC-like CPU

- Only 33 single-word instructions to learn
- All instructions are single-cycle except for program branches, which are two cycle
- Operating speed: MTA854XX, DC - 4 MHz
  MTA858XX, DC - 4 and 10 MHz
- 12-bit wide instructions
- 8-bit wide data path
- 512 or 2048 x 12 on-chip EPROM program memory
- 25 or 73 x 8 general purpose registers (SRAM)
- Seven special function hardware registers
- Two-level deep hardware stack
- Direct, indirect, and relative addressing modes for data and instructions

Peripheral Features

- 12 I/O pins with individual direction control (RB7 dedicated for SEE VDD in MTA85X1X devices)
- 8-bit real time clock/counter (T0CKI) with 8-bit programmable prescaler
- Power-On Reset
- Oscillator Start-Up timer
- Watchdog timer (WDT) with its own on-chip RC oscillator for reliable operation
- Security EPROM bit for code-protection
- Power saving SLEEP mode
- EPROM selectable oscillator options:
  - Low-cost RC oscillator: RC
  - Standard crystal/resonator: XT
  - High-speed crystal/resonator: HS
  - Power-saving low frequency crystal: LP

PACKAGE TYPE



Serial EEPROM Features

- 1K or 2K of EEPROM memory, organized as a single block: 128 x 8 or 256 x 8
- Two-wire serial interface bus, I^2C™ compatible
- 100 kHz and 400 kHz compatibility
- Self-timed write cycle (including auto-erase)
- Page-write buffer for up to 8 bytes
- 2 ms typical cycle times for page-write
- 1,000,000 ERASE/WRITE cycles typical
- Data retention > 40 years

CMOS Technology

- Low-power, high-speed CMOS EPROM and EEPROM technologies, in a single package
- Fully static design
- Low-power consumption (PIC16C54/58A)
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 3V, 32 kHz
  - < 0.3 µA typical standby current (with WDT disabled) @ 3V, 0°C to 70°C
- Low-power consumption (24LC01B/02B)
  - 1 mA active current typical
  - 10 µA standby current typical @ 5.5V
  - 5 µA standby current typical @ 3.0V

PIN DESCRIPTIONS

Name	Function	Description
RA3: RA0	I/O PORTA	4 input/output lines.
RB7: RB0	I/O PORTB	8 input/output lines.
SVDD/RB7	Shared VDD-I/O pin	Input/Output pin dedicated to EEPROM VDD. No external connection needed. MTA85X1X only.
T0CKI*	Clock input to TMR0 Register	Schmitt Trigger Input.
		Clock input to T0CKI register. Must be tied to VSS or VDD if not in use to avoid unintended entering of test modes and to reduce current consumption.
$\overline{\text{MCLR}}$	Master Clear	Schmitt Trigger Input.
		A low voltage on this input generates a RESET for the microcontroller.
		A rising voltage triggers the on-chip oscillator start-up timer which keeps the chip in RESET mode for about 18 ms. This input must be tied directly, or via a pull-up resistor, to VDD.
OSC1	Oscillator (input)	**XT, HS and LP devices:** Input terminal for crystal, ceramic resonator, or external clock generator.
		RC devices: Driver terminal for external RC combination to establish oscillation.
OSC2/CLKOUT	Oscillator (output)	**For XT, HS and LP devices:** Output terminal for crystal and ceramic resonator. Do not connect any other load to this output. Leave open if external clock generator is used.
		For RC devices: A CLKOUT signal with a frequency of 1/4 FOSC1 is put out on this pin.
SDA	Serial EEPROM Data	EEPROM data line.
SCL	Serial EEPROM Clock	EEPROM clock line.
VDD	Power supply	
VSS	Ground	

* Formerly RTCC.

Table of Contents

1.0 GENERAL DESCRIPTION

The MTA85XXX devices from Microchip Technology Inc. are a family of multi-chip products which offer a unique combination of EPROM-based Microcontrollers and Serial EEPROM data memory in a single package. The MTA85XXX line features the PIC16C5XA family of Microcontrollers combined with Microchip's 24LC0XB family of Serial EEPROMs.

The Microcontroller and Serial EEPROM portions of these multi-chip devices are equivalent to their respective individual components chips, except for the electrical specifications on shared pins. Please refer to the datasheets of the component die for information on each device's architecture, functionality, and other important user information.

Two unique pinouts are available in this family of devices, regardless of which combination of component chips are used. The first pinout (MTA85X0X series) features shared power and ground pins for the Microcontroller and Serial EEPROM. All other Microcontroller and Serial EEPROM pins are electrically independent. The second available pinout (MTA85X1X series) features Microcontroller control of the Serial EEPROM V_{DD}. This allows the Serial EEPROM to be powered down when going into a standby mode. This is often desirable in power conscious applications to reduce current when the Serial EEPROM is not being accessed. In this configuration the Microcontroller I/O pin RB7 is used to supply power to the Serial EEPROM. It is the user's responsibility to ensure that RB7 is driving a '1' while the Serial EEPROM is being used.

The MTA85XXX devices are supported by an in-circuit emulator, an assembler, and a production quality programmer. All tools are supported by IBM PC® and compatible machines.

1.1 Applications

The MTA85XXX family is ideally suited to a wide variety of applications including, but not limited to: keyless entry, remote control, smart cards and automotive controllers. The EPROM program memory makes customization of application programs fast and convenient. The EEPROM data memory is ideal for storing configuration information, access codes, serial numbers, and adaptive look-up tables. The small footprint package makes the MTA85XXX devices perfect for applications with physical space limitations. This small size coupled with the low-cost, low-power, wide voltage range, and high performance of this flexible family of devices makes the MTA85XXX the microcontroller of choice for a wide variety of applications which utilize EEPROM memory.

1.2 MTA85XXX Series Overview

A variety of EPROM program memory sizes, EEPROM data memory sizes and frequency ranges are available. Depending on the application and production requirements, the proper device option can be selected using the information in Table 1-1 and Table 1-2. When placing orders, please use the "MTA85XXX Product Identification System" on the back page of this data sheet to specify the correct part.

TABLE 1-1: FAMILY OVERVIEW

Part Number	Microcontroller	SEE	PGM EPROM	EEPROM	RAM	I/O
MTA85401	PIC16C54A	24LC01B	512 x 12	128 x 8	32 x 8	12
MTA85402	PIC16C54A	24LC02B	512 x 12	256 x 8	32 x 8	12
MTA85411	PIC16C54A	24LC01B	512 x 12	128 x 8	32 x 8	12 note 1
MTA85412	PIC16C54A	24LC02B	512 x 12	256 x 8	32 x 8	12 note 1
MTA85801	PIC16C58A	24LC01B	2048 x 12	128 x 8	80 x 8	12
MTA85802	PIC16C58A	24LC02B	2048 x 12	256 x 8	80 x 8	12
MTA85811	PIC16C58A	24LC01B	2048 x 12	128 x 8	80 x 8	12 note 1
MTA85812	PIC16C58A	24LC02B	2048 x 12	256 x 8	80 x 8	12 note 1

Note: RB7 dedicated to SEE V_{DD}

2.0 ARCHITECTURAL DESCRIPTION

2.1 Harvard Architecture

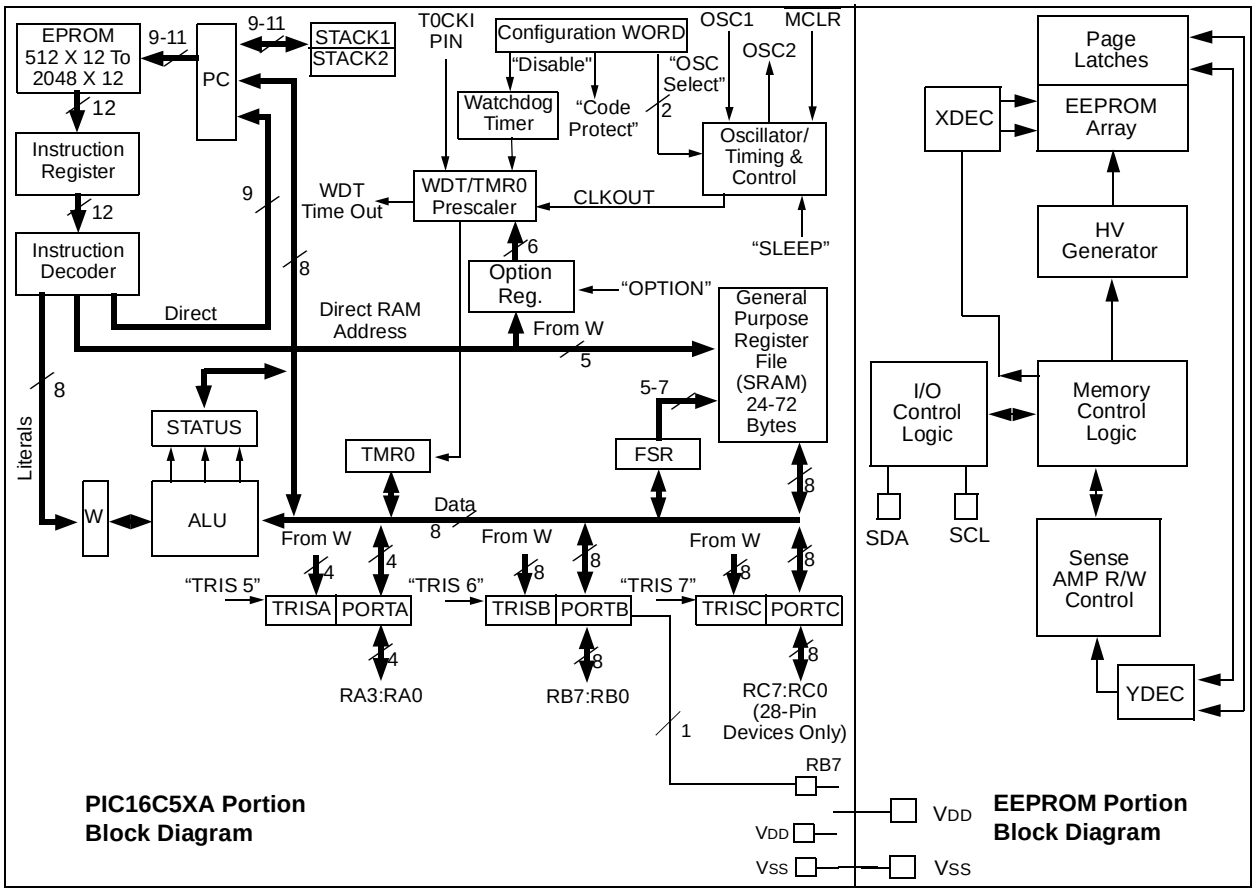
The MTA85XXX microcontrollers are low-power, high-speed, full static CMOS devices containing EEPROM, EPROM, RAM, I/O and a central processing unit in a single package.

The architecture is based on a register file concept with separate bus and memories for data and instructions (Harvard architecture). The data bus and memory (RAM) are 8-bits wide while the program bus and program memory (EPROM) have a width of 12-bits. This concept allows a simple yet powerful instruction set designed to emphasize bit, byte and register operations under high speed with overlapping instruction fetch and execution cycles. That means that, while one instruction is executed, the following instruction is already being read from the program memory. A block diagram of the MTA85XXX is given in Figure 2-1.

2.2 Clocking Scheme/Instruction Cycle

The clock input (from pin OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, PC is incremented every Q1, instruction is fetched from program memory and latched into instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 2-2.

FIGURE 2-1: MTA85XXX SERIES BLOCK DIAGRAM



MTA85X0X devices have PIC16C5XA V_{DD} tied to EEPROM V_{DD}.

MTA85X1X devices have PIC16C5XA RB7 tied to EEPROM V_{DD}.

2.3 Data Register File

The 8-bit data bus connects two basic functional elements together: the Register File composed of addressable 8-bit registers including the I/O Ports, and an 8-bit wide Arithmetic Logic Unit. The 32 bytes of RAM are directly addressable while a "banking" scheme, with banks of 16 bytes each, is employed to address larger data memories (Figure 5-1). Data can be addressed direct, or indirect using the file select register (f4). Immediate data addressing is supported by special "literal" instructions which load data from program memory into the W register.

The register file is divided into two functional groups: operational registers and general purpose registers. The operational registers include the Real Time Clock Counter (T0CKI) register, the Program Counter (PC), the Status Register, the I/O registers (PORTs), and the File Select Register. The general purpose registers are used for data and control information under command of the instructions.

In addition, special purpose registers are used to control the I/O port configuration, and the prescaler options.

2.4 Arithmetic/Logic Unit (ALU)

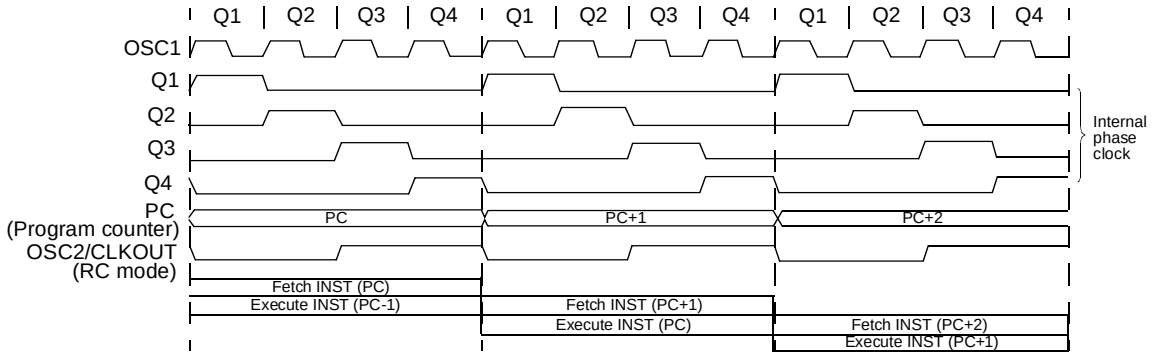
The 8-bit wide ALU contains one temporary working register (W Register). It performs arithmetic and Boolean functions between data held in the W Register and any file register. It also does single operand operations on either the W register or any file register.

2.5 Program Memory

512 or 2048 words of 12-bit wide on-chip program memory (EPROM) can be directly addressed.

Sequencing of microinstructions is controlled via the Program Counter (PC) which automatically increments to execute in-line programs. Program control operations, supporting direct, indirect, relative addressing modes, can be performed by Bit Test and Skip instructions, Call instructions, Jump instructions or by loading computed addresses into the PC. In addition, an on-chip two-level stack is employed to provide easy to use subroutine nesting.

FIGURE 2-2: CLOCKS/INSTRUCTION CYCLE



Note: For additional details on the PIC16C5X architecture, please refer to document # DS30236.

3.0 FACTORY PROGRAMMING OPTIONS

A variety of EPROM program memory sizes, EEPROM data memory sizes and frequency ranges are available. Depending on the application and production requirements, the proper device option can be selected using the information in Table 1-1 and Table 1-2. When placing orders, please use the “MTA85XXX Product Identification System” on the back page of this data sheet to specify the correct part.

3.1 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

With OTP devices the program EPROM is erased, allowing the user to write the application code into it. Additionally the watchdog timer can be disabled, and/or the code protection logic can be activated by programming special EPROM fuses. 16 non-dedicated EPROM bits are available for the customer ID or other customer information and are also user programmable.

3.2 Quick-Turn-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for volume users with stable code, who choose not to program the devices themselves. A QTP device is identical to an OTP device, except that the program memory and special EPROM fuses are programmed at the factory, with the customer’s code. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your Microchip Technology Inc. sales office for more details.

3.3 Serialized-Quick-Turnaround-Production (SQTP) Devices

Microchip offers the unique programming service where few locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

4.0 PROGRAM MEMORY

The MTA854XX devices contain 512 12-bit words of program memory. The MTA858XX devices contain 2048 12-bit words. Refer to Figure 4-1 and Figure 4-2 for a description of the program memory organization.

4.1 Program Memory Organization

Up to 512 words of 12-bit wide on-chip program memory (EPROM/ROM) can be directly addressed. Larger program memories can be addressed by selecting one of up to four available pages of 512 words each (Figure 4-2). Sequencing of instructions is controlled via the Program Counter (PC) which automatically increments to execute in-line programs. Program control operations supporting direct, indirect, and relative addressing modes, can be performed by bit test, skip, call, and jump type instructions, or by loading computed addresses into the PC. In addition, an on-chip two-level stack is employed to provide easy to use subroutine nesting.

4.2 Program Counter

The program counter generates addresses for on-chip EEPROM containing the program instruction words.

The program counter is set to all '1's upon a RESET condition. During program execution, it is auto incremented with each instruction unless the result of that instruction changes the PC itself:

a) GOTO instructions allow the direct loading of the lower nine program counter bits (PC8:PC0). For MTA858XX devices, the upper two bits of PC (PC10:PC9) are loaded with page select bits PA1:PA0 (STATUS <6:5>). Thus GOTO permits jumping to any location on any page.

b) CALL instructions load the lower 8-bits of the PC directly while the 9-bits are cleared. The PC value, incremented by one, will be PUSH'ed onto the stack. For MTA858XX, the upper two bits of PC (PC10:PC9) are loaded with Page Select bits PA1:PA0 (STATUS <6:5>).

c) RETLW instructions load the program counter with the top of stack contents.

d) If the PC is the destination in any instruction (e.g., MOVWF PC, ADDWF PC, or BSF PC, 5), then the computed 8-bit result will be loaded into the low 8-bits of program counter. The ninth bit of PC will be cleared. In MTA858XX devices PC10:PC9 will be loaded with the page select bits.

The MTA858XX devices have multiple program memory pages. It should be noted for the MTA858XX products that because bit 8 (ninth bit) of PC is cleared in CALL instruction or any instruction which writes to the PC (e.g., MOVWF PC), all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

Incrementing the program counter when it is pointing to the last address of a selected memory page is also possible and will cause the program to continue in the next page. However, the page pre-select bits in the STATUS register will not be changed and the next GOTO, CALL, ADDWF PC, MOVWF PC instruction will return to the previous page unless the page pre-select bits have been updated under program control. For example, a NOP at location 1FFh (page 0) increments the PC to 200h (page 1). A "GOTO xxx" at 200h will return the program to address "xxx" on page 0 (assuming that the page preselect bits in the STATUS register are '0').

Upon a RESET condition, page 0 is pre-selected while the program counter addresses the last location in the last page. Thus, a GOTO instruction at this location will automatically cause the program to continue in page 0.

Note:	The MTA854XX devices only have a single page, page 0 (Figure 4-1).

4.3 Stack

The MTA85XXX employs a two-level hardware PUSH/POP stack (Figure 4-1 and Figure 4-2).

The CALL instruction pushes the current program counter value, incremented by one, into stack level 1. Stack level 1 is automatically pushed to level 2. If more than 2 subsequent CALLs are executed, only the most recent two return addresses are stored.

The page preselect bits of the STATUS register will be loaded into the most significant bits of the program counter. The ninth bit is always cleared upon a CALL instruction. This means that subroutine entry addresses have to be located always within the lower half of a memory page (addresses 000h-0FFh, 200h-2FFh, 400h-4FFh, 600h-6FFh). However, as the stack has the same width as the PC, subroutines can be called from anywhere in the program.

The RETLW instruction loads the contents of the stack level 1 into the program counter while stack level 2 gets copied into level 1. If more than 2 subsequent RETLWs are executed, the stack will be filled with the address previously stored in level 2. The return will be always to the page from where the subroutine was called, regardless of the current setting of the page pre-select bits in the STATUS register. Note that the W register will be loaded with the literal value specified in the RETLW instruction. This is particularly useful for the implementation of “data” tables within the program memory.

FIGURE 4-1: PROGRAM MEMORY ORGANIZATION MTA854XX

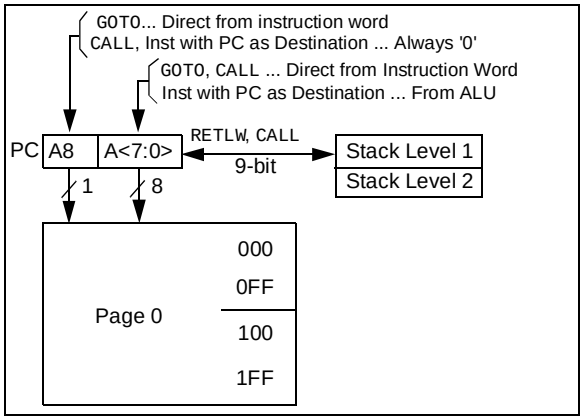
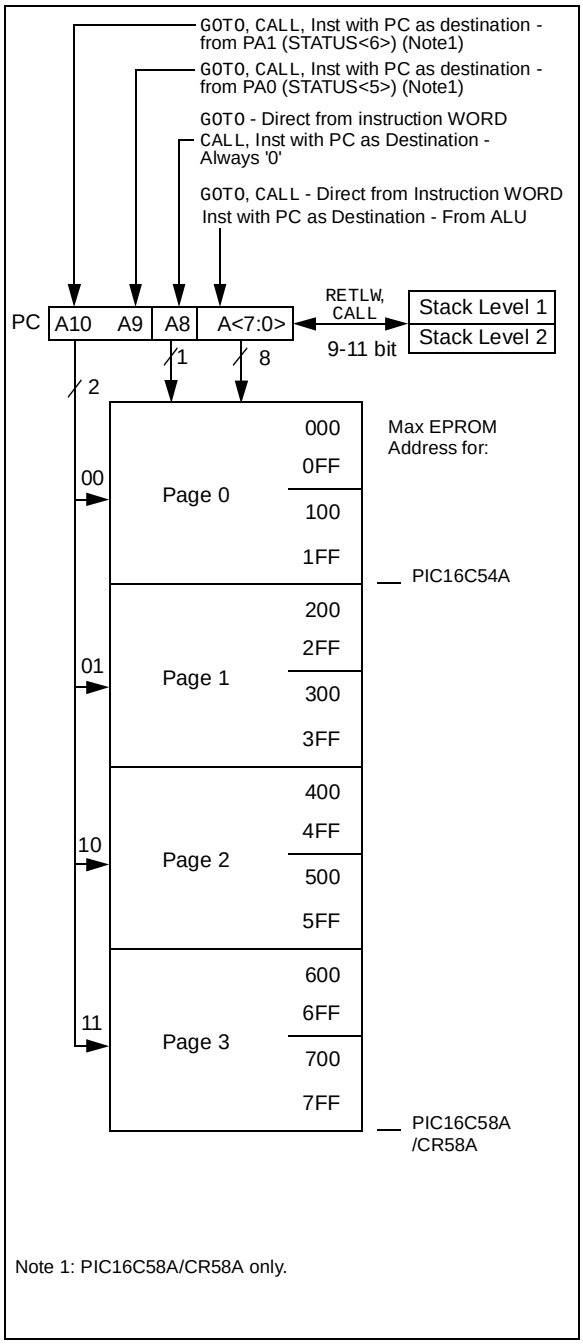


FIGURE 4-2: PROGRAM MEMORY ORGANIZATION MTA858XX

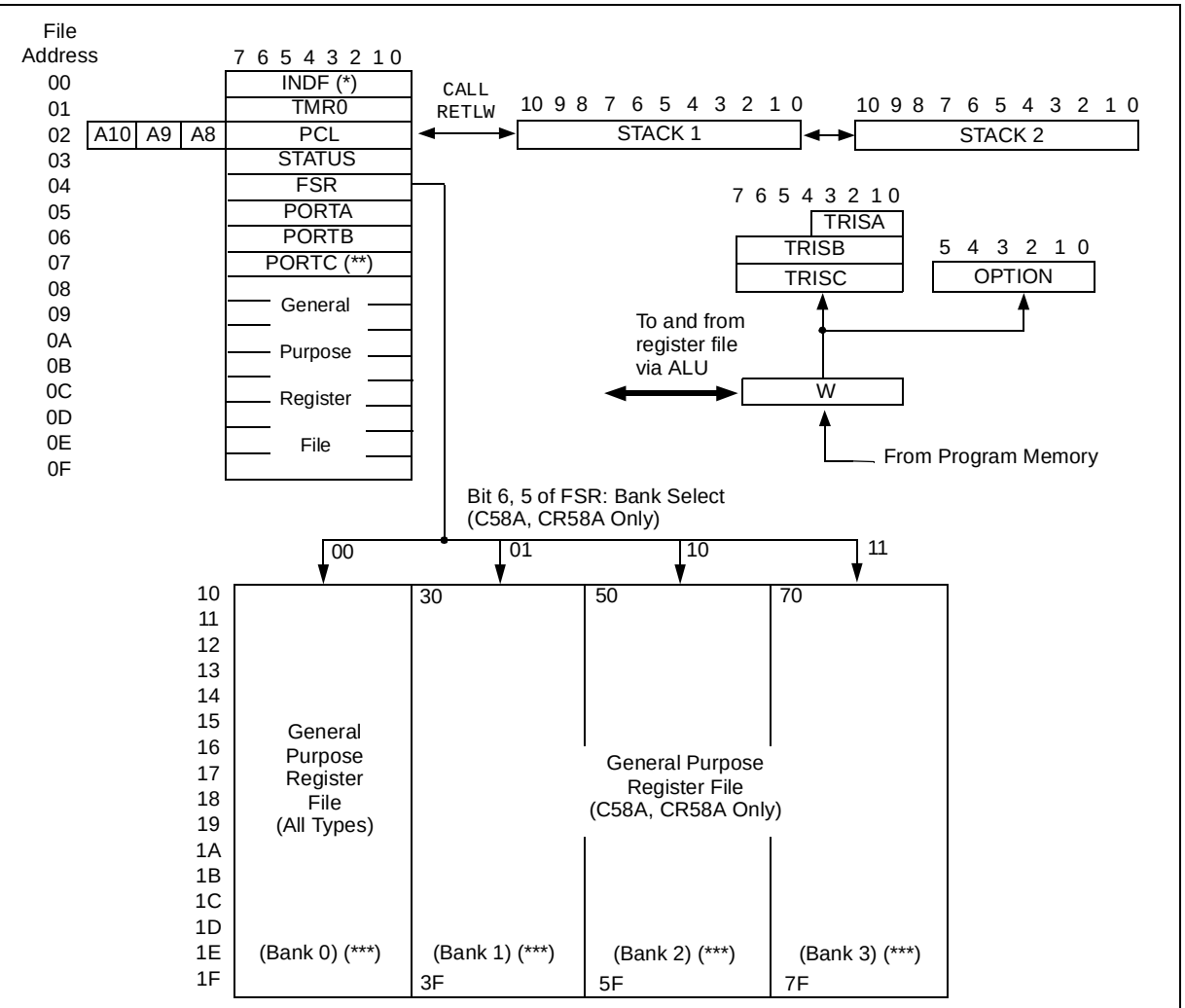


5.0 DATA MEMORY AND OPERATIONAL REGISTER FILES

For MTA854XX devices, there are seven special function registers (operational register files) and 25 general purpose registers. Data addresses 00h-06h are reserved for the operational register files, addresses 07h-1Fh are used for the general purpose file registers. FSR bits (STATUS <6:5>) are not used.

For MTA858XX devices, there are seven special function register (operational register files) and 73 general purpose registers. These registers are mapped according to Figure 5-1. Note that several address blocks are mapped to the same physical registers. To address the banked registers (addresses above 0Fh) bits (STATUS <6:5>) of the FSR are used.

FIGURE 5-1: DATA MEMORY MAP



(*) Not a physically implemented register. See Section 4.5 for details.

(**) File address 7h is a general purpose register on the PIC16C54A, and PIC16C58A/CR58A.

(***) Bank 0 is available on all microcontrollers while Bank 1, Bank 2, and Bank 3 are only available on the C58A, CR58A. (Section 4.5).

5.1 Data Memory Organization

The 8-bit data bus connects two basic functional elements together: the register file composed of up to 80 addressable 8-bit registers including the I/O ports, and an 8-bit wide Arithmetic Logic Unit (ALU). 32 bytes of RAM are directly addressable while a “banking” scheme, with banks of 16 bytes each, is employed to address larger data memories (Figure 5-1). Data can be addressed directly, or indirectly using the File Select Register (FSR). Immediate data addressing is supported by special “literal” instructions which load data from program memory into the W register.

The register file is divided into two functional groups: Special Function registers and General Purpose registers. The special function registers include the Timer0 (TMR0) register, the Program Counter (PC), the Status Register, the I/O registers (ports), and the File Select Register (FSR). The general purpose registers are used for data and control information under command of the instructions.

In addition, special purpose registers are used to control the I/O port configuration and prescaler options.

5.1.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly or indirectly through the file select register FSR.

5.1.2 SPECIAL FUNCTION REGISTERS:

The Special Function Registers are registers used by the CPU and peripheral functions to control the operation of the device (Table 5-1).

The special registers can be classified into two sets. The special registers associated with the “core” functions are described in this section. Those related to the operation of the peripheral features are described in the section for each peripheral feature.

TABLE 5-1: PIC16C5X REGISTER FILE SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on $\overline{MCLR}$ and WDT resets
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of PC								1111 1111	1111 1111
03h	STATUS	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
07h	PORTC [2]	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged. - = unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. The upper bits can be set or cleared by writing to PA1:PA0 (STATUS<6:5>).

2: File address 7h is a general purpose register on the PIC16C54A and PIC16C58A/CR58A.

3: Shading indicates unimplemented bits.

5.2 Indirect Data Addressing (INDF) f0

This is not a physically implemented register. Addressing INDF calls for the contents of the File Select Register to be used to select a file register. The INDF register is useful as an indirect address pointer. For example, in the instruction `ADDWF INDF, W` will add the contents of the register pointed to by the FSR to the content of the W Register and place the result in W.

If INDF itself is read through indirect addressing (i.e., FSR = 0h), then 00h is read. If the INDF register is written to via indirect addressing, the result will be a no operation (NOP).

5.3 Real Time Clock/Counter Register (T0CKI) f1

This register can be loaded and read by the program as any other register. In addition, its contents can be incremented by an external signal edge applied to the T0CKI pin or by the internal instruction cycle clock (CLKOUT = FOSC/4). Figure 5-2 is a simplified block diagram of the T0CKI module.

An 8-bit prescaler can be assigned to the T0CKI by writing the proper values to the PSA bit and the PS bits in the OPTION register. The OPTION register is a special register (not mapped in data memory) addressable using the `OPTION` instruction (Section 6.4). If the prescaler is assigned to the T0CKI, instructions writing to the T0CKI register (e.g., `CLRF T0CKI`, or `BSF T0CKI,5`,...etc.) clear the prescaler.

The bit RTS (T0CKI Signal Source) in the OPTION register determines if the T0CKI register is incremented internally or externally.

RTS = 1: The clock source for the T0CKI or the prescaler, if assigned to it, is the signal on the T0CKI pin. Bit4 of the OPTION register (RTE) determines if an increment occurs on the falling (RTE = 1) or rising (RTE = 0) edge of the signal presented to the T0CKI pin.

RTS = 0: The T0CKI register or its prescaler, respectively, will be incremented with the internal instruction clock (= FOSC/4). The RTE bit in the OPTION register and the T0CKI pin are "don't care" in this case. However, the T0CKI pin must not be left floating (tie to VDD or VSS). This prevents unintended operation and to reduce the current consumption in low-power applications.

As long as clocks are applied to the T0CKI (from internal or external source, with or without prescaler), the T0CKI register keeps incrementing and just rolls over when the value FFh is reached. All increment pulses for the T0CKI register are delayed by two instruction cycles. After writing to the T0CKI register, for example, no increment takes place for the following two instruction cycles. This is independent if internal or external clock source is selected. If a prescaler is assigned to the T0CKI, the output of the prescaler will be delayed by two cycles before the T0CKI register is incremented. This is true for instructions that either write to or read-modify-write T0CKI (e.g., `MOVF T0CKI, CLRF T0CKI`). For applications where T0CKI needs to be tested for '0' without affecting its count, use of the `MOVF T0CKI, W` instruction is recommended. Timing diagrams in Figure 5-3 and Figure 5-4 show T0CKI read, write and increment timing.

5.3.1 USING T0CKI WITH EXTERNAL CLOCK

When external clock input is used for T0CKI, it is synchronized with internal phase clocks. Therefore, external clock input must meet certain requirements. Also there is some delay from the occurrence of the external clock edge to the actual incrementing of T0CKI. Referring to Figure 5-5, the synchronization is done after the prescaler. Output of the prescaler is sampled twice in every instruction cycle to detect rising or falling edges. Therefore, it is necessary for PSOUT to be high for at least 2 TOST and low for at least 2 TOSC where:

TOSC = oscillator time period.

When no prescaler is used, PSOUT (Prescaler output, Figure 5-3) is the same as T0CKI clock input and, therefore, the requirements are:

TRTH = T0CKI high time ≥ 2 TOSC + 20 ns

TRTL = T0CKI low time ≥ 2 TOSC + 20 ns

When prescaler is used, the T0CKI input is divided by the asynchronous ripple counter-type prescaler so the prescaler output is symmetrical.

Then:

$$\text{PSOUT high time} = \text{PSOUT low time} = \frac{N \bullet T_{RT}}{2}$$

where TRT = T0CKI input period and N = prescale value (2, 4, ..., 256).
The requirement is, therefore, $\frac{N \bullet T_{RT}}{2}$

$\geq$ 2 TOSC + 20 ns, or $T_{RT} \geq \frac{4\,T_{OSC} + 40\text{ ns}}{N}$

The user will notice that no requirement on T0CKI high time or low time is specified. However, if the high time or low time on T0CKI is too small, then the pulse may not be detected. Hence, a minimum high or low time of 10 ns is required. In summary, the T0CKI input requirements are:

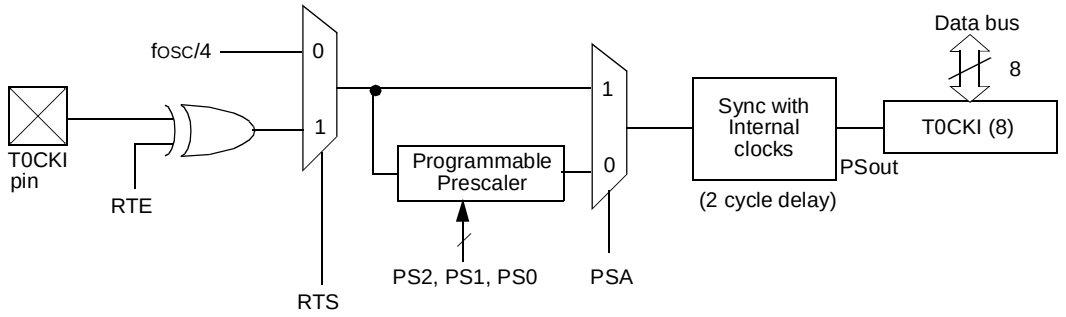
TRT = T0CKI period ≥ (4 TOSC + 40 ns)/N

TRTH = T0CKI high time ≥ 10 ns

TRTL = T0CKI low time ≥ 10 ns

Delay from external clock edge: Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the T0CKI is actually incremented. This delay is between 3 TOSC and 7 TOSC (Figure 5-5). Thus, for example, measuring the interval between two edges (e.g., period) will be accurate within ±4 TOSC (±200 ns @ 20 MHz).

FIGURE 5-2: T0CKI BLOCK DIAGRAM (SIMPLIFIED)



Note 1: Bits, RTE, RTS, PS2, PS1, PS0 and PSA are located in the OPTION register.
Note 2: The prescaler is shared with Watchdog Timer.

FIGURE 5-3: T0CKI TIMING: INT CLOCK/NO PRESCALE

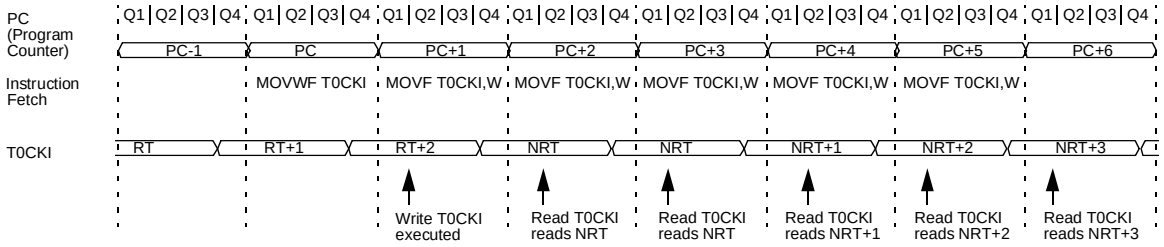


FIGURE 5-4: T0CKI TIMING: INT CLOCK/PRESCALE 1:2

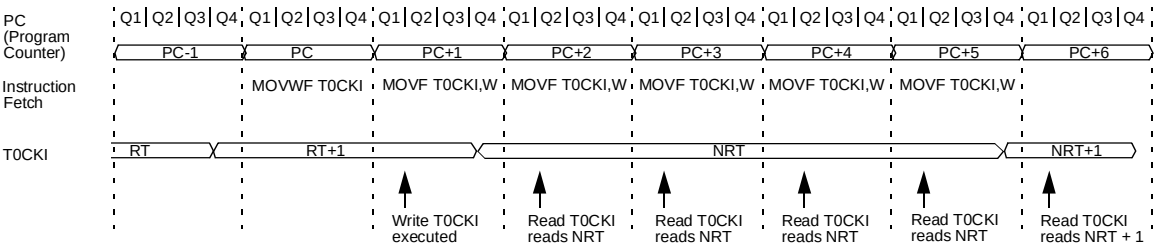
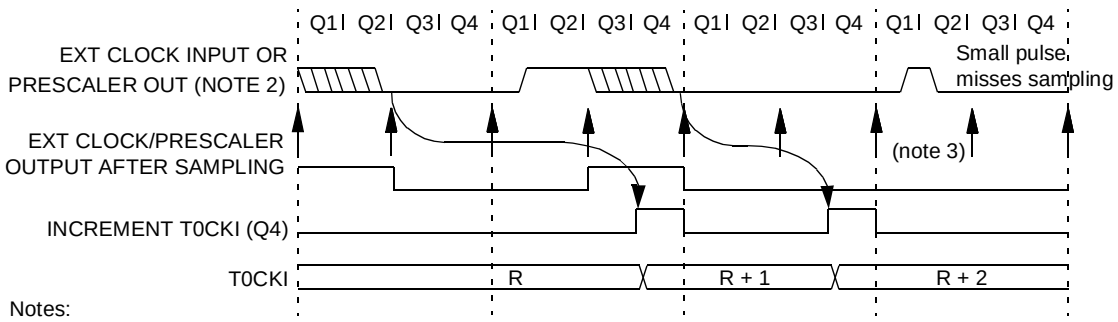


FIGURE 5-5: T0CKI TIMING WITH EXTERNAL CLOCK



Notes:

1. Delay from clock input change to T0CKI increment is 3 TOSC to 7 TOSC. (Duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on T0CKI input = ± 4 TOSC maximum.
2. External clock if no prescaler selected, Prescaler output otherwise.
3. The arrows indicate the points in time where sampling occurs.

5.4 Program Counter (PC) f2

See Section 4.2

5.5 Stack

See Section 4.3

5.6 Status Word Register (STATUS) f3

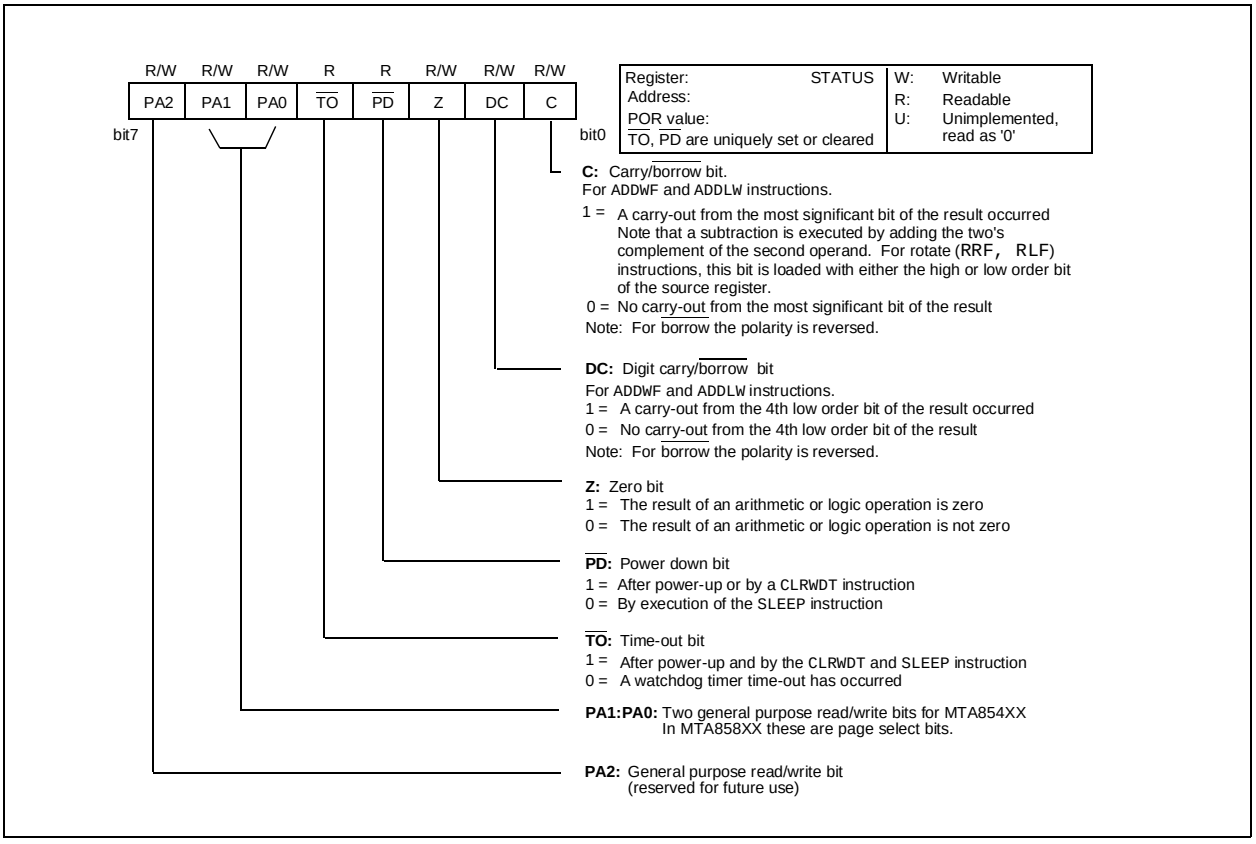
This register contains the arithmetic status of the ALU, the RESET status.

The STATUS register can be destination for any instruction like any other register. However, the status bits are set after the following write. Furthermore, $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with STATUS register as destination may be different than intended. For example, `CLRF STATUS` will clear all bits except for $\overline{TO}$ and $\overline{PD}$ and then set the Z bit and leave STATUS register as `000uu100` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF` and `MOVWF` instructions are used to alter the STATUS registers because these instructions do not affect any status bit.

For other instructions, affecting any status bits, see Section "Instruction Set Summary" (Table 9-1).

FIGURE 5-6: STATUS WORD REGISTER



5.6.1 CARRY/BORROW AND DIGIT CARRY/BORROW BITS:

The carry bit (C) is a carry out in addition operation (ADDWF) and a borrow out in subtract operation (SUBWF).

It is also affected by RRF and RLF instructions. The following examples explain carry/borrow bit operation:

```
;SUBWF Example #1
;
clrf    0x1f    ;f(1fh)=0
movlw   1       ;wreg=1
subwf   0x1f    ;f(1fh) =f(1fh)- wreg
                ;=0-1=FFh
                ;Carry=0: Result is
                ;negative
;
;SUBWF Example #2
movlw   0xFF    ;
movwf   0x1F    ;f(0x1F)=FFh
clrw            ;wreg=0
subwf   0x1F    ;f(0x1F)=f(0x1F)- wreg
                ;=FFh-0=FFh
                ;Carry=1:Result is
                ;positive
;
```

The digit carry operates in the same way as the carry bit, (i.e., it is a borrow in a subtract operation).

5.6.2 TIME OUT AND POWER DOWN STATUS BITS ($\overline{TO}$, $\overline{PD}$)

The $\overline{TO}$ and $\overline{PD}$ bits (STATUS<4:3>) can be tested to determine if a RESET condition has been caused by a Watchdog Timer time-out, a power-up condition, or a wake-up from SLEEP by the Watchdog Timer or $\overline{MCLR}$ pin.

These status bits are only affected by events listed in Table 5-2.

TABLE 5-2: EVENTS AFFECTING $\overline{TO}$/$\overline{PD}$ STATUS BITS

Event	$\overline{TO}$	$\overline{PD}$	Remarks
Power-Up	1	1	No effect on $\overline{PD}$
WDT Time-out	0	x	
SLEEP instruction	1	0	
CLRWDT instruction	1	1	

Note 1: A WDT time-out will occur regardless of the status of the bit $\overline{TO}$. A SLEEP instruction will be executed, regardless of the status of the $\overline{PD}$ bit. Table 5-3 reflects the status of $\overline{TO}$ and $\overline{PD}$ after the corresponding event.

TABLE 5-3: $\overline{TO}$/$\overline{PD}$ STATUS AFTER RESET

$\overline{TO}$	$\overline{PD}$	Reset was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
1	0	$\overline{MCLR}$ wake-up from SLEEP
1	1	Power-up
u	u	Low pulse on $\overline{MCLR}$ input

Note 1: The $\overline{TO}$ and $\overline{PD}$ bit maintain their status (u) until an event of Table 5-2 occurs. A low pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$ and $\overline{PD}$ status bits.

5.6.3 PROGRAM PAGE PRESELECT FOR MTA858XX

Bits (STATUS <6:5>) are defined as page address bits PA1:PA0 and are used to preselect a program memory page. When executing a GOTO, CALL, or an instruction with PC as destination (e.g., MOVWF PC), PA1:PA0 are loaded into bit A10:A9 of the program counter, selecting one of the available program memory pages. The direct address specified in the instruction is only valid within this particular memory page.

RETLW instructions do not change the page preselect bits.

Upon a RESET condition, PA2:PA0 are cleared to '0's.

5.7 File Select Register (FSR) f4

Bits (STATUS <4:0>) select one of the 32 available file registers in the indirect addressing mode (that is, calling for the INDF register in any of the file oriented instructions).

Bit7 of the FSR is read-only and is always read as a '1'.

If no indirect addressing is used, the FSR can be used as a 5-bit wide general purpose register.

Bits (STATUS <6:5>) of the FSR will always read as a '1' for MTA854XX devices. For MTA858XX products, bits (STATUS <6:5>) of the FSR will select the current data memory bank (Figure 5-1).

The lower 16 bytes of each bank are physically identical and are always selected when bit4 of the FSR (in case of indirect addressing) is '0', or bit4 of the direct file register address of the currently executing instruction is '0' (e.g., MOVWF DATAMEM).

Only if bit4 in the above mentioned cases is '1', (STATUS <6:5>) of the FSR select one of the four available register banks with 16 bytes each (MTA858XX only).

5.8 I/O Registers (Ports)

The I/O registers can be written and read under program control like any other register of the register file. However, "read" instructions (e.g., `MOVF PORTB,W`) always read the I/O pins, regardless if a pin is defined as "input" or "output." Upon a RESET, all I/O ports are defined as "input" (= hi impedance mode) as the I/O control registers (TRISA, TRISB) are all set to '1's.

The execution of a "TRIS f" instruction with corresponding '0's in the W register is necessary to define any of the I/O pins as output.

5.8.1 PORTA f5

4-bit I/O register. Low-order 4-bits only are used (RA3: RA0). Bits (STATUS <7:4>) are unimplemented and read as '0's.

5.8.2 PORTB f6

8-bit I/O register. Note that RB7 is tied to SEE V_{DD} on MTA85XIX devices.

5.8.3 I/O INTERFACING

The equivalent circuit for an I/O port bit is shown in Figure 5-7. All ports may be used for both input and output operations. For input operations these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF PORTB, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit (in TRISA, TRISB) must be set to '0'. For use as an input, the corresponding TRIS bit must be '1'. Any I/O pin can be programmed individually as an input or output.

FIGURE 5-7: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN

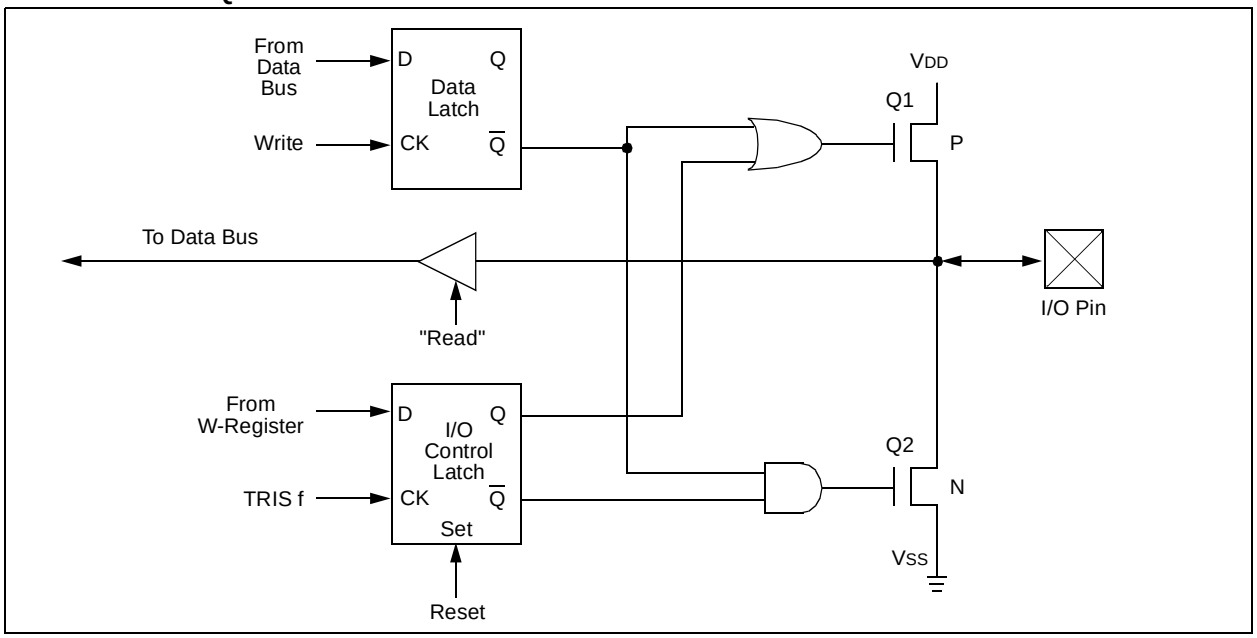
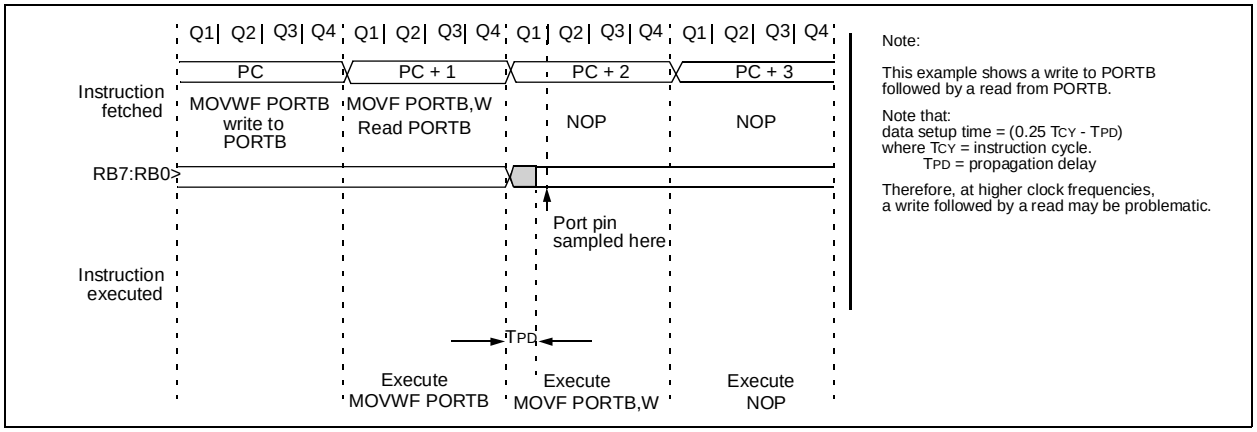


FIGURE 5-8: I/O PORT READ/WRITE TIMING



5.8.4 BIDIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation, and re-output the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs.

For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and PORTB is re-output to the output latches. If another bit of PORTB is used as a bidirectional I/O pin (say bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, over-writing the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

A pin actively outputting a '0' or '1' should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

5.8.5 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port will happen at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-8). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or an other instruction not accessing this I/O port.

6.0 SPECIAL PURPOSE REGISTERS

6.1 W (Working) Register

Holds second operand in two operand instructions and/ or supports the internal data transfer.

6.2 TRISA I/O Control Register For PORTA

Only bits (STATUS <3:0>) are available. The corresponding I/O port (f5) is only 4-bit wide.

6.3 TRISB I/O Control Register For PORTB

The I/O control registers will be loaded with the content of the W register, by executing of the TRIS f instruction. A '1' in the I/O control register will put the corresponding I/O pin into a high impedance mode. A '0' puts the contents of file register PORTA or PORTB, respectively, out on the selected I/O pins.

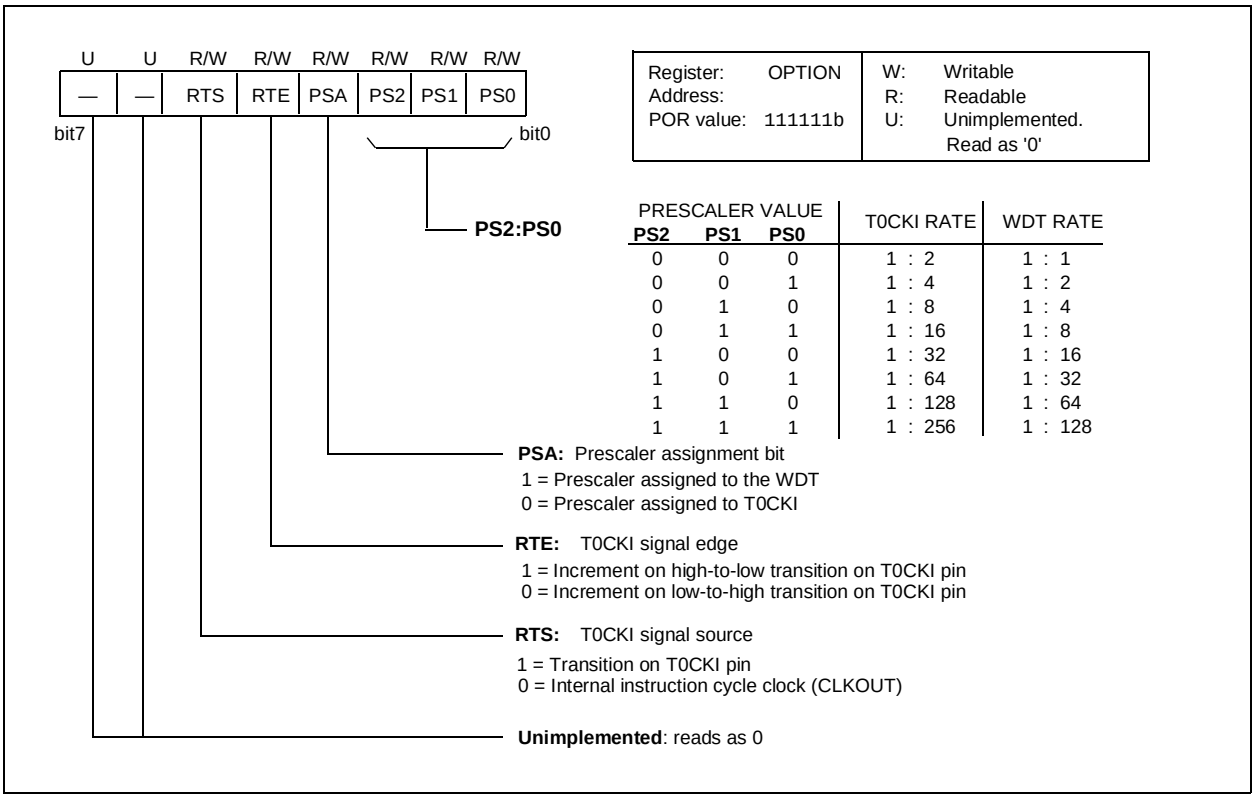
These registers are "write-only" and are set to all '1's upon a RESET.

6.4 OPTION Prescaler/T0CKI OPTION Register

Defines prescaler assignment (T0CKI or WDT), prescaler value, signal source and signal edge for the T0CKI. The OPTION register is "write-only" and is 6-bits wide.

By executing the `OPTION` instruction, the contents of the W register will be transferred to the OPTION register. Upon a RESET, the OPTION register is set.

FIGURE 6-1: OPTION REGISTER



7.0 RESET CONDITION

A RESET can be caused by applying power to the chip (power-up), pulling the MCLR input "low" or by a Watchdog Timer time-out. The device will stay in RESET as long as the Oscillator Start-Up Timer (OST) is active or the MCLR input is "low."

The Oscillator Start-Up Timer is activated as soon as MCLR input is sensed to be high. This implies that in case of Power-On Reset (POR) with MCLR tied to V_{DD} the OST starts from power-up. In case of WDT time-out, it will start at the end of the time-out (since MCLR is high). In case of MCLR reset, the OST will start when MCLR goes high. The nominal OST time-out period is 18 ms. See Section 12.0 for detailed information on OST and Power-On Reset.

During a RESET the state of the microcontroller is defined as:

a) The oscillator is running or will be started (power-up or wake-up from SLEEP).

b) All I/O port pins (RA3:RA0 and RB7:RB0) are put into the hi impedance state by setting the TRIS registers to all '1's (= input mode).

c) The Program Counter is set to all '1's.

d) The OPTION register is set to all '1's.

e) The Watchdog Timer and its prescaler are cleared.

f) The upper-three bits (page select bits) in the STATUS Register are cleared.

g) RC mode only: The CLKOUT signal on the OSC2 pin is held at a"low" level.

TABLE 7-1: RESET CONDITION FOR REGISTERS

Register	Address	Power-on Reset	MCLR Reset during: - normal operation - SLEEP WDT time-out during normal operation	Wake up from SLEEP through WDT time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
OPTION	—	0011 1111	0011 1111	0011 1111
INDF	00h	—	—	—
T0CKI	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PC	02h	1FFh MTA854XX 7FFh MTA858XX	1FFh MTA854XX 7FFh MTA858XX	1FFh MTA854XX 7FFh MTA858XX
STATUS	03h	0001 1xxx	000? ?uuu	uuu? ?uuu
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, reads as '0', ? = value depends on condition.

8.0 PRESCALER

An 8-bit counter is available as a prescaler for the T0CKI, or as a post-scaler for the Watchdog Timer. (Figure 8-1). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the T0CKI and the Watchdog Timer. Thus, a prescaler assignment for the T0CKI means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS0-PS2 bits in the OPTION register determine the prescaler assignment and pre-scale ratio. When assigned to the T0CKI, all instructions writing to the T0CKI (e.g., `CLRF T0CKI, MOVWF T0CKI, BSF T0CKI,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer.

8.1 Switching Prescaler Assignment

Changing prescaler from T0CKI to WDT

The prescaler assignment is fully under software control, (i.e., it can be changed "on the fly" during program execution). To avoid an unintended device RESET, the following instruction sequence must be executed when changing the prescaler assignment from T0CKI to WDT:

```
1. MOVLW  'xx0x0xxx'b ; Select internal
                      ; clock and select new
2. OPTION             ; prescaler value. If
                      ; new prescale value
                      ; is = '000' or '001',
                      ; then select any
                      ; other prescale
                      ; value temporarily.
3. CLRF 1             ; Clear T0CKI and
                      ; prescaler.
4. MOVLW   xxxx1xxx'b ; Select WDT, do not
                      ; change prescale
                      ; value.
5. OPTION             ;
6. CLRWDT             ; Clears WDT and
                      ; prescaler.
7. MOVLW  'xxxx1xxx'b ; Select new prescale
                      ; value.
8. OPTION             ;
```

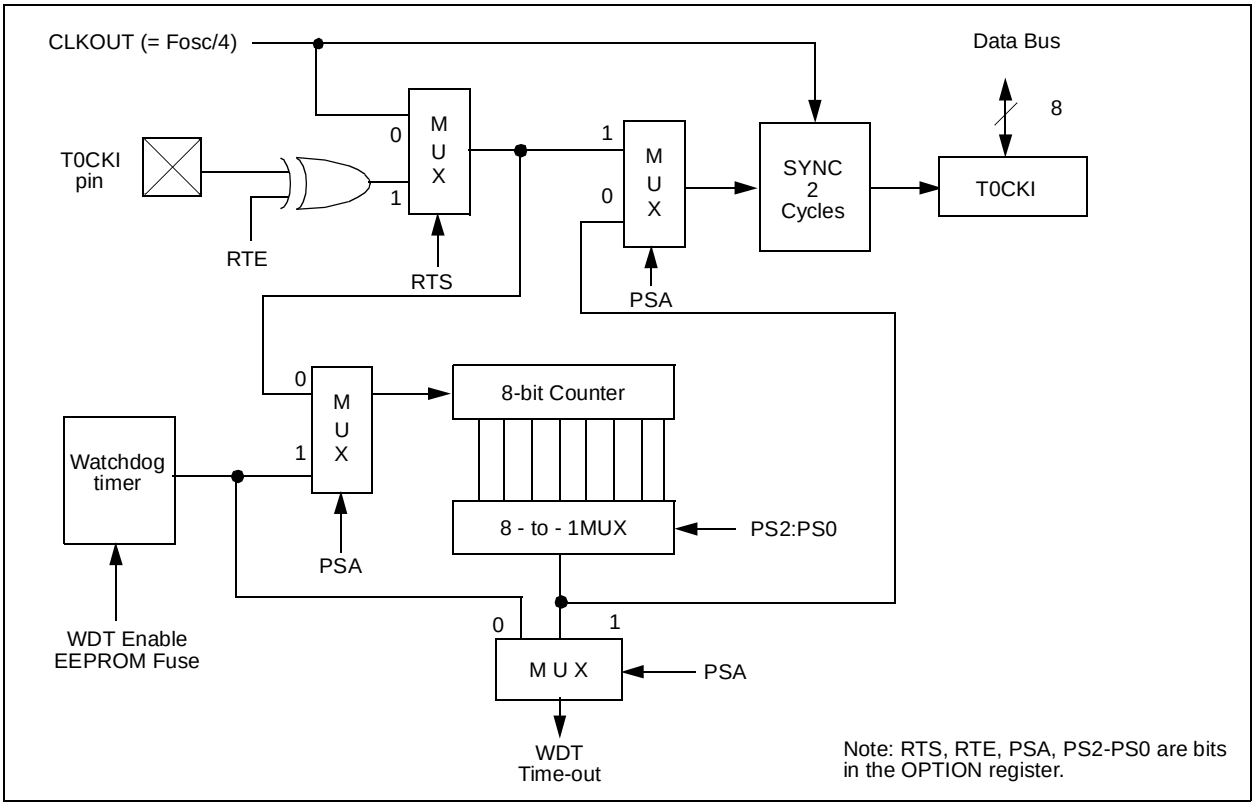
Steps 1 and 2 are only required if an external T0CKI source is used. Steps 7 and 8 are necessary only if the desired prescale value is '000' or '001'.

Changing prescaler from WDT to T0CKI

To change prescaler from WDT to T0CKI use the following sequence:

```
1. CLRWDT             ; Clear WDT and
                      ; prescaler.
2. MOVLW   'xxxx0xxx'b ; Select T0CKI, new
                      ; prescale value
                      ; and clock source
3. OPTION             ;
```

FIGURE 8-1: BLOCK DIAGRAM T0CKI/WDT PRESCALER



9.0 BASIC INSTRUCTION SET SUMMARY

Each instruction is a 12-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The Instruction Set Summary in Table 9-1 lists byte-oriented, bit-oriented, and literal and control operations.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which one of the file registers is to be utilized by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or nine bit constant or literal value.

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Notes to Table 9-1

Note 1: The ninth bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for `GOTO` (e.g., `CALL, MOVWF PC, ...`etc.). See Section 5.6.3 for details.

Note 2: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB,1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as output and is driven low by an external device, the data will be written back with a '0'.

Note 3: The instruction "TRIS f", where f = PORTA, PORTB, or PORTC causes the contents of the W register to be written to the three-state latches of the specified file (port). A '1' forces the pin to a hi impedance state and disables the output buffers.

Note 4: If this instruction is executed on the T0CKI register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the T0CKI.

TABLE 9-1: INSTRUCTION SET SUMMARY

BYTE-ORIENTED FILE REGISTER OPERATIONS

(11-6)	(5)	(4-0)
OPCODE	d	f(FILE #)

d = 0 for destination W
d = 1 for destination f

Instruction-Binary (Hex)				Name	Mnemonic, Operands		Operation	Status Affected	Notes
0001	11df	ffff	1Cf	Add W and f	**ADDWF**	**f, d**	W + f → d	C,DC,Z	1,2,4
0001	01df	ffff	14f	AND W and f	**ANDWF**	**f, d**	W ∧ f → d	Z	2,4
0000	011f	ffff	06f	Clear f	**CLRF**	**f**	0 → f	Z	4
0000	0100	0000	040	Clear W	**CLRW**	**-**	0 → W	Z	
0010	01df	ffff	24f	Complement f	**COMF**	**f, d**	$\bar{f}$ → d	Z	2,4
0000	11df	ffff	0Cf	Decrement f	**DECF**	**f, d**	f -1 → d	Z	2,4
0010	11df	ffff	2Cf	Decrement f,Skip if Zero	**DECFSZ**	**f, d**	f - 1 → d, skip if zero	None	2,4
0010	10df	ffff	28f	Increment f	**INCF**	**f, d**	f + 1 → d	Z	2,4
0011	11df	ffff	3Cf	Increment f,Skip if zero	**INCFSZ**	**f, d**	f + 1 → d, skip if zero	None	2,4
0001	00df	ffff	10f	Inclusive OR W and f	**IORWF**	**f, d**	W ∨ f → d	Z	2,4
0010	00df	ffff	20f	Move f	**MOVF**	**f, d**	f → d	Z	2,4
0000	001f	ffff	02f	Move W to f	**MOVWF**	**f**	W → f	None	1,4
0000	0000	0000	000	No Operation	**NOP**	**-**	-	None	
0011	01df	ffff	34f	Rotate left f	**RLF**	**f, d**	f(n) → d(n+1), C → d(0), f(7) → C	C	2,4
0011	00df	ffff	30f	Rotate right f	**RRF**	**f, d**	f(n) → d(n-1), C → d(7), f(0) → C	C	2,4
0000	10df	ffff	08f	Subtract W from f	**SUBWF**	**f, d**	f - W → d [f + $\overline{W}$ + 1 → d]	C,DC,Z	1,2,4
0011	10df	ffff	38f	Swap halves f	**SWAPF**	**f, d**	f(0-3) ↔ f(4-7) → d	None	2,4
0001	10df	ffff	18f	Exclusive OR W and f	**XORWF**	**f, d**	W ⊕ f → d	Z	2,4

BIT-ORIENTED FILE REGISTER OPERATIONS

(11-8)	(7-5)	(4-0)
OPCODE	b(bit #)	f(FILE #)

Instruction-Binary (Hex)				Name	Mnemonic, Operands		Operation	Status Affected	Notes
0100	bbbf	ffff	4bf	Bit Clear f	**BCF**	**f, b**	0 → f(b)	None	2,4
0101	bbbf	ffff	5bf	Bit Set f	**BSF**	**f, b**	1 → f(b)	None	2,4
0110	bbbf	ffff	6bf	Bit Test f,Skip if Clear	**BTFSC**	**f, b**	Test bit (b) in file (f): Skip if clear	None	
0111	bbbf	ffff	7bf	Bit Test f, Skip if Set	**BTFSS**	**f, b**	Test bit (b) in file (f): Skip if set	None	

LITERAL AND CONTROL OPERATIONS

(11-8)	(7-0)
OPCODE	k(literal)

Instruction-Binary (Hex)				Name	Mnemonic, Operands		Operation	Status Affected	Notes
1110	kkkk	kkkk	Ekk	AND Literal and W	**ANDLW**	**k**	k ∧ W→ W	Z	
1001	kkkk	kkkk	9kk	Call subroutine	**CALL**	**k**	PC + 1 → Stack, k → PC	None	1
0000	0000	0100	004	Clear Watchdog timer	**CLRWDT**	**-**	0 → WDT (and prescaler, if assigned)	$\overline{TO}$, $\overline{PD}$	
101k	kkkk	kkkk	Akk	Go To address (k is 9 bit)	**GOTO**	**k**	k → PC (9 bits)	None	
1101	kkkk	kkkk	Dkk	Incl. OR Literal and W	**IORLW**	**k**	k ∨ W → W	Z	
1100	kkkk	kkkk	Ckk	Move Literal to W	**MOVLW**	**k**	k → W	None	
0000	0000	0010	002	Load OPTION register	**OPTION**	**-**	W → OPTION register	None	
1000	kkkk	kkkk	8kk	Return,place Literal in W	**RETLW**	**k**	k → W, Stack → PC	None	
0000	0000	0011	003	Go into standby mode	**SLEEP**	**-**	0 → WDT, stop oscillator	$\overline{TO}$, $\overline{PD}$	
0000	0000	0fff	00f	Tri-State™ port f	**TRIS**	**f**	W→ I/O control register f	None	3
1111	kkkk	kkkk	Fkk	Excl. OR Literal and W	**XORLW**	**k**	k ⊕ W → W	Z	

Note 1: See previous page.

10.0 WATCHDOG TIMER (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device have been stopped (i.e., by executing the SLEEP instruction). During normal operation a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as a '0'.

10.1 WDT Period

The WDT has a nominal time-out period of 18 ms, (with no prescaler). These periods vary with temperature, V_{DD} and process variations from part to part (see DC specifications). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out and generating a device RESET.

The $\overline{TO}$ bit (STATUS<4>) will be cleared upon a WDT time-out.

10.2 WDT Programming Considerations

At worst case conditions (V_{DD} = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

FIGURE 10-1: WATCHDOG TIMER BLOCK DIAGRAM

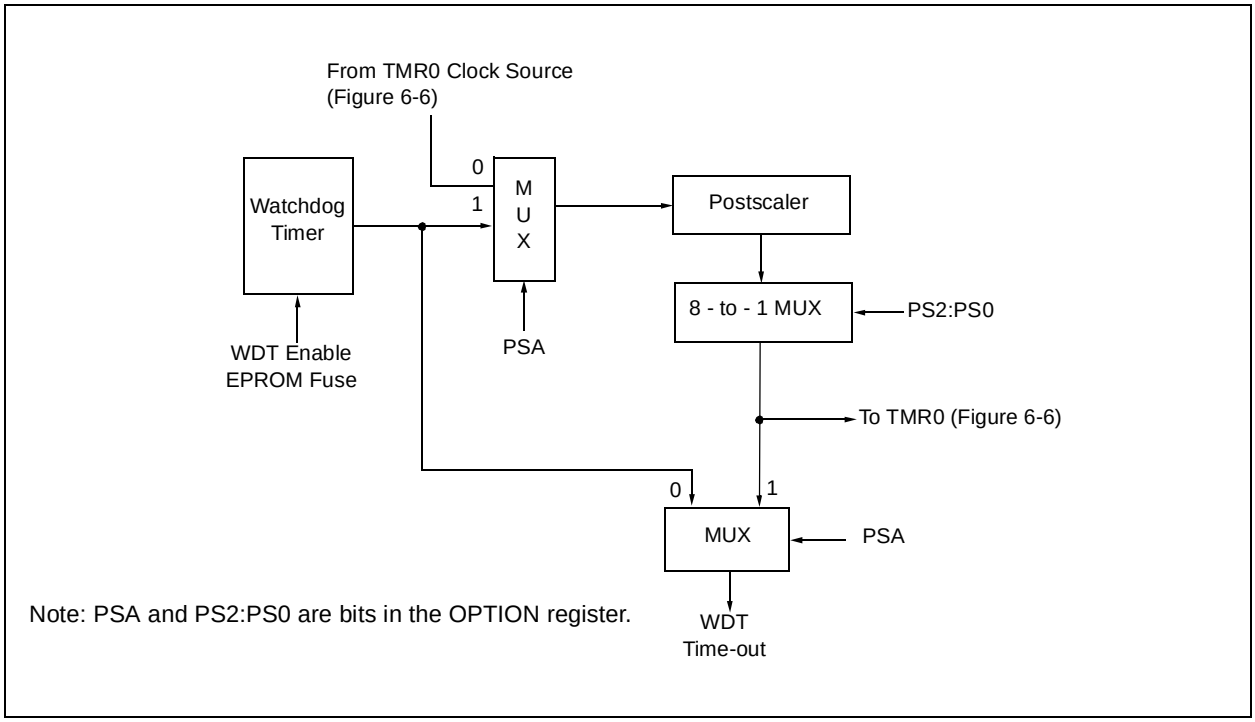


TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Config. Word	CP	CP	CP	CP	CP	WDTE	FOSC1	FOSC0
OPTION	__	__	T0CS	T0SE	PSA	PS2	PS1	PS0

Note 1: CP7:CP4 are used by the PIC16CCR58A only. Unused in all other devices.
2: Shaded cells are not used by the Watchdog Timer.

11.0 OSCILLATOR CONFIGURATIONS

11.0.1 OSCILLATOR TYPES

The PIC16C5X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

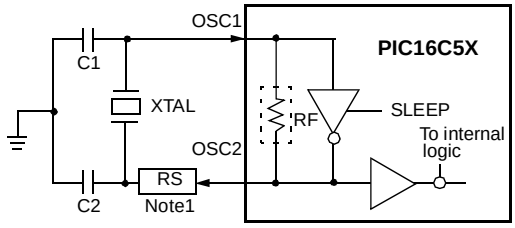
- LP: Low Power Crystal
- XT: Crystal/Resonator
- HS: High Speed Crystal/Resonator
- RC: Resistor/Capacitor

Note:	The MTA854XX operates at 4 MHz and the MTA858XX operates up to 10 MHz.

11.0.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 11-1). The PIC16C5X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source drive the OSC1/CLKIN pin (Figure 11-2).

FIGURE 11-1: CRYSTAL /CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



See Table 11-1 and Table 11-2 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 11-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC MODE)

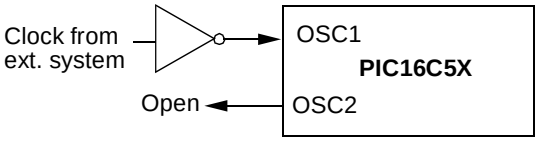


TABLE 11-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
XT	455 kHz	22-100 pF	22-100 pF
	2.0 MHz	15-68 pF	15-68 pF
	4.0 MHz	15-68 pF	15-68 pF
HS	8.0 MHz	10-68 pF	10-68 pF

These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

TABLE 11-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz†	33-68 pF	33-68 pF
	200 kHz	15-47 pF	15-47 pF
XT	100 kHz	47-100 pF	47-100 pF
	2 MHz	15-33 pF	15-33 pF
	4 MHz	10-33 pF	10-33 pF
HS	8 MHz	15-47 pF	15-47 pF

† For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.

These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

11.1 External Crystal Oscillator circuit

A prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 11-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

FIGURE 11-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

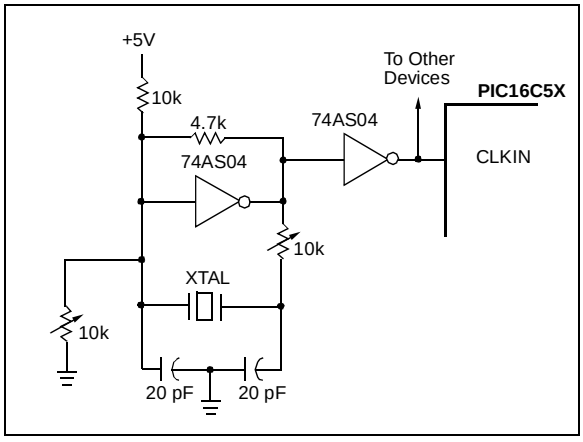
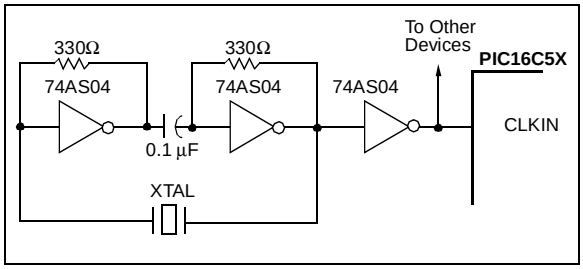


Figure 11-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 11-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



11.1.1 RC OSCILLATOR

For timing insensitive applications the RC device option offers additional cost savings. RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 11-5 shows how the R/C combination is connected to the PIC16C5X. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g., 1 MΩ) the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping Rext between 3 kΩ and 100 kΩ.

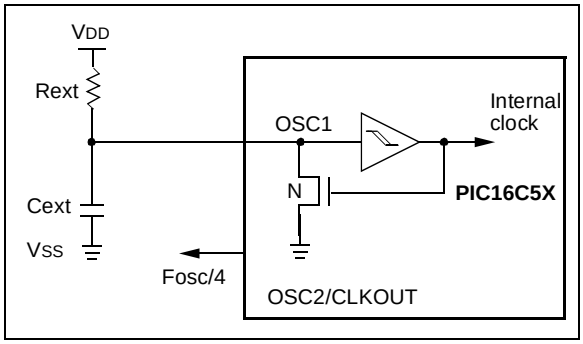
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 15.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 15.0 for variation of oscillator frequency due to V_{DD} for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and V_{DD} values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic.

FIGURE 11-5: RC OSCILLATOR MODE



12.0 RESET

The PIC16C5X differentiates between various kinds of resets:

- Power-On Reset (POR)
- MCLR reset during normal operation
- MCLR reset during SLEEP
- WDT time-out reset

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in all other resets. Most other registers are reset to a "reset state" on Power-On Reset (POR), MCLR or a WDT reset. Note that the PIC16C5X does not differentiate between a WDT reset during SLEEP or during normal operation. The TO and PD bits are set or cleared depending upon the reset situation (Table 12-1). These bits may be used to determine the nature of the reset. See Table 12-3 for a full description of reset states of all registers.

Figure 12-1 shows the simplified block diagram of the on-chip reset circuit.

12.1 Power-On Reset (POR) and Device-Reset Timer (DRT)

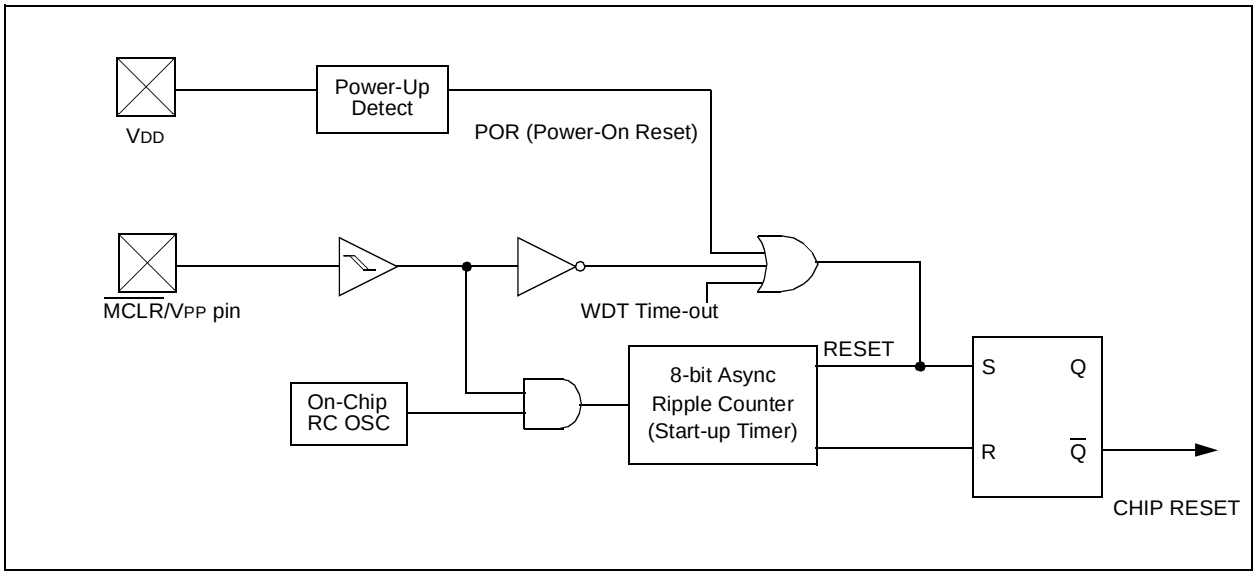
12.1.1 POWER-ON RESET (POR)

The PIC16C5X family incorporates an on-chip Power-On Reset (POR) circuitry which provides an internal chip reset for most power-up situations. To use this feature the user merely needs to tie the MCLR/VPP pin to VDD. Figure 12-8 shows the electrical structure of TMR0 inputs. The Power-On Reset circuit and the Device Reset Timer circuit are closely related. On power-up the reset latch is set and the DRT is reset. The DRT timer begins counting once it detects MCLR to be high. After the time-out period, which is typically 18 ms, it will reset the reset-latch and thus end the on-chip reset signal.

Figure 12-2 and Figure 12-3 are two power-up situations with relatively fast rise time on VDD. VDD is allowed to rise and stabilize (Figure 12-2) before bringing MCLR high. The chip will actually come out of reset (TDRT msec) after MCLR goes high. The on-chip Power-On Reset feature (Figure 12-3) is being used (MCLR and VDD are tied together). VDD is stable before the start-up timer times out and there is no problem in getting a proper reset. Figure 12-4 depicts a potentially problematic situation where VDD rises too slowly. In this situation, when the start-up timer times out, VDD has not reached the VDD (min) value and the chip is, therefore, not guaranteed to function correctly.

To summarize, the on-chip POR is guaranteed to work if the rate of rise of VDD is no slower than 0.05V/ms, and VDD starts from 0V. The on-chip POR time delay is too short for low frequency crystals which require much longer than 18 ms to start-up and stabilize. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times.

FIGURE 12-1: ON-CHIP RESET CIRCUIT BLOCK DIAGRAM



12.1.2 DEVICE RESET TIMER (DRT)

The Device Reset Timer provides a fixed 18 ms nominal time-out on RESET. The Device Reset Timer operates with an internal RC oscillator. The processor is kept in RESET as long as the DRT is active. The DRT delay allows V_{DD} to rise above V_{DD} min., and allows the oscillator to stabilize.

Oscillator circuits based on crystals or ceramic resonators require a certain time after power-up to establish a stable oscillation. The on-chip DRT keeps the device in a RESET condition for approximately 18 ms after the voltage on the $\overline{MCLR}/V_{PP}$ pin has reached a logic high (V_{IHMC}) level. Thus, external RC networks connected to the $\overline{MCLR}$ input are not required in most cases, allowing for savings in cost-sensitive and/or space restricted applications.

The Device Reset time delay will vary from chip to chip and due to V_{DD}, temperature, and process variation.

The DRT will also be triggered upon a WDT time-out. This is particularly important for applications using the WDT to waken the PIC16C5X from SLEEP automatically.

12.1.3 TIME-OUT SEQUENCE

Table 12-2 lists the reset conditions for the special function registers while Table 12-3 lists the reset conditions for all the registers.

TABLE 12-1: $\overline{TO}/\overline{PD}$ STATUS AFTER RESET

$\overline{TO}$	$\overline{PD}$	RESET was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
1	0	$\overline{MCLR}$ wake-up from SLEEP
1	1	Power-up
u	u	= Low pulse on $\overline{MCLR}$ input

The $\overline{TO}$ and $\overline{PD}$ bits maintain their status (u) until a reset occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$ and $\overline{PD}$ status bits.

TABLE 12-2: RESET CONDITIONS FOR SPECIAL REGISTERS

Condition	STATUS Addr: 03h	PCL Addr: 02h
Power-On Reset	0001 1xxx	1111 1111
$\overline{MCLR}$ reset during normal operation	000u uuuu(1)	1111 1111
$\overline{MCLR}$ reset during SLEEP	0001 0uuu	1111 1111
WDT reset during SLEEP	0000 0uuu	1111 1111
WDT reset during normal operation	0000 1uuu	1111 1111

Legend: u = unchanged, x = unknown, - = unimplemented read as '0'.
Note 1: The $\overline{TO}$ and $\overline{PD}$ bits retain their last value until one of the other reset conditions occur.
2: The CLRWDT instruction will set the $\overline{TO}$ and $\overline{PD}$ bits

TABLE 12-3: RESET CONDITIONS FOR ALL REGISTERS

Register	Address	Power-on Reset	$\overline{MCLR}$ or WDT Reset
W	N/A	xxxx xxxx	uuuu uuuu
TRIS	N/A	1111 1111	1111 1111
OPTION	N/A	--11 1111	--11 1111
INDF	00h	—	—
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000? ?uuu (1)
FSR	04h	xxxx xxxx	uuuu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
PORTC	07h	xxxx xxxx	uuuu uuuu
General Purpose register files	08-7Fh	xxxx xxxx	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented read as '0', ? = value depends on condition.
Note 1: See Table 12-2 for reset value for specific conditions.

FIGURE 12-2: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): Case 1

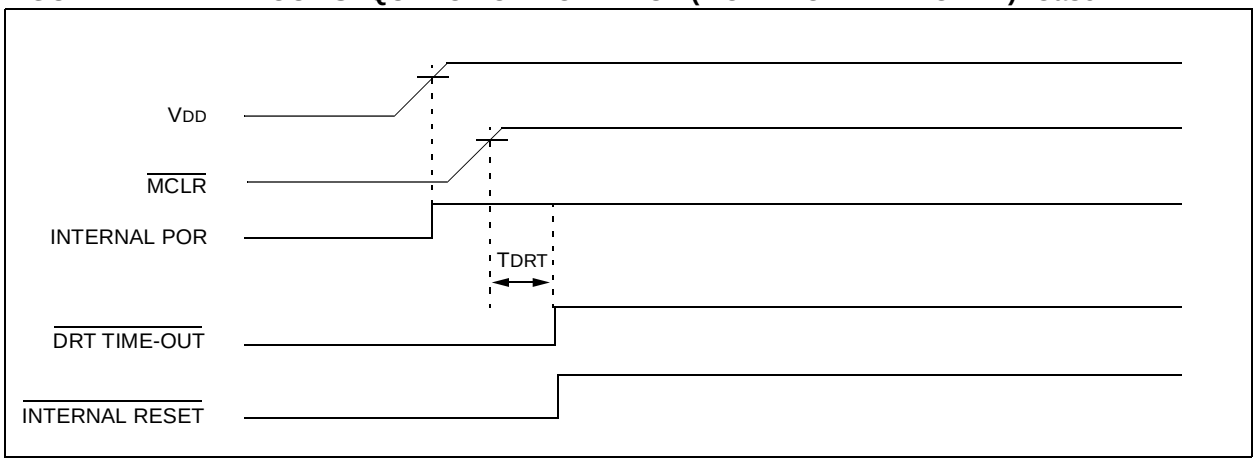


FIGURE 12-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): Case 2

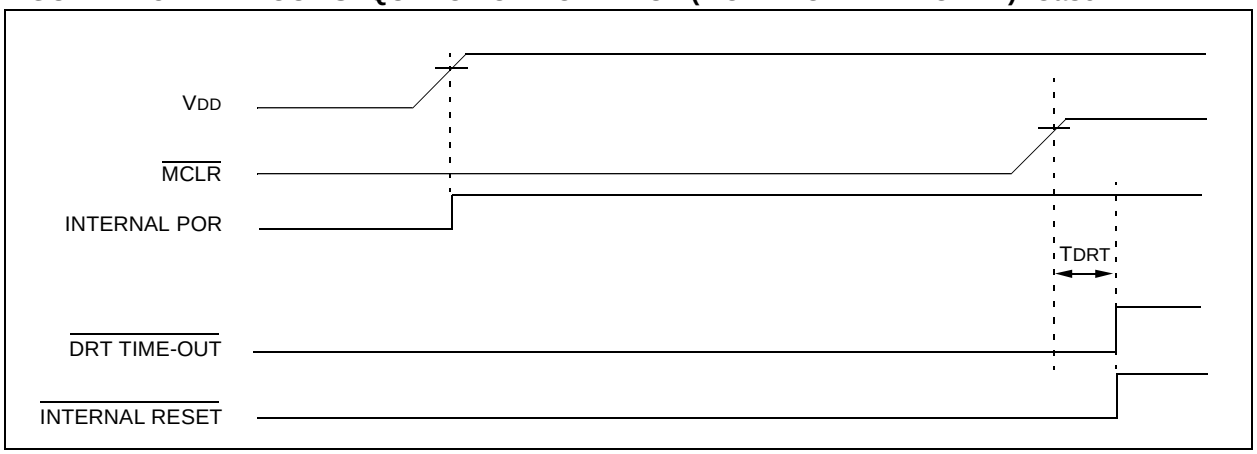


FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD})

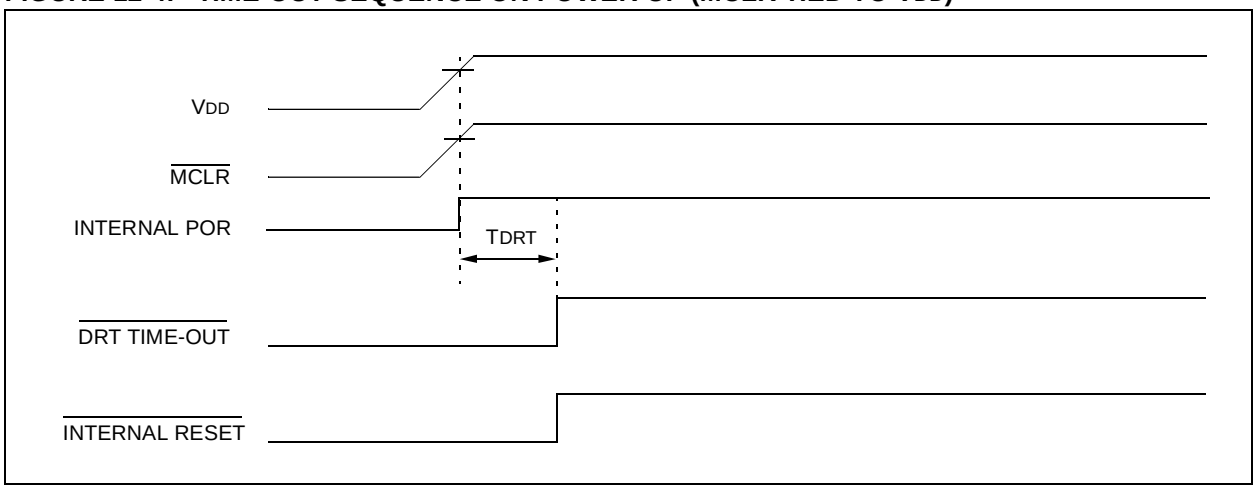
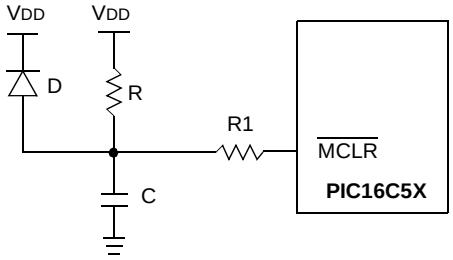


FIGURE 12-5: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

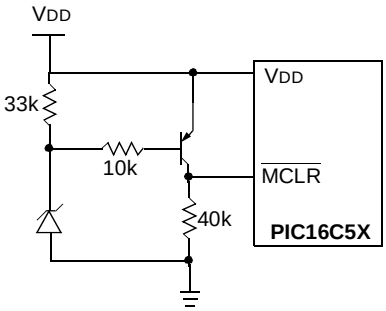


Note 1: External Power-On Reset circuit is required only if VDD power-up is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on MCLR pin is 5 μA). A larger voltage drop will degrade VIH level on the MCLR/VPP pin.

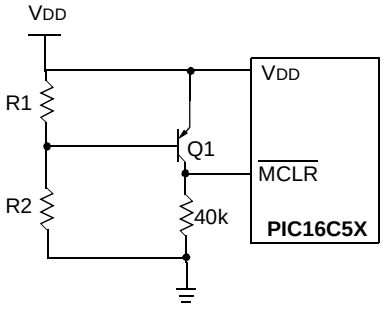
3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to ESD or EOS.

FIGURE 12-6: BROWN-OUT PROTECTION CIRCUIT 1



This circuit will activate reset when VDD goes below Vz + 0.7V (where Vz = Zener voltage).

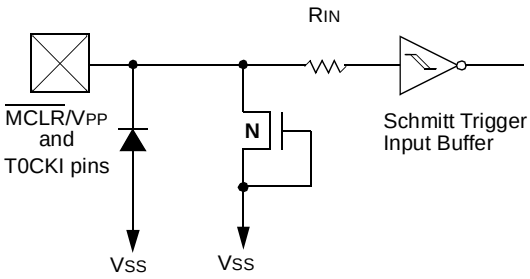
FIGURE 12-7: BROWN-OUT PROTECTION CIRCUIT 2



This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V.$$

FIGURE 12-8: ELECTRICAL STRUCTURE OF THE MCLR/VPP AND T0CKI PINS



13.0 POWER-DOWN MODE (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{TO}$ bit (STATUS<4>) is set, the $\overline{PD}$ bit (STATUS<3>) is cleared, and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low, or hi-impedance).

It should be noted that a RESET generated by a WDT time-out does not drive the $\overline{MCLR}/V_{PP}$ pin low.

For lowest current consumption while powered down, the T0CKI input should be at V_{DD} or V_{SS} and the $\overline{MCLR}/V_{PP}$ pin must be at a logic high level (V_{IHMC}).

13.0.1 WAKE-UP FROM SLEEP

The device can wake from SLEEP through one of the following events:

1. An external reset input on the $\overline{MCLR}/V_{PP}$ pin.
2. A WDT time-out reset (if WDT was enabled).

Both of these events cause a device reset. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of device reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked.

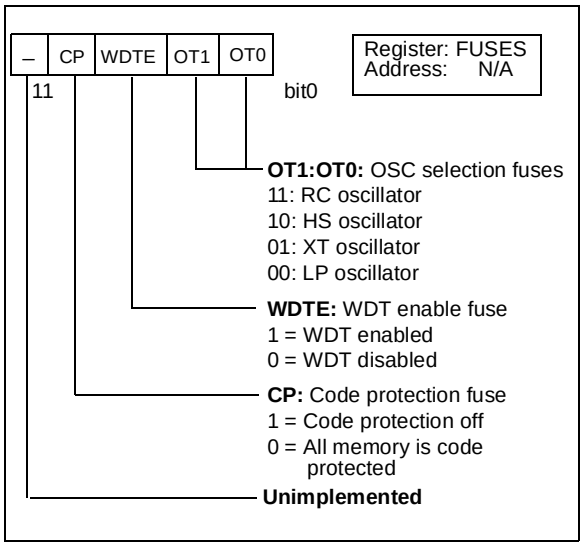
The WDT is cleared when the device wakes from SLEEP, regardless of the wake-up source.

14.0 CONFIGURATION FUSES

The configuration EPROM consists of four EPROM fuses.

Two are for the selection of the oscillator type, one is the watchdog timer enable fuse and one is the code protection fuse.

FIGURE 14-1: CONFIGURATION WORD



14.1 ID Locations

Four memory locations 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.

Use only the lower 4-bits of the ID locations and always program the upper 8-bits as '1's.

*** Note:**	Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

14.2 Code Protection

The code in the program memory can be protected by clearing the code protect bits.

In code protected mode, the configuration word will not be protected, allowing reading of all bits.

14.2.1 PIC16C54A AND PIC16C58A

Once code protected, all memory locations read out in a scrambled fashion. For EPROM devices, program memory locations 40h and above cannot be further programmed. However, the first 64 locations, 00h - 3Fh, may be programmed. These locations are not considered secure.

14.2.2 PIC16CR58A

In a protected device, program memory locations 00h-3Fh read out normally. Locations 40h and higher cannot be read out.

15.0 ELECTRICAL CHARACTERISTICS

15.1 Absolute Maximum Ratings*

Ambient temperature under bias ..... -55°C to +125°C

Storage Temperature ..... -65°C to +150°C

Voltage on any pin with respect to V_{SS} (except V_{DD} and $\overline{MCLR}$) ..... -0.6V to (V_{DD} + 0.6V)

Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V

Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2) ..... 0 to +14V

Total power Dissipation (Note 1) ..... 800 mW

Maximum Current out of V_{SS} pin ..... 150 mA

Maximum Current into V_{DD} pin ..... 50 mA

Maximum Current into an input pin ..... ±500 µA

Maximum Output Current sunk by any I/O pin ..... 25 mA

Maximum Output Current sourced by any I/O pin ..... 20 mA

Maximum Output Current sourced by a single I/O port (PortA or B) ..... 40 mA

Maximum Output Current sunk by a single I/O port (PortA or B) ..... 50 mA

Note 1: Total power dissipation should not exceed 800 mW for the package. Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

*** 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.

TABLE 15-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C54A-04
RC	V_{DD}: 3.0V to 6.25V I_{DD}: 2.4 mA max. at 5.5V I_{PD}: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.
XT	V_{DD}: 3.0V to 6.25V I_{DD} 2.4 mA max. at 5.5V I_{PD}: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 2.4 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.
LP	V_{DD}: 3.0V to 6.25V I_{DD}: 14 µA typ. at 32kHz, 3.0V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 200 kHz max.

*** Note:** The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

15.2 DC Characteristics

TABLE 15-2: DC CHARACTERISTICS OF MICROCONTROLLER MTA854XX-04 (INDUSTRIAL)

DC CHARACTERISTICS **Power Supply Pins**		**Standard Operating Conditions** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Characteristic	**Sym**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 4.5		6.25 5.5	V V	XT, RC and LP options HS option
RAM Data Retention Voltage (Note 3)	VDR	1.5			V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR		VSS		V	See POR section in microcontroller datasheet for details on Power-On Reset
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*			V/ms	See POR section in microcontroller datasheet for details on Power-On Reset
Supply Current (Note 2)	IDD		1.8	2.4	mA	XT and RC options (C-04) FOSC = 4 MHz, VDD = 5.5V
			5.8	13	mA	HS option (C-10) FOSC = 10 MHz, VDD = 5.5V
			17	70	µA	LP option, Industrial, EEPROM standby FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 4) WDT enabled WDT disabled	IPD		 5 13	 14 5	 µA µA	 VDD = 3.0V, Industrial VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

Note 1: Data in the column labeled "Typ" is based on characterization results at 25°C. These parameters are for design guidance only and are not tested for, or guaranteed by Microchip Technology.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.
a) The test conditions for all IDD measurements in active operation mode are:
OSC1= external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, RT = VDD,
MCLR = VDD; WDT enabled/disabled as specified; EEPROM in write condition (except LP mode).
b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode and SDA and SCL are tied to VSS.

Note 3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 4: 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, with all I/O pins in hi-impedance state and tied to VDD and VSS; SDA and SCL tied to VSS.

Note 5: Does not include current through Rext. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kΩ.

TABLE 15-3: DC CHARACTERISTICS OF MICROCONTROLLER MTA858XX-04 (INDUSTRIAL) MTA858XX-10 (INDUSTRIAL)

DC CHARACTERISTICS **Power Supply Pins**		**Standard Operating Conditions** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Characteristic	**Sym**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 4.5		6.25 5.5	V V	XT, RC and LP options HS option
RAM Data Retention Voltage (Note 3)	VDR	1.5			V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR		VSS		V	See POR section in microcontroller datasheet for details on Power-On Reset
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*			V/ms	See POR section in microcontroller datasheet for details on Power-On Reset
Supply Current (Note 2)	IDD		1.8	2.4	mA	XT and RC options (C-04) FOSC = 4 MHz, VDD = 5.5V
			5.8	13	mA	HS option (C-10) FOSC = 4 MHz, VDD = 5.5V
			17	40	µA	LP option, Industrial, EEPROM standby FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 4) WDT enabled WDT disabled	IPD		 5 0.8	 14 12	 µA µA	 VDD = 3.0V, Industrial VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

Note 1: Data in the column labeled "Typ" is based on characterization results at 25°C. These parameters are for design guidance only and are not tested for, or guaranteed by Microchip Technology.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are: OSC1= external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, except RB7 driving '1' for SEE VDD, RT = VDD, $\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified; EEPROM in write condition (except LP mode).

b) For standby current measurements, the conditions are the same, except that the device is in SLEEP mode and SDA and SCL are tied to VSS.

Note 3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 4: 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, with all I/O pins in hi-impedance state and tied to VDD and VSS; SDA and SCL tied to VSS.

Note 5: Does not include current through Rext. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kΩ.

TABLE 15-4: DC CHARACTERISTICS OF INPUTS/OUTPUTS: MTA854XX-04 (INDUSTRIAL) MTA858XX-04 (INDUSTRIAL) MTA858XX-10 (INDUSTRIAL

DC CHARACTERISTICS **All Pins Except Power Supply**		**Standard Operating Conditions** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Characteristic	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	VSS		0.2 VDD	V	Pin at hi-impedance
$\overline{MCLR}$ (Schmitt trigger)		VSS		0.15 VDD	V	
T0CKI (Schmitt trigger)		VSS		0.15 VDD	V	
OSC1 (Schmitt trigger)		Vss		0.15 VDD	V	RC option only (Note 5)
OSC1, SCL		Vss		0.3 VDD	V	XT, HS and LP options
Input High Voltage						
I/O ports	VIH	0.45 VDD		VDD	V	For all VDD (Note 6)
		2.0		VDD	V	4.0V < VDD ≤ 5.5V (Note 6)
		0.36VDD		VDD	V	VDD > 5.5V
$\overline{MCLR}$ (Schmitt trigger)		0.85 VDD		VDD	V	
T0CKI (Schmitt trigger)		0.85 VDD		VDD	V	
OSC1 (Schmitt trigger)		0.85 VDD		VDD	V	RC option only (Note 5)
OSC1, SCL		0.7 VDD		VDD	V	XT, HS, and LP options
Input Leakage Current (Note 4)						
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance
$\overline{MCLR}$		-5			µA	VPIN = VSS + 0.25V (Note 3)
$\overline{MCLR}$			0.5	+5	µA	VPIN = VDD (Note 3)
T0CKI		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
SDA, SCL		-10		10	µA	XT, HS and LP options
Output Low Voltage						
I/O ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT						
(RC option only)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
SDA				0.4	V	IOL = 3.0 mA, VDD = 3.0V
Output High Voltage						
I/O ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT						
(RC option only)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

Note 1: Data in "Typ" column is based on characterization results at 25°C. These parameters are for design guidance only and are not tested for, or guaranteed by Microchip Technology.

Note 2: Total power dissipation as stated under absolute maximum ratings must not be exceeded.

Note 3: 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.

Note 4: Negative current is defined as coming out of the pin.

Note 5: In RC oscillator mode, the OSC1 pin is a Schmitt trigger input. It is not recommended that the microcontroller be driven with external clock in RC mode.

Note 6: The user may use better of the two specifications.

15.3 AC Characteristics

TABLE 15-5: AC CHARACTERISTICS OF MICROCONTROLLER: MTA854XX-04 (INDUSTRIAL)
MTA858XX-04 (INDUSTRIAL)
MTA858XX-10 (INDUSTRIAL)

AC CHARACTERISTICS		Standard Operating Conditions Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Characteristic	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
External CLOCKIN Frequency (Note 2)	FOSC	DC		4	MHz	RC mode
		DC		4	MHz	XT mode
		DC		20	MHz	HS mode (Com/Ind) (Note 5)
		DC		40	kHz	LP mode
Oscillator Frequency (Note 2)	FOSC	DC		4	MHz	RC mode
		0.1		4	MHz	XT mode
		4		20	MHz	HS mode (Com/Ind) (Note 5)
		DC		40	kHz	LP mode
Instruction Cycle Time (Note 2)	TCY	1.0	4/Fosc	DC	µs	RC mode
		1.0		DC	µs	XT mode
		0.2		DC	µs	HS mode (Note 5)
		100		DC	µs	LP mode
External Clock in Timing (Note 4)						
Clock in (OSC1) High or Low time						
XT oscillator type	TCKHLXT	50*			ns	
LP oscillator type	TCKHLLP	2*			µs	
HS oscillator type	TCKHLHS	20*			ns	
Clock in (OSC1) Rise or fall time						
XT oscillator type	TCKRFXT	25*			ns	
LP oscillator type	TCKRFLP	50*			ns	
HS oscillator type	TCKRFHS	25*			ns	
RESET Timing						
MCLR Pulse Width (low)	TMCL	100*			ns	
T0CKI Input Timing, No prescaler						
T0CKI High Pulse Width	TRTH	0.5 TCY+20*			ns	Note 3
T0CKI Low Pulse Width	TRTL	0.5 TCY+20*			ns	Note 3

* Guaranteed by characterization but not tested.

Note 1: Data in the column labeled "Typ" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: 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 "minimum" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "maximum" cycle time limit is "DC" (no clock) for all devices.

Note 3: For a detailed explanation of T0CKI input clock requirements see microcontroller datasheet Section 5.2.1.

Note 4: Clock-in high-time is the duration for which clock input is at VIHOSC or higher.
Clock-in low-time is the duration for which clock input is at VILOSC or lower.

Note 5: This HS specification is only for the -20 device. The -10 device has a maximum of 10 MHz and the -04 device has a maximum of 4 MHz.

AC CHARACTERISTICS		Standard Operating Conditions Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Characteristic	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
T0CKI Input Timing, With prescaler						
T0CKI High Pulse Width	TRTH	10*			ns	Note 3
T0CKI Low Pulse Width	TRTL	10*			ns	Note 3
T0CKI period	TRTP	$\frac{T_{CY}+40^*}{N}$			ns	Note 3. Where N = prescale value (2,4,...,256)
Watchdog Timer Time-out Period, No prescaler	TWDT	9*	18*	30*	ms	VDD = 5.0V
Oscillation Start-up Timer Period	TOST	9*	18*	30*	ms	VDD = 5.0V
I/O Timing						
I/O pin input valid before CLKOUT↑ (RC mode)	TDS	0.25 TCY+30*			ns	
I/O Pin input hold after CLKOUT↑ (RC mode)	TDH	0*			ns	
I/O pin output valid after CLKOUT↓ (RC Mode)	TPD			40*	ns	
I/O pin input valid before OSC↑ (I/O setup time)	TioV2osH	TBD			ns	
OSC1↑ to I/O pin input invalid (I/O holdup time)	TosH2ioL	TBD			ns	
OSC1↑ to I/O pin output valid	TosH2ioV			TBD	ns	
I/O pin output rise time	TioR			TBD	ns	
I/O pin output fall time	TioF			TBD	ns	
Capacitive loading specs on output pins						In Xt, HS or LP modes when external clock is used to drive
OSC2 pin	COSC2			15	pF	OSC1
All I/O pins	CIO			50	pF	Excludes RB7 on MTA85X1X devices

* Guaranteed by characterization but not tested.

Note 1: Data in the column labeled "Typ" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

Note 2: 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 "minimum" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "maximum" cycle time limit is "DC" (no clock) for all devices.

Note 3: For a detailed explanation of T0CKI input clock requirements see microcontroller datasheet Section 5.2.1.

Note 4: Clock-in high-time is the duration for which clock input is at VIHOSC or higher.
Clock-in low-time is the duration for which clock input is at VILOSC or lower.

Note 5: This HS specification is only for the -20 device. The -10 device has a maximum of 10 MHz and the -04 device has a maximum of 4 MHz.

TABLE 15-6: AC CHARACTERISTICS OF EEPROM

AC CHARACTERISTICS Parameter	Symbol	Standard Mode		Vcc = 4.5-5.5V Fast Mode		Units	Remarks
		Min	Max	Min	Max		
Clock frequency	FCLK	0	100	0	400	kHz	
Clock high time	THIGH	4000	—	600	—	ns	
Clock low time	TLOW	4700	—	1300	—	ns	
SDA and SCL rise time	TR	—	1000	UF	300	ns	Note 2
SDA and SCL fall time	TF	—	300	UF	300	ns	Note 2
START condition hold time	THD:STA	4000	—	600	—	ns	After this period the first clock pulse is generated
START condition setup time	TSU:STA	4700	—	600	—	ns	Only relevant for repeated START condition
Data input hold time	THD:DAT	0	—	0	—	ns	Note 1
Data input setup time	TSU:DAT	250	—	100	—	ns	
STOP condition setup time	TSU:STO	4000	—	600	—	ns	
Output valid from clock	TAA	—	3500	—	900	ns	Note 1
Bus free time	TBUF	4700	—	1300	—	ns	Time the bus must be free before a new transmission can start
Output fall time from VIH minimum to VIL maximum	TOF	—	250	20+0.1 CB	250	ns	Note 2, CB ≤ 100 pF
Input filter spike suppression (SDA and SCL pins)	TSP	N/A	N/A	0	50	ns	Note 3
Write cycle time	TWR	—	10	—	10	ms	Byte or Page mode
Endurance	—	100,000	—	100,000	—	E/W cycles	

Note 1: As a transmitter, the device must provide an internal minimum delay time to bridge the undefined region (minimum 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

Note 2: Not 100% tested. CB = total capacitance of one bus line in pF.

Note 3: The combined TSP and VHYS specifications are due to new Schmitt trigger inputs which provide improved noise spike suppression. This eliminates the need for a TI specification for standard operation.

15.4 Electrical Structure of Pins

FIGURE 15-1: ELECTRICAL STRUCTURE OF I/O PINS (RA, RB)

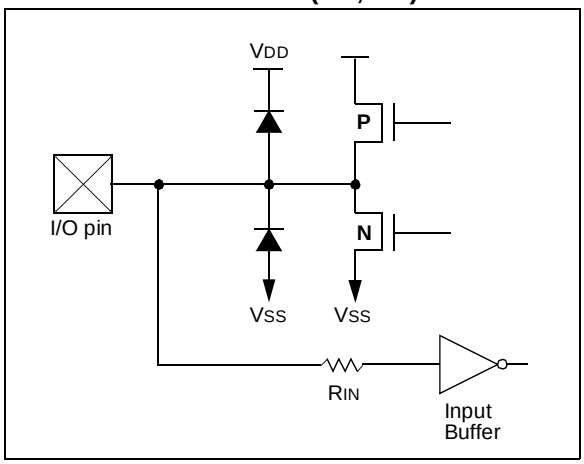
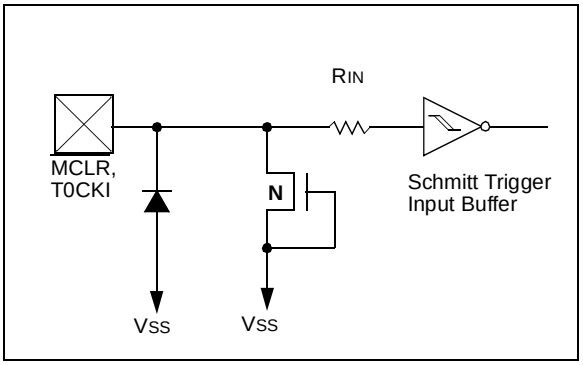


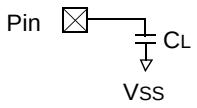
FIGURE 15-2: ELECTRICAL STRUCTURE OF MCLR AND T0CKI PINS



Notes to Figure 15-1 and Figure 15-2: The diodes and the grounded gate (or output driver) NMOS device are carefully designed to protect against ESD (Electrostatic discharge) and EOS (Electrical overstress). RIN is a small resistance to further protect the input buffer from ESD.

15.5 Timing Diagrams

FIGURE 15-3: LOAD CONDITIONS



C_L = 50 pF for all pins except OSC2
15 pF for OSC2 in XT, HS or
LP modes when external clock
is used to drive OSC1.

FIGURE 15-4: EXTERNAL CLOCK TIMING

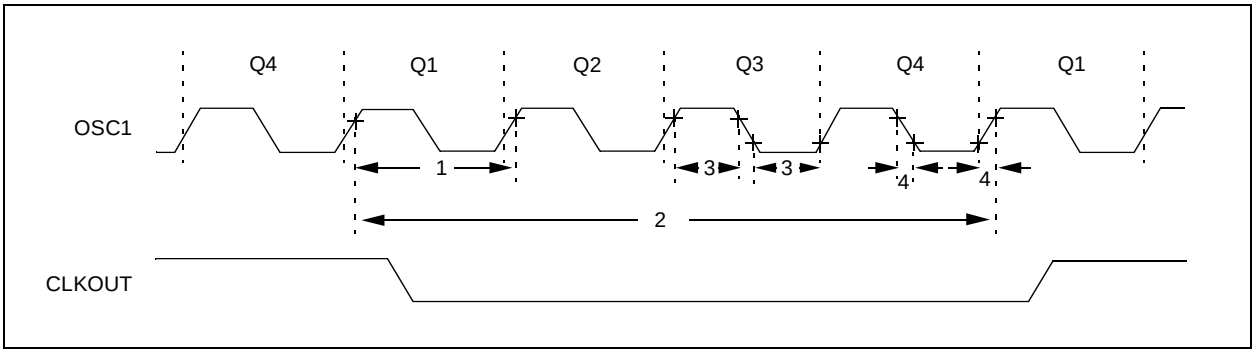


TABLE 15-7: EXTERNAL CLOCK TIMING REQUIREMENTS

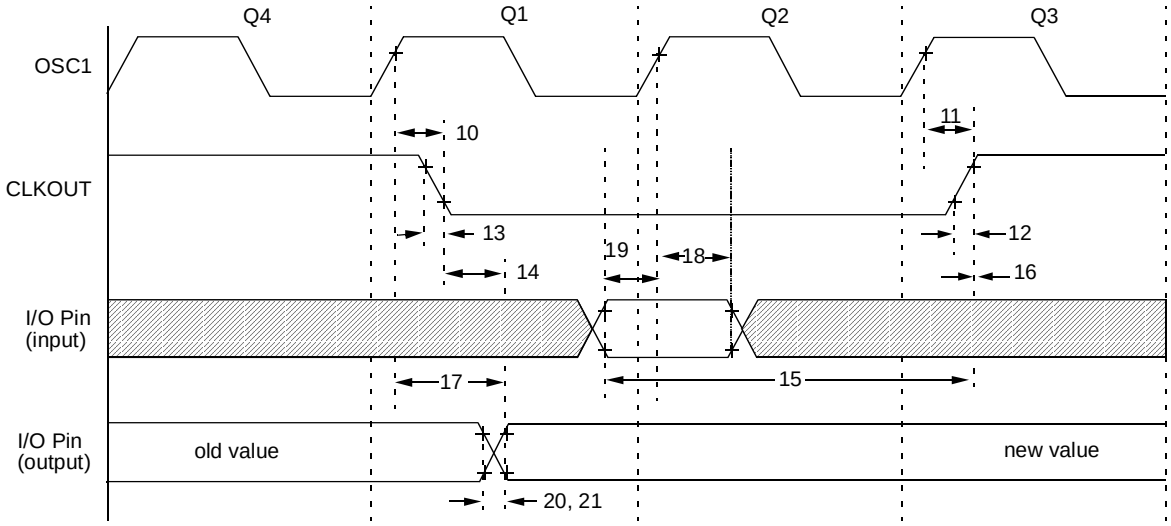
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C5XA-04)
			DC	—	10	MHz	HS osc mode (PIC16C5XA-10)
			DC	—	20	MHz	HS osc mode (PIC16C5XA-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C5XA-04)
			4	—	10	MHz	HS osc mode (PIC16C5XA-10)
			4	—	20	MHz	HS osc mode (PIC16C5XA-20)
			5	—	200	kHz	LP osc mode
1	TOSC	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C5XA-04)
			100	—	—	ns	HS osc mode (PIC16C5XA-10)
			50	—	—	ns	HS osc mode (PIC16C5XA-20)
			5.0	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C5XA-04)
			100	—	250	ns	HS osc mode (PIC16C5XA-10)
			50	—	250	ns	HS osc mode (PIC16C5XA-20)
			5	—	200	µs	LP osc mode
2	TCY	Instruction Cycle Time (Note 1)	1.0	—	DC	µs	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† 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: 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 an 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/CLKIN pin.

When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 15-5: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT.

TABLE 15-8: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5 TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	Note 2
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	Note 2
21	TioF	Port output fall time	—	10	25	ns	Note 2

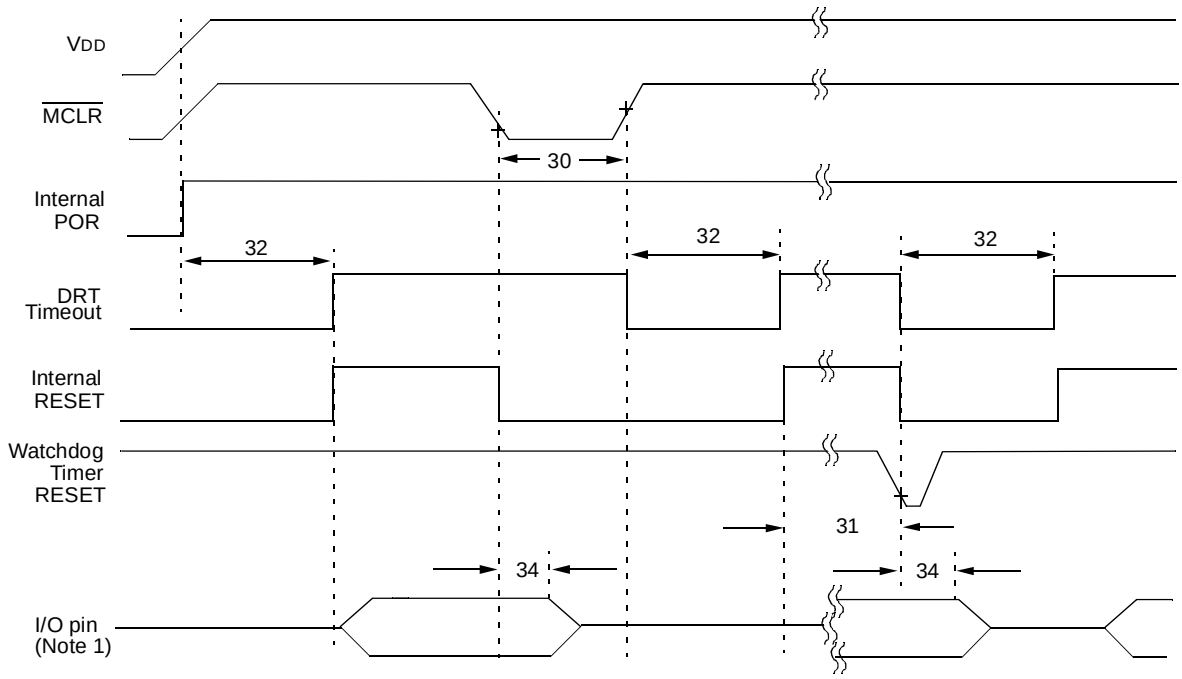
* These parameters are characterized but not tested.

† 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: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

2: See Figure 15-3 for loading conditions.

FIGURE 15-6: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of hi-impedance mode by enabling the output drivers in software.

TABLE 15-9: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	9*	18	30*	ms	VDD = 5V, -40°C to +125°C
32	TDRT	Device Reset Timer Period	9*	18*	30*	ms	VDD = 5V, -40°C to +125°C
34	TioZ	I/O Hi-impedance from $\overline{\text{MCLR}}$ Low or WDT timeout			100	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.

FIGURE 15-7: TIMER0 CLOCK TIMINGS

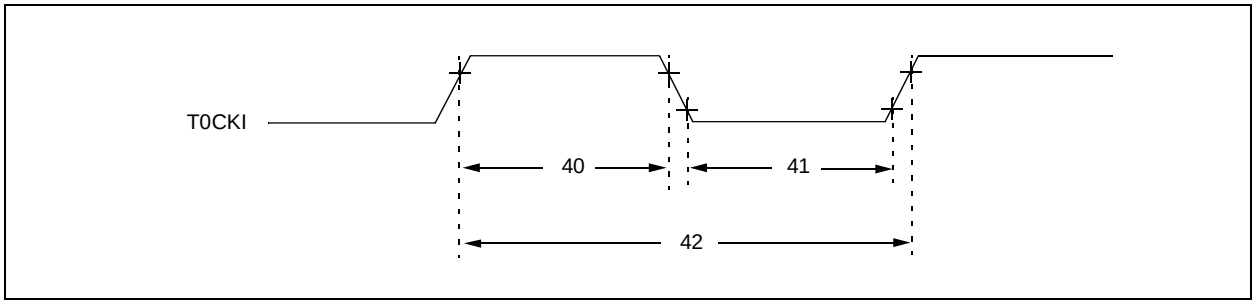


TABLE 15-10: TIMER0 CLOCK REQUIREMENTS

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

* These parameters are characterized but not tested.

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

16.0 DC / AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g., outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

FIGURE 16-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

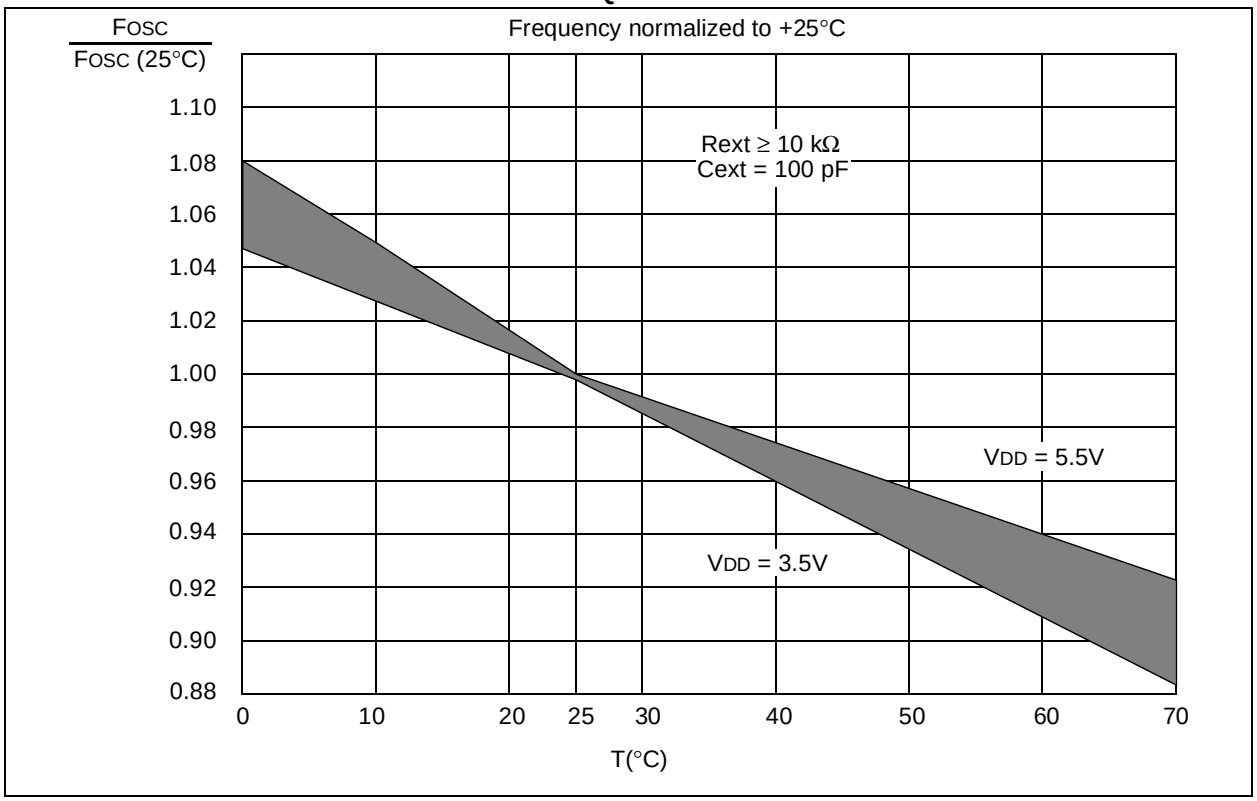


TABLE 16-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
20 pF	3.3k	4.973 MHz	± 27%
	5k	3.82 MHz	± 21%
	10k	2.22 MHz	± 21%
	100k	262.15 kHz	± 31%
100 pF	3.3k	1.63 MHz	± 13%
	5k	1.19 MHz	± 13%
	10k	684.64 kHz	± 18%
	100k	71.56 kHz	± 25%
300 pF	3.3k	660.0 kHz	± 10%
	5.k	484.1 kHz	± 14%
	10k	267.63 kHz	± 15%
	160k	29.44 kHz	± 19%

Note 1: The frequencies are measured on DIP packages.

Note 2: The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5V.

FIGURE 16-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

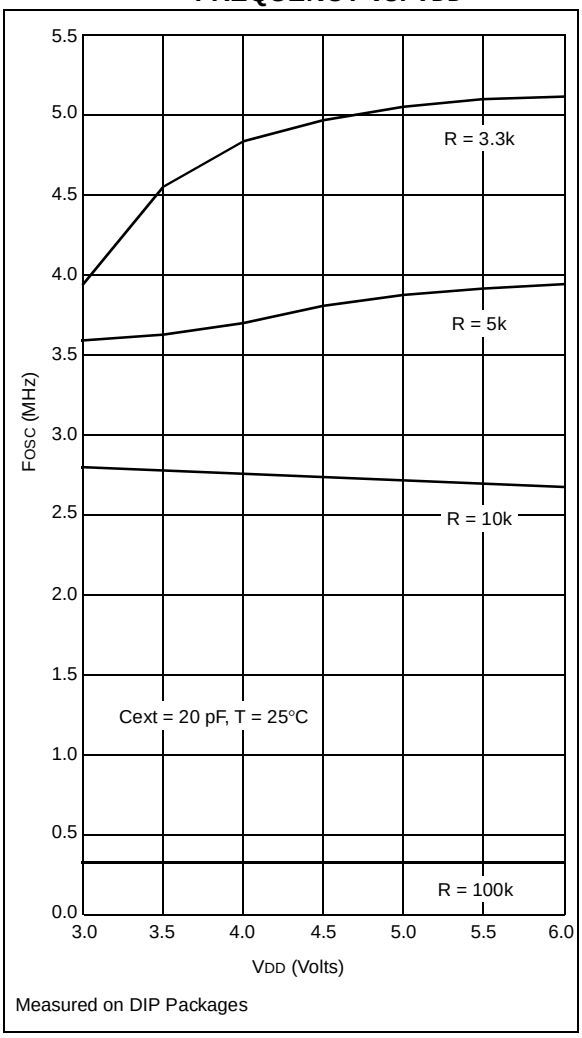


FIGURE 16-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

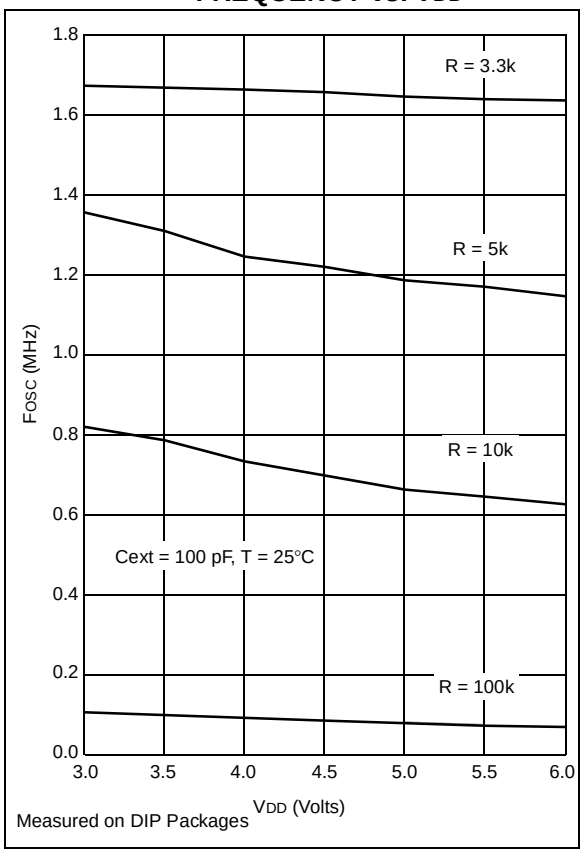


FIGURE 16-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

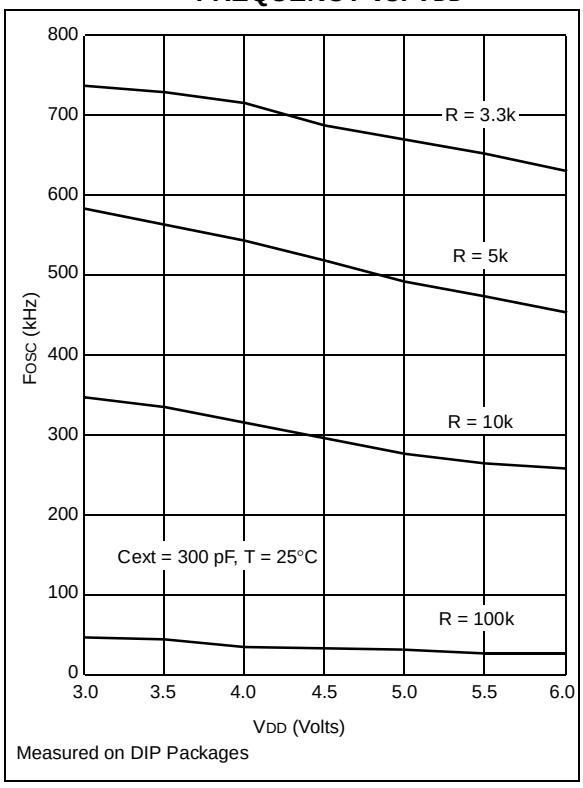


FIGURE 16-5: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

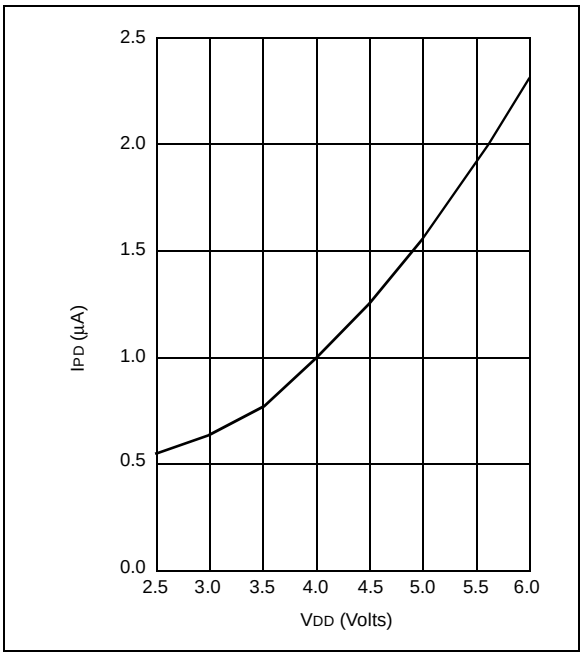


FIGURE 16-6: MAXIMUM IPD vs. VDD WATCHDOG DISABLED

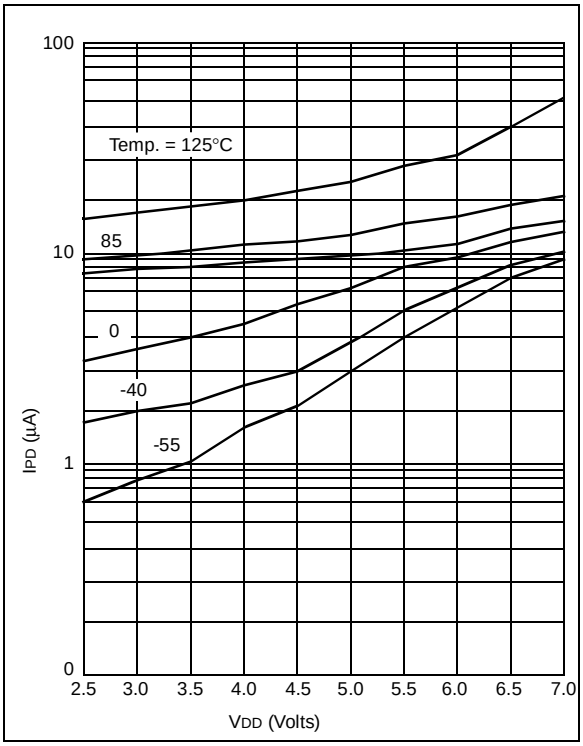


FIGURE 16-7: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

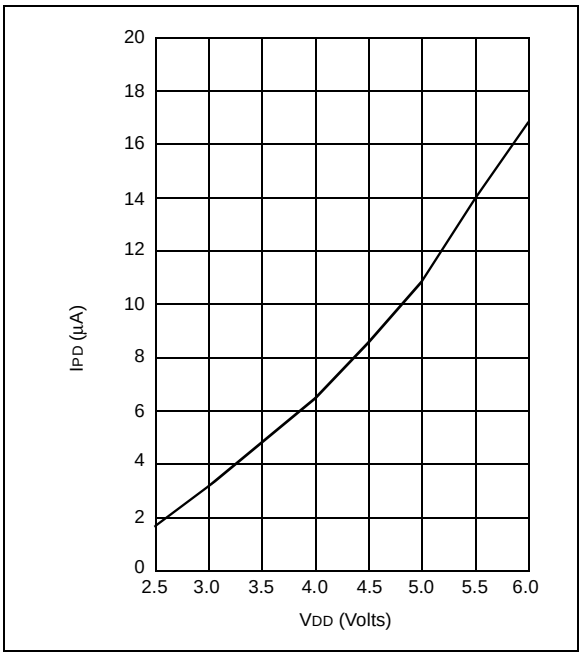
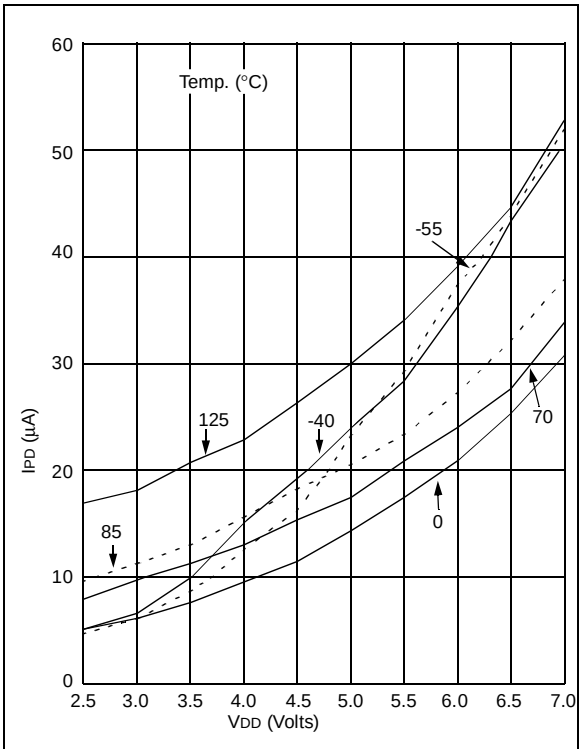


FIGURE 16-8: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



IPD, with WDT enabled, has two components:
The leakage current which increases with higher temperature and the operating current of the WDT logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 16-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD

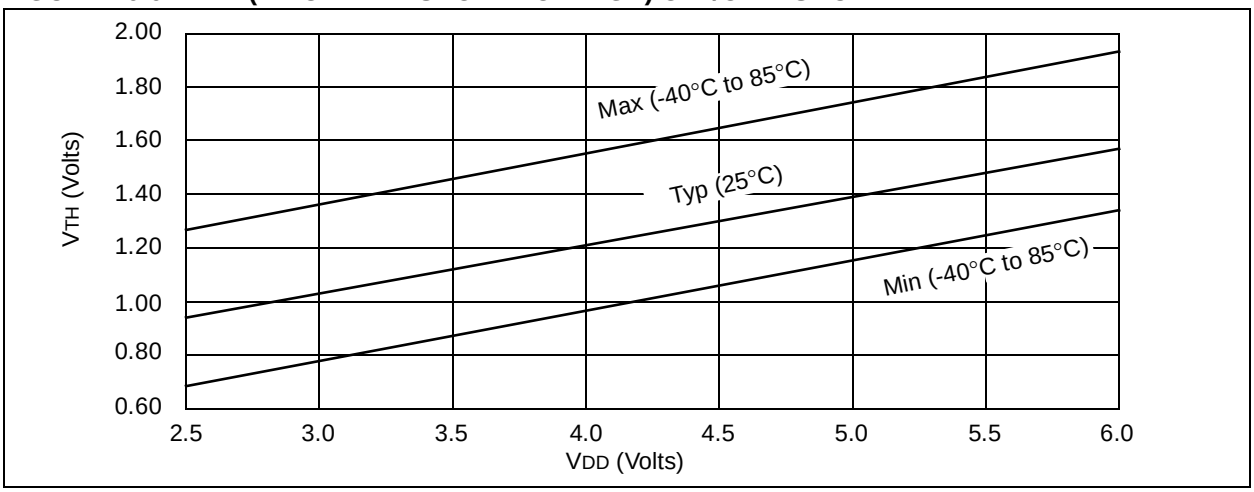
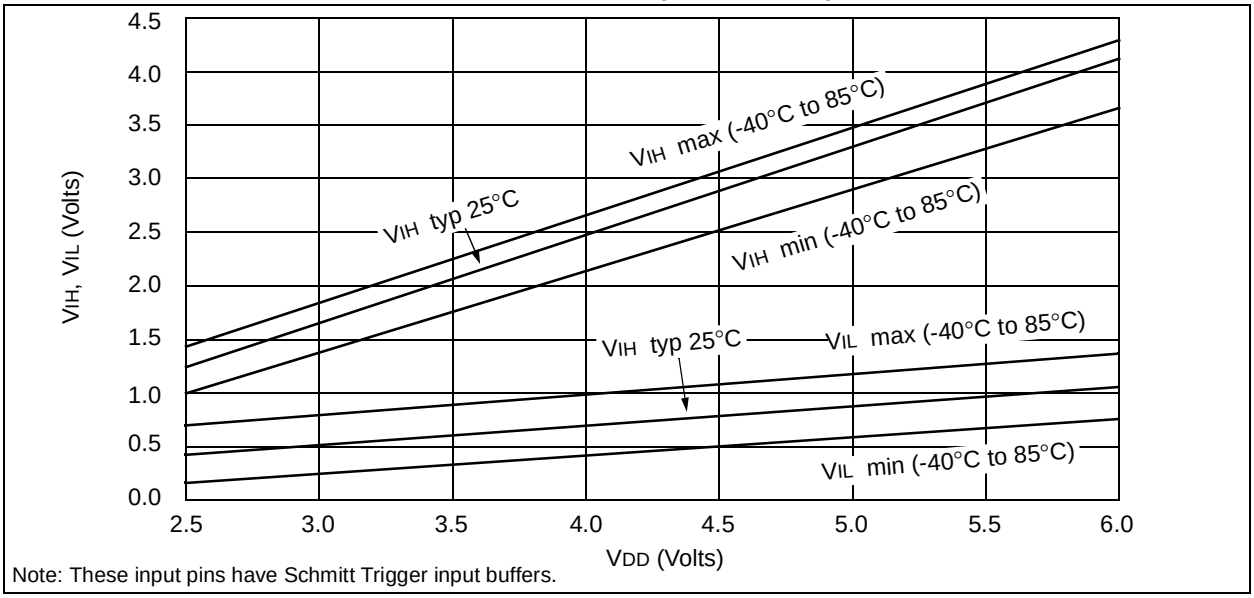


FIGURE 16-10: VIH, VIL OF $\overline{\text{MCLR}}$, T0CKI AND OSC1 (IN RC MODE) vs. VDD



Note: These input pins have Schmitt Trigger input buffers.

FIGURE 16-11: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD

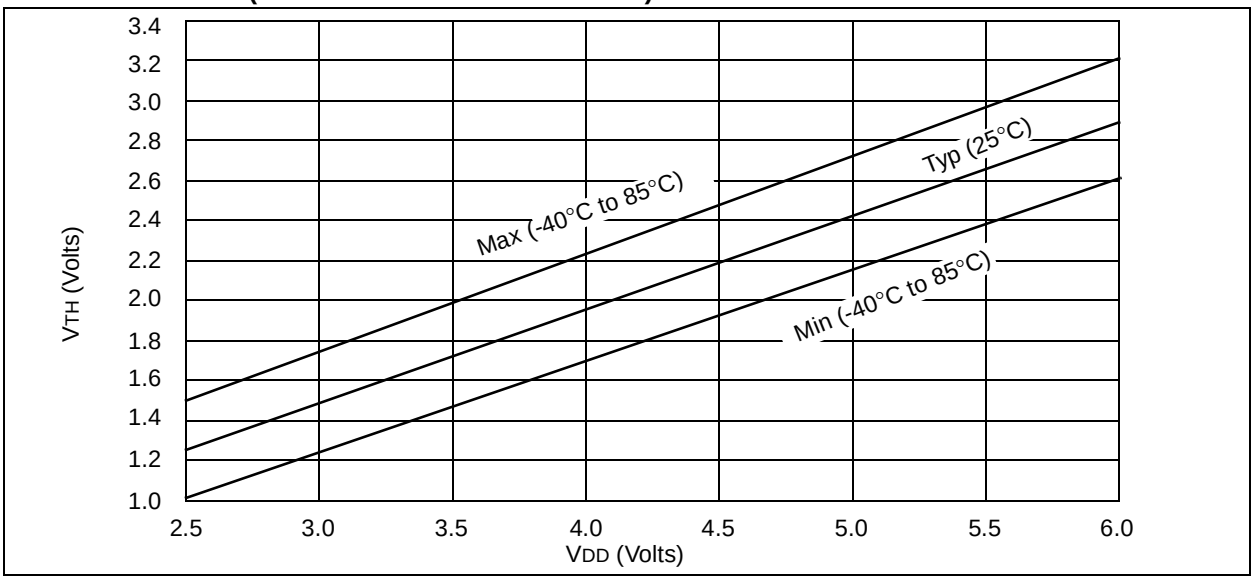


FIGURE 16-12: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

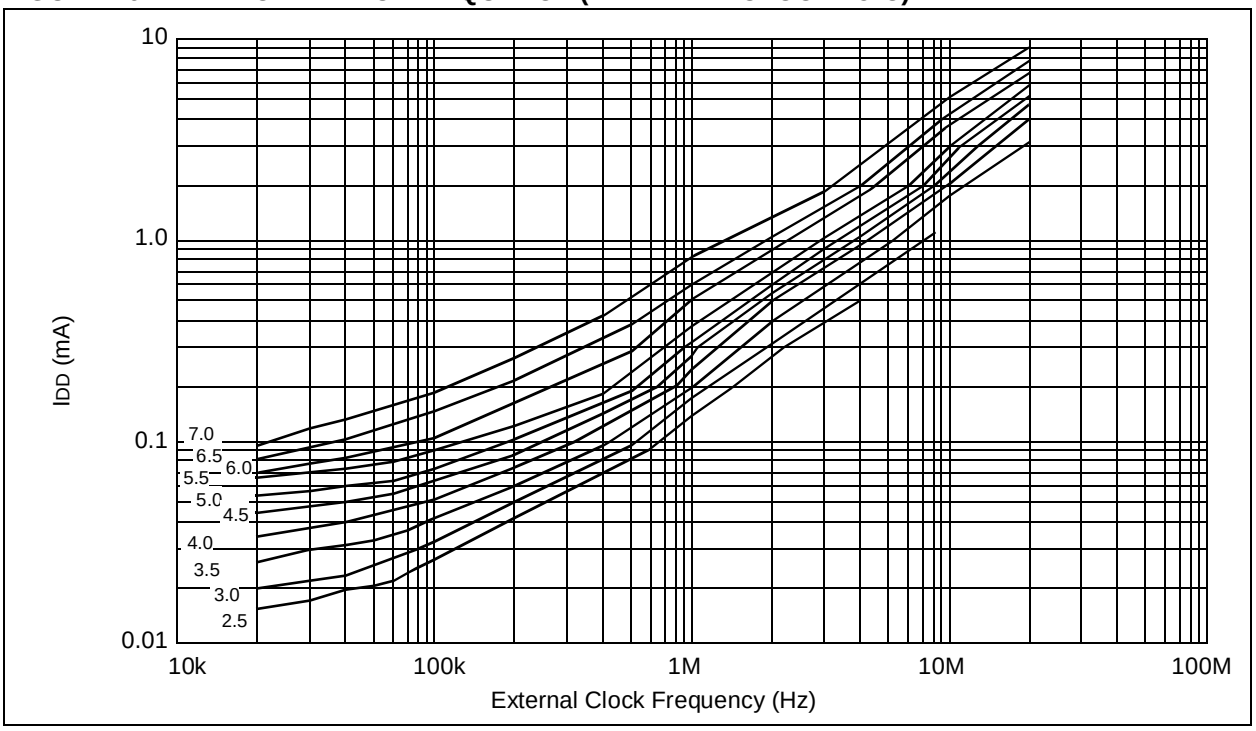


FIGURE 16-13: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK -40°C TO +85°C)

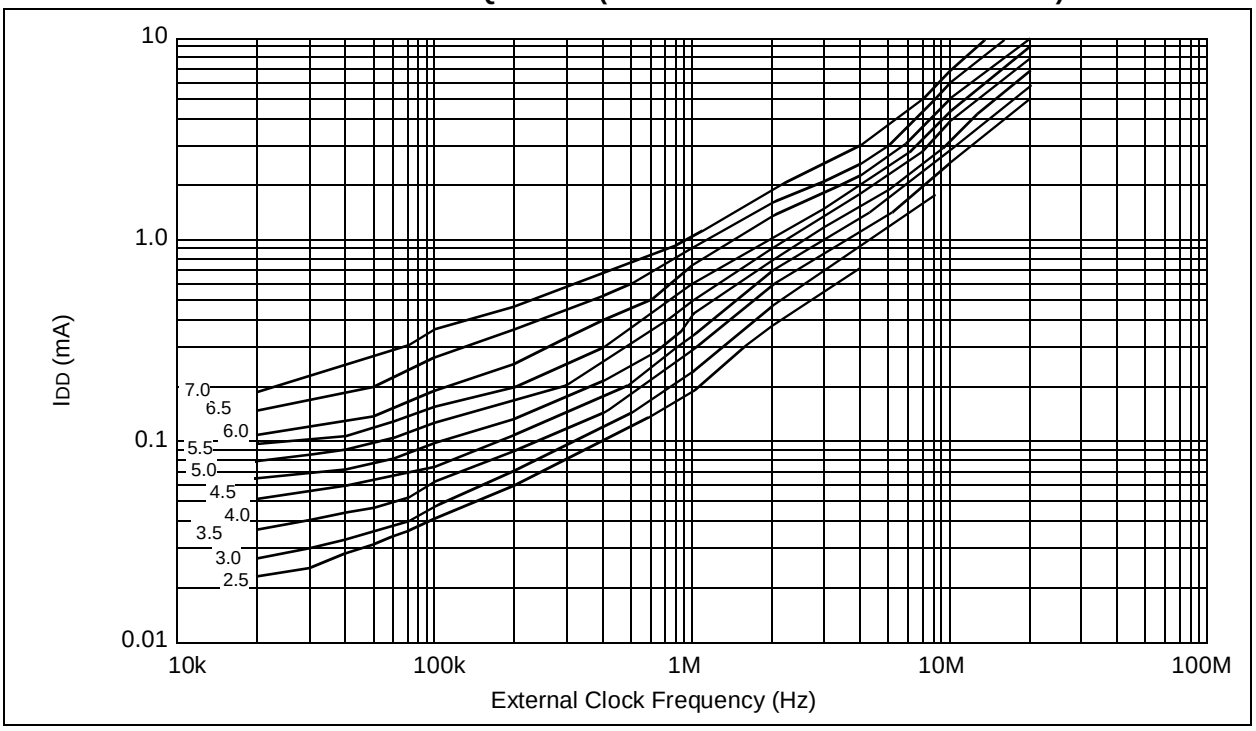


FIGURE 16-14: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK -55°C TO +125°C)

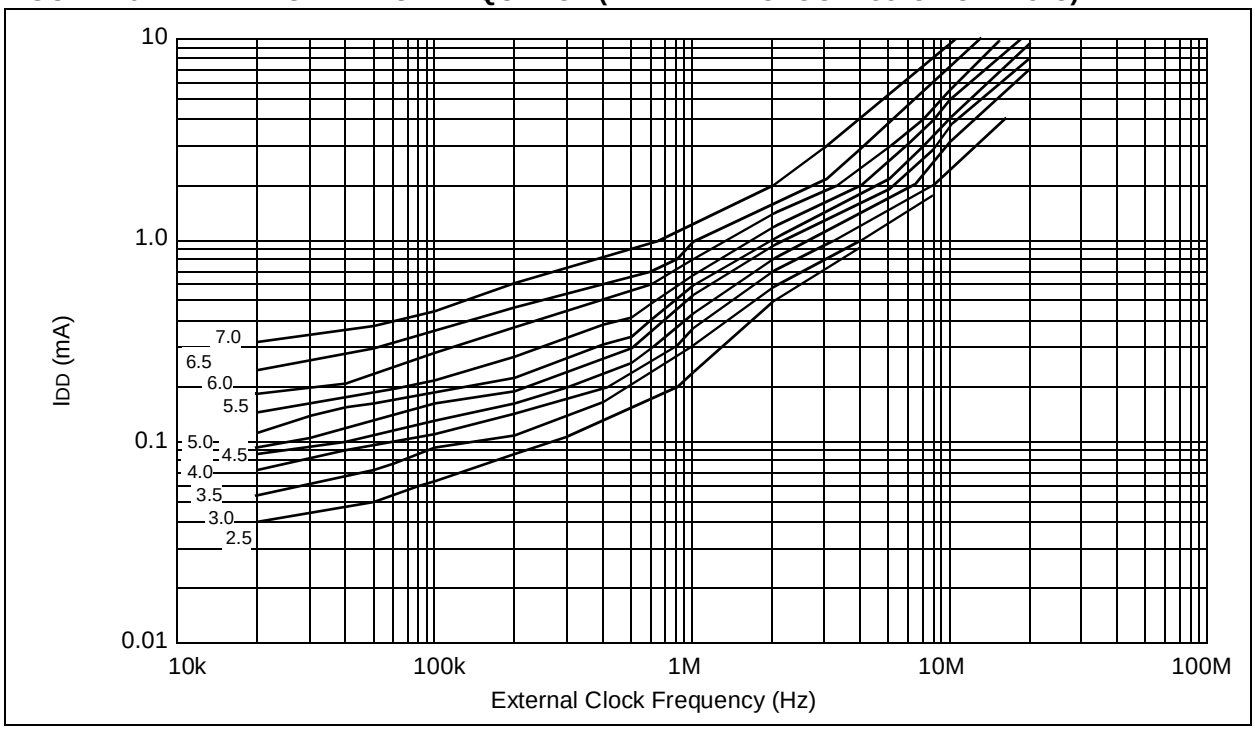


FIGURE 16-15: WDT TIMER TIME-OUT PERIOD vs. VDD

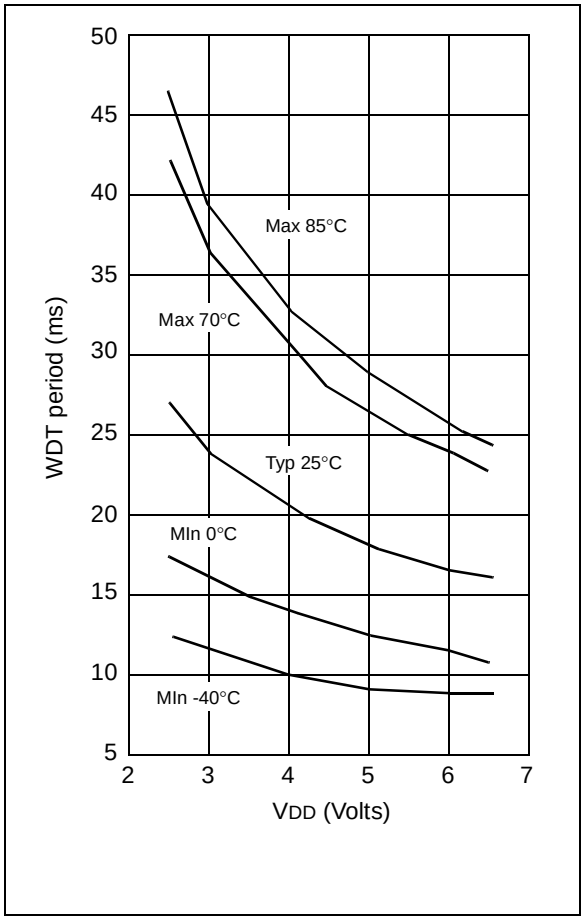


FIGURE 16-16: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. VDD

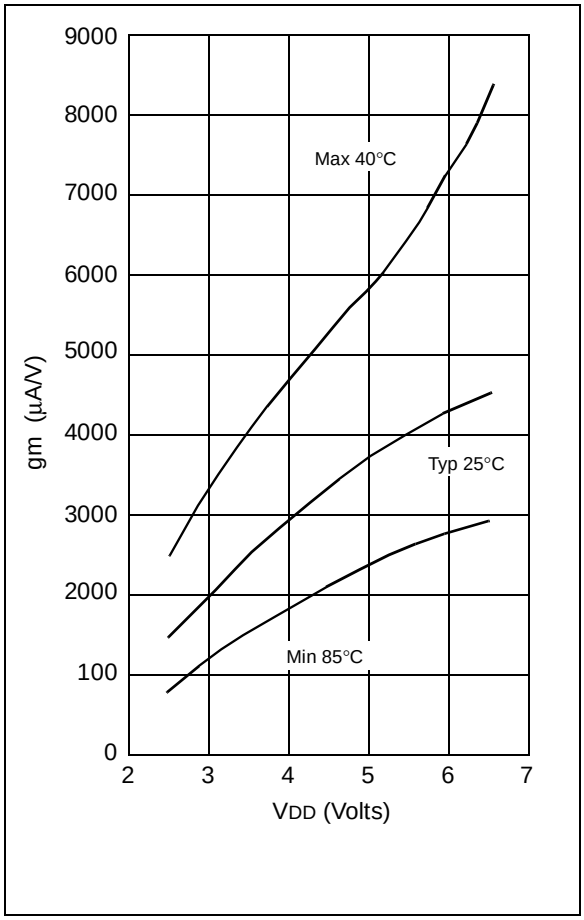


FIGURE 16-17: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR vs. VDD

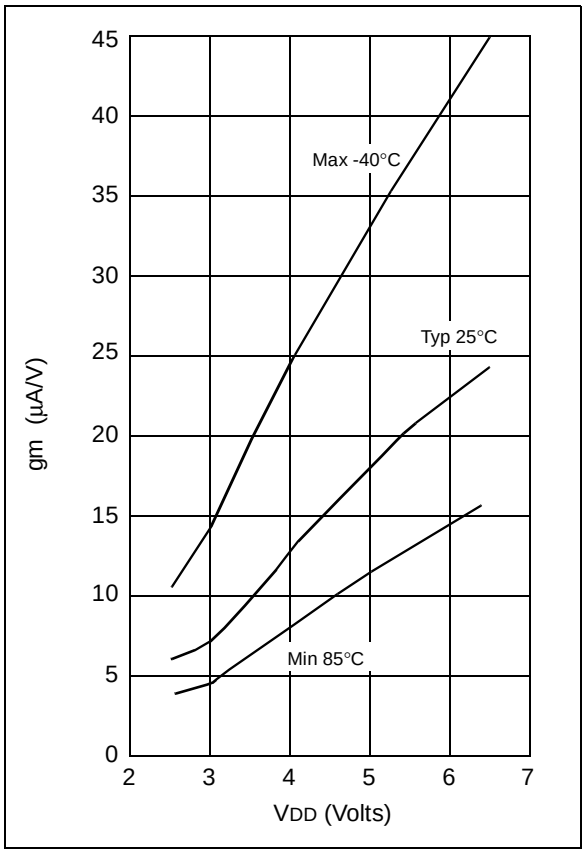


FIGURE 16-18: IOH vs. VOH, VDD = 3V

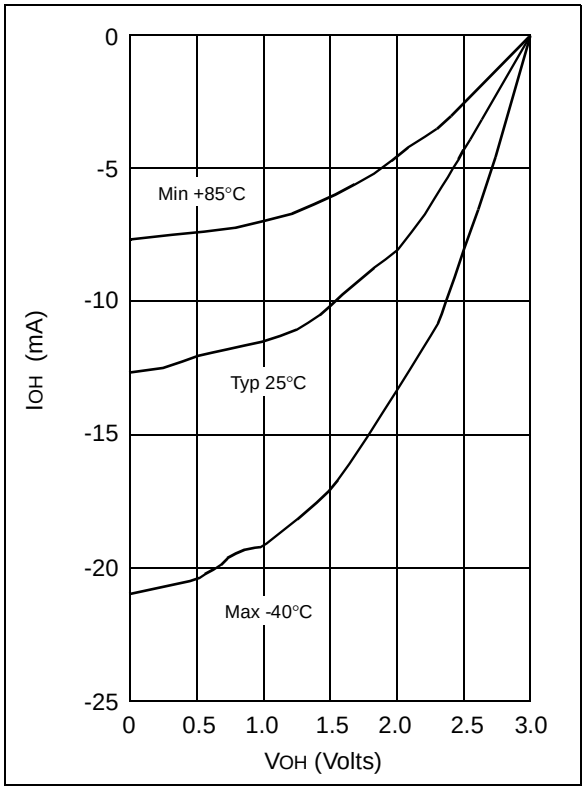


FIGURE 16-19: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

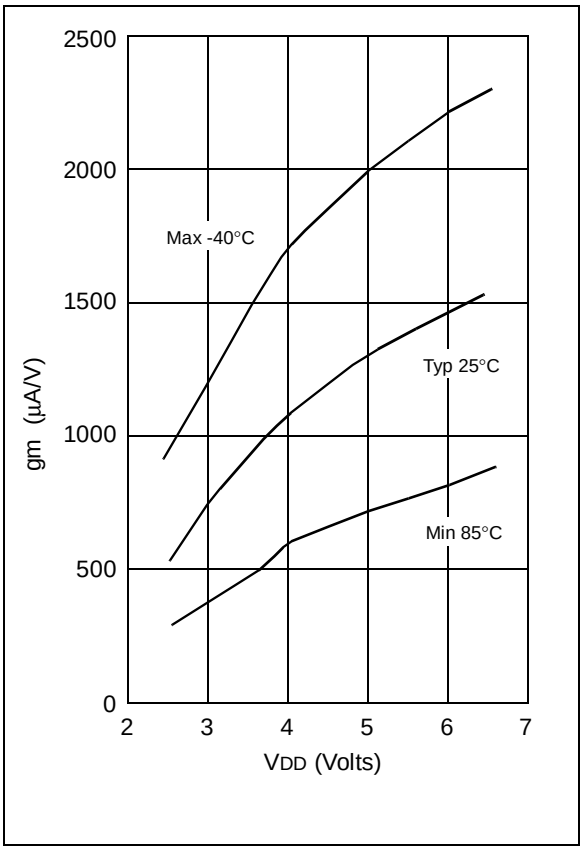


FIGURE 16-20: IOH vs. VOH, VDD = 5V

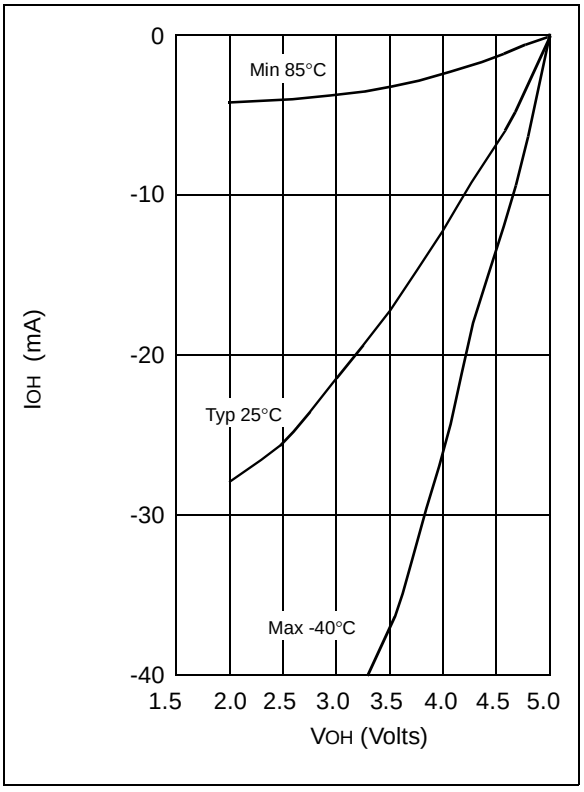


FIGURE 16-21: I_{OL} vs. V_{OL}, V_{DD} = 3V

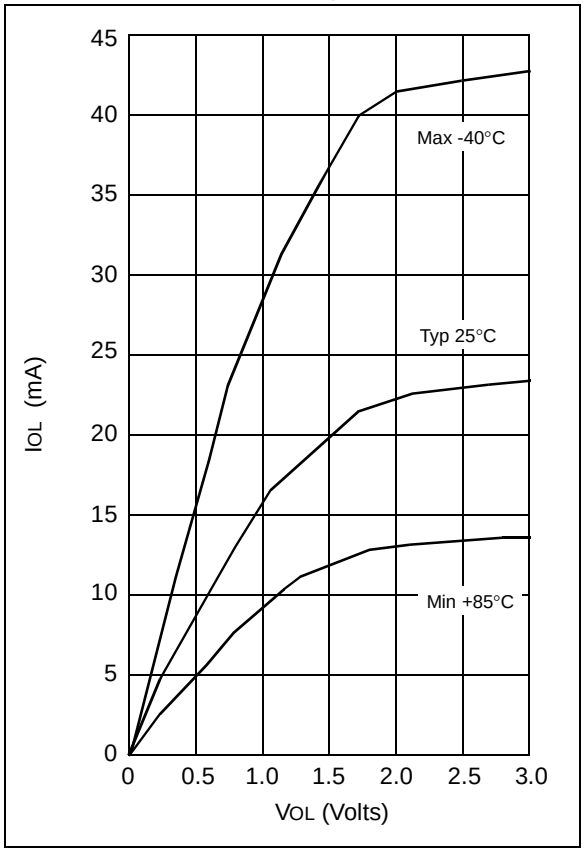
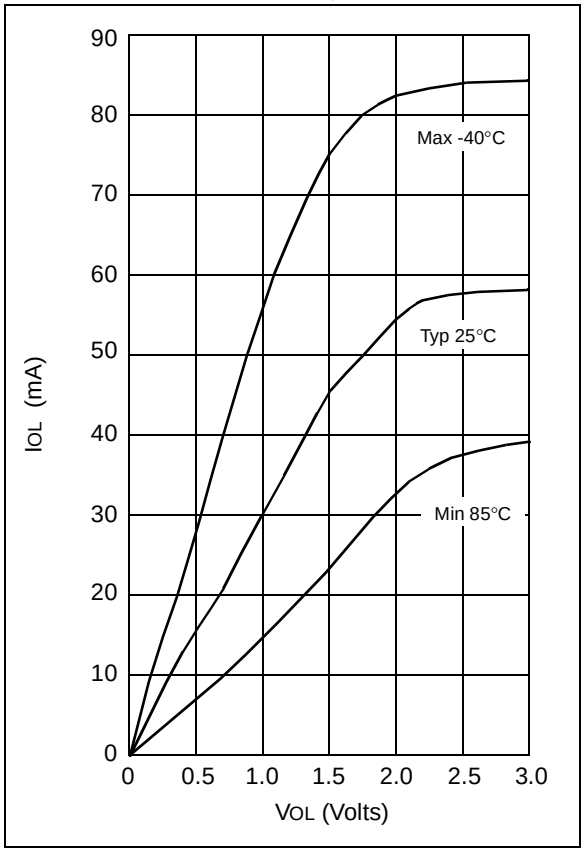


FIGURE 16-22: I_{OL} vs. V_{OL}, V_{DD} = 5V



17.0 EEPROM BUS DESCRIPTION

The MTA85XXX supports a bidirectional two wire bus and data transmission protocol. A device that sends data onto the bus is defined as transmitter, and a device receiving data as receiver. The bus has to be controlled by a master device (microcontroller) which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions, while the EEPROM (24LC01B/02B) works as slave. Both master and slave can operate as transmitter or receiver but the master device determines which mode is activated.

17.1 Bus Characteristics

The following **bus protocol** has been defined:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted as a START or STOP condition.

Accordingly, the following bus conditions have been defined (Figure 17-1):

17.1.1 BUS NOT BUSY (A)

Both data and clock lines remain HIGH.

17.1.2 START DATA TRANSFER (B)

A HIGH to LOW transition of the SDA line while the clock (SCL) is HIGH determines a START condition. All commands must be preceded by a START condition.

17.1.3 STOP DATA TRANSFER (C)

A LOW to HIGH transition of the SDA line while the clock (SCL) is HIGH determines a STOP condition. All operations must be ended with a STOP condition.

17.1.4 DATA VALID (D)

The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal.

The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of the data bytes transferred between the START and STOP conditions is determined by the master device and is theoretically unlimited, although only the last sixteen will be stored when doing a write operation. When an overwrite does occur it will replace data in a first in first out fashion.

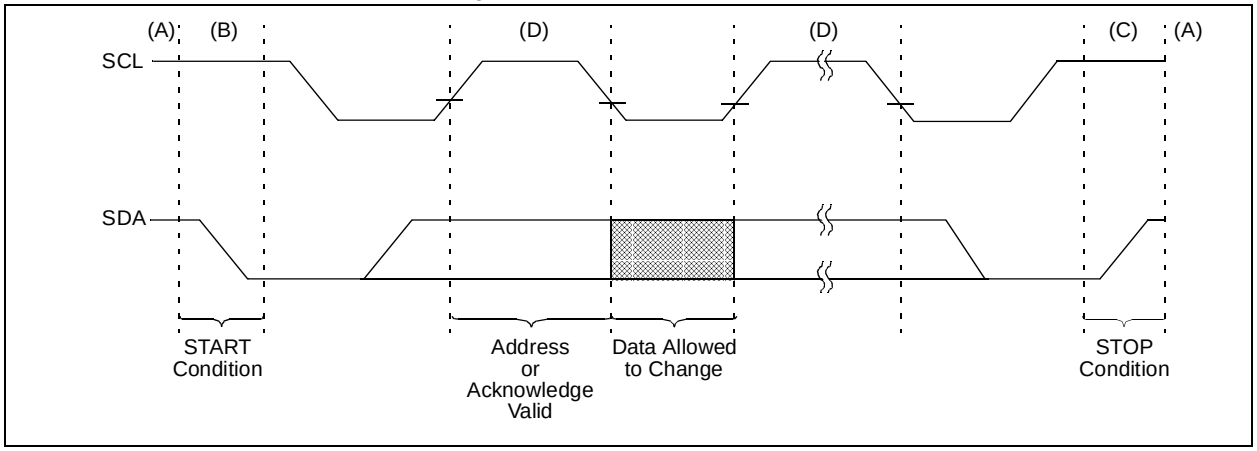
17.1.5 ACKNOWLEDGE

Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

*** Note: The EEPROM does not generate any acknowledge bits if an internal programming cycle is in progress.**

The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

FIGURE 17-1: DATA TRANSFER SEQUENCE ON THE SERIAL BUS



17.1.6 SLAVE ADDRESS

The 24LC01B/02B (EEPROM) are software-compatible with older devices such as the 24C01A, 24C02A, 24LC01, and the 24LC02. A single 24LC02B can be used in place of two 24LC01's, for example, without any modifications to software. The "chip select" portion of the control byte is a don't care.

After generating a START condition, the bus master transmits the slave address consisting of a 4-bit device code (1010) for the EEPROM, followed by three don't care bits.

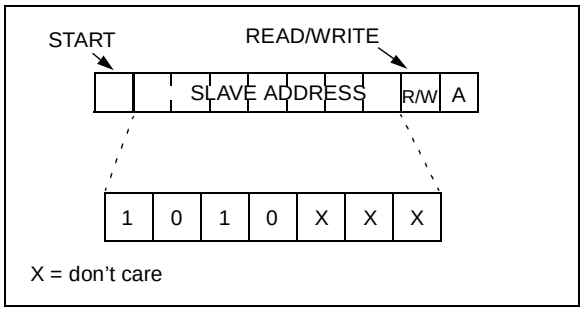
The eighth bit of slave address determines if the master device wants to read or write to the EEPROM (Figure 17-3).

The EEPROM monitors the bus for its corresponding slave address all the time. It generates an acknowledge bit if the slave address was true and it is not in a programming mode.

FIGURE 17-2: EEPROM CONTROL CODES

Operation	Control Code	Chip Select	R/$\overline{W}$
Read	1010	xxx	1
Write	1010	xxx	0

FIGURE 17-3: CONTROL BYTE ALLOCATION



18.0 WRITE OPERATION

18.1 Byte Write

Following the START signal from the master, the device code (4-bits), the don't care bits (3-bits), and the R/$\overline{W}$ bit which is a logic low is placed onto the bus by the master transmitter. This indicates to the addressed slave receiver that a byte with a word address will follow after it has generated an acknowledge bit during the ninth clock cycle. Therefore the next byte transmitted by the master is the word address and will be written into the address pointer of the EEPROM. After receiving another acknowledge signal from the EEPROM the master device will transmit the data word to be written into the addressed memory location. The EEPROM acknowledges again and the master generates a stop condition. This initiates the internal write cycle, and during this time the EEPROM will not generate acknowledge signals (Figure 18-1).

18.2 Page Write

The write control byte, word address and the first data byte are transmitted to the EEPROM in the same way as in a byte write. But instead of generating a STOP condition the master transmits up to eight data bytes to the EEPROM which are temporarily stored in the on-chip page buffer and will be written into the memory after the master has transmitted a STOP condition. After the receipt of each word, the three lower order address pointer bits are internally incremented by one. The higher order five bits of the word address remains constant. If the master should transmit more than eight words prior to generating the STOP condition, the address counter will roll over and the previously received data will be overwritten. As with the byte write operation, once the STOP condition is received an internal write cycle will begin (Figure 18-2).

FIGURE 18-1: BYTE WRITE

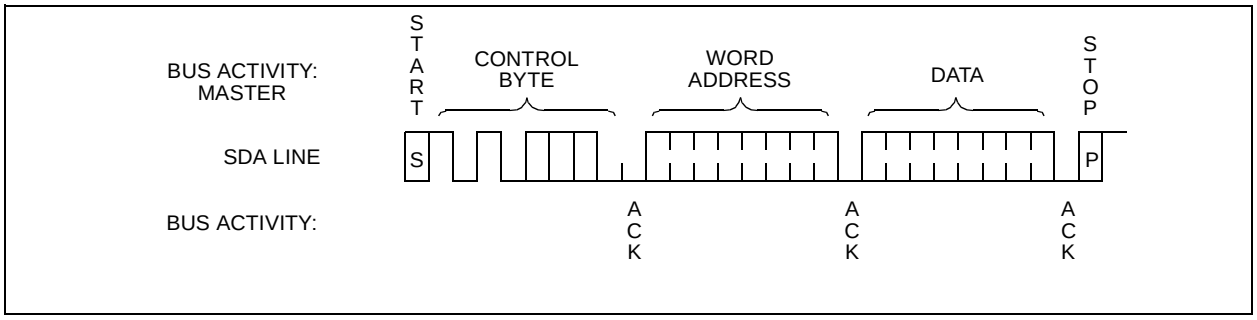
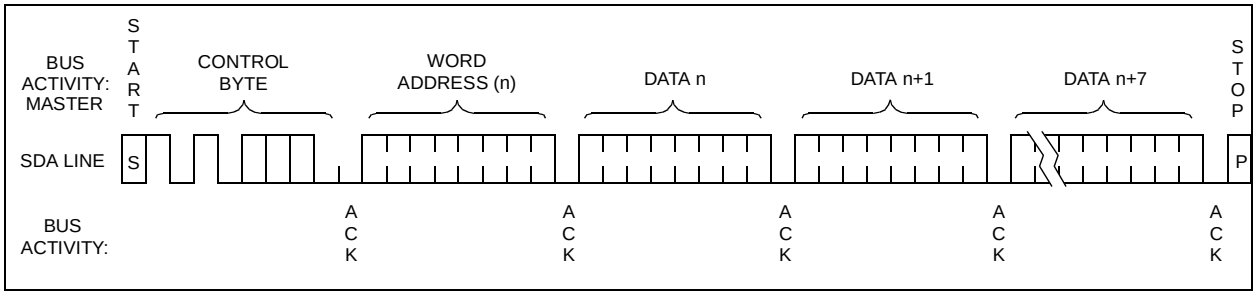


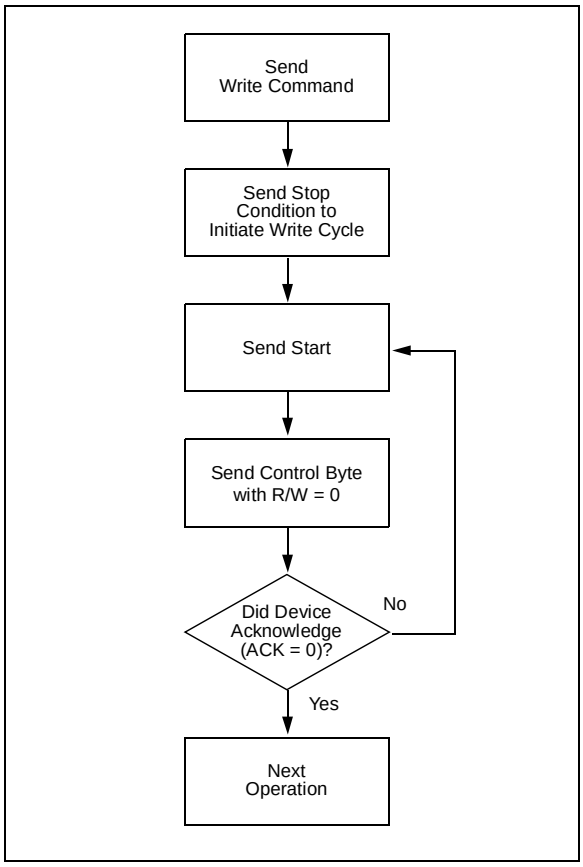
FIGURE 18-2: PAGE WRITE



18.3 Acknowledge Polling

Since the device will not acknowledge during a write cycle, this can be used to determine when the cycle is complete (this feature can be used to maximize bus throughput). Once the STOP condition for a write command has been issued from the master, the device initiates the internally timed write cycle. ACK polling can be initiated immediately. This involves the master sending a START condition followed by the control byte for a write command (R/$\overline{W}$ = 0). If the device is still busy with the write cycle, then no ACK will be returned. If the cycle is complete, then the device will return the ACK and the master can then proceed with the next read or write command. See Figure 18-3 for flow diagram.

FIGURE 18-3: ACKNOWLEDGE POLLING FLOW



19.0 READ OPERATION

Read operations are initiated in the same way as write operations with the exception that the R/$\overline{W}$ bit of the slave address is set to. There are three basic types of read operations:

- Current address read
- Random read
- Sequential read.

19.1 Current Address Read

The EEPROM contains an address counter that maintains the address of the last word accessed, internally incremented by one. Therefore, if the previous access (either a read or write operation) was to address "n", the next current address read operation would access data from address "n + 1". Upon receipt of the slave address with R/$\overline{W}$ bit set, the EEPROM issues an acknowledge and transmits the eight-bit data word. The master will not acknowledge the transfer but does generate a STOP condition and the EEPROM discontinues transmission (Figure 19-1). If a current address read is performed after a Power-Up, the last address will be read.

19.2 Random Read

Random read operations allow the master to access any memory location in a random manner. To perform this type of read operation, first the word address must be set. This is done by sending the word address to the EEPROM as part of a write operation. After the word address is sent, the master generates a START condition following the acknowledge. This terminates the write operation, but not before the internal address pointer is set. Then the master issues the control byte again but with the R/$\overline{W}$ bit set to a '1'. The EEPROM will then issue an acknowledge and transmits the eight-bit data word. The master will not acknowledge the transfer but does generate a STOP condition and the EEPROM discontinues transmission (Figure 19-2).

19.3 Sequential Read

Sequential reads are initiated in the same way as a random read except that after the EEPROM transmits the first data byte, the master issues an acknowledge as opposed to a STOP condition in a random read. This directs the EEPROM to transmit the next sequentially addressed 8-bit word (Figure 19-3).

FIGURE 19-1: CURRENT ADDRESS READ

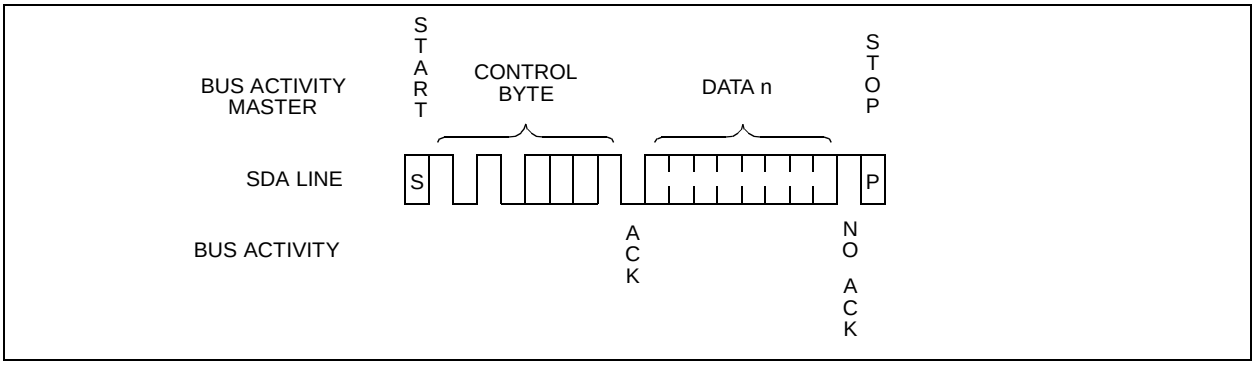


FIGURE 19-2: RANDOM READ

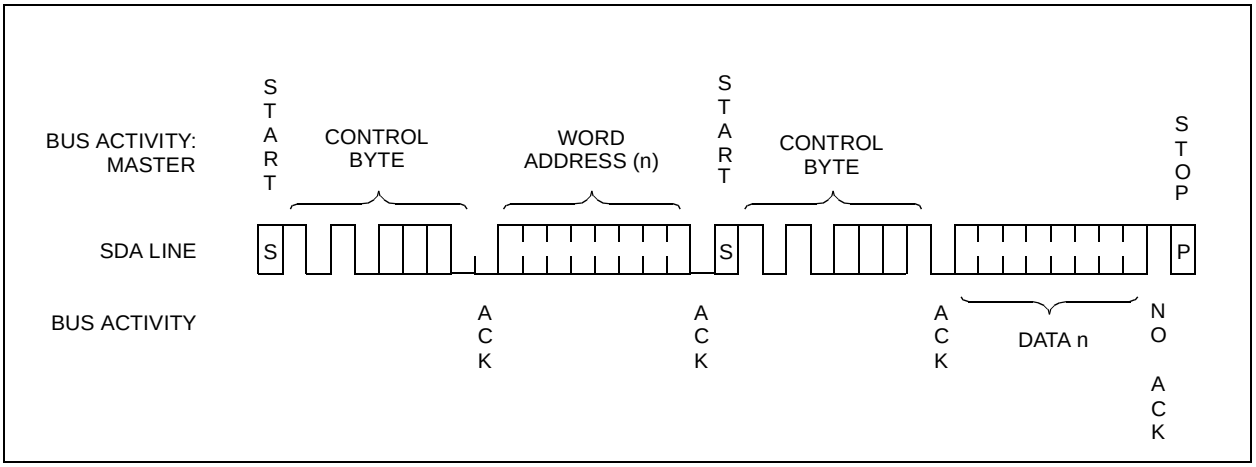
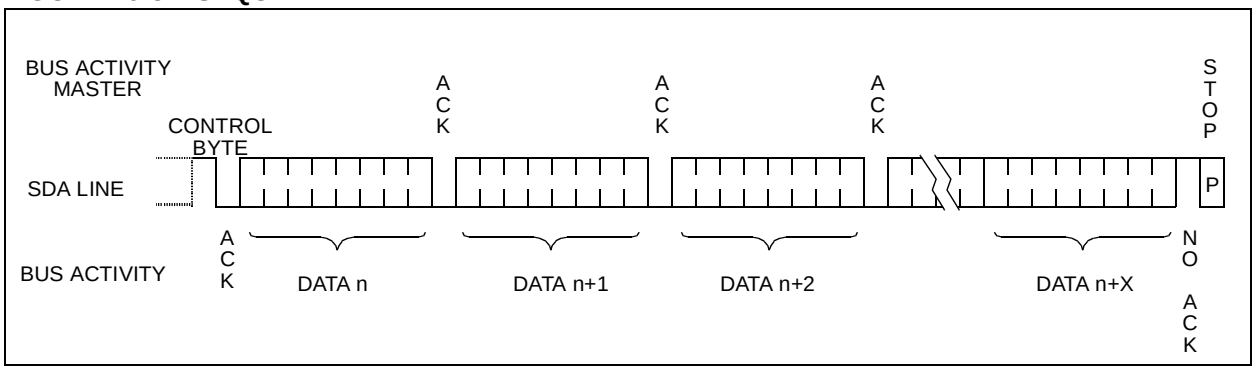


FIGURE 19-3: SEQUENTIAL READ



To provide sequential reads the EEPROM contains an internal address pointer which is incremented by one at the completion of each operation. This address pointer allows the entire memory contents to be serially read during one operation.

20.0 GENERAL EEPROM INFORMATION

20.1 Noise Protection

The EEPROM employs a VCC threshold detector circuit which disables the internal erase/write logic if the VCC is below 1.5V at nominal conditions.

The SCL and SDA inputs have Schmitt trigger and filter circuits which suppress noise spikes to assure proper device operation even on a noisy bus.

20.2 SDA Serial Address/Data Input/ Output

This is a bidirectional pin used to transfer addresses and data into and data out of the device. It is an open drain terminal, therefore the SDA bus requires a pullup resistor to VCC (typical 10 kΩ for 100 kHz and 1 kΩ for 400 kHz).

For normal data transfer SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

20.3 SCL Serial Clock

This input is used to synchronize the data transfer from and to the device.

21.0 DEVELOPMENT SUPPORT

21.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER™ Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART™ Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

21.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 21-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 (and better) machines in the Microsoft Windows® 3.x environment. Thus, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC-Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host Emulation Control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x, two or more PICMASTER emulators can be run simultaneously from the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes specifications are shown in Table 21-1.

FIGURE 21-1: PICMASTER SYSTEM CONFIGURATION

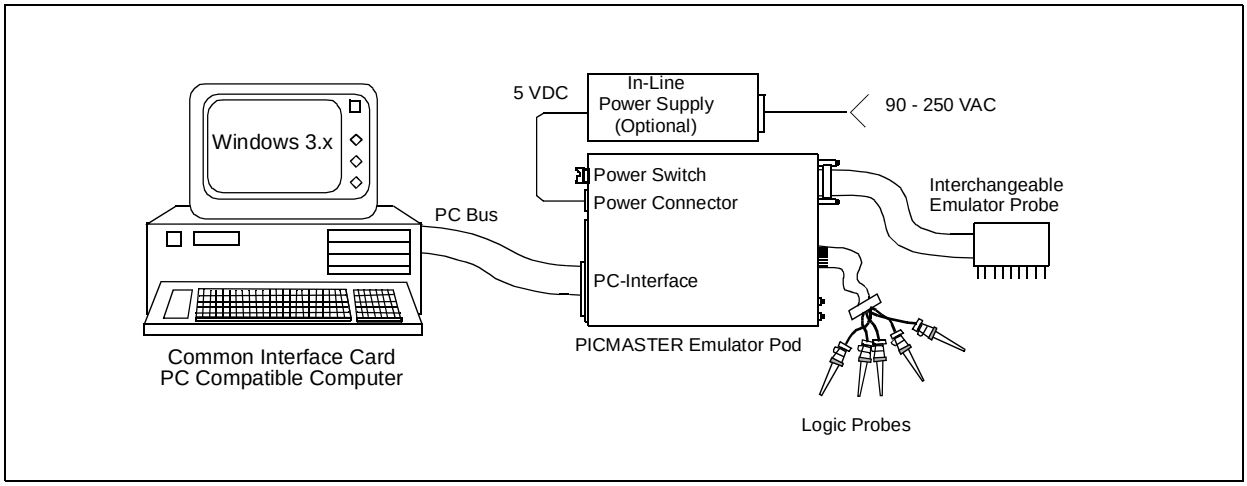


TABLE 21-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

21.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable V_{DD} and V_{PP} supplies which allows it to verify programmed memory at V_{DD} min and V_{DD} max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of V_{DD} min, V_{DD} max and V_{PP} levels, load and store to and from disk files (hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular “programming socket module”. Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

21.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

21.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

21.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C63, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

21.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically (i.e., by meaningful names).
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and sub-titles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

21.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

21.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

21.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP Edition, for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

21.11 Development Systems

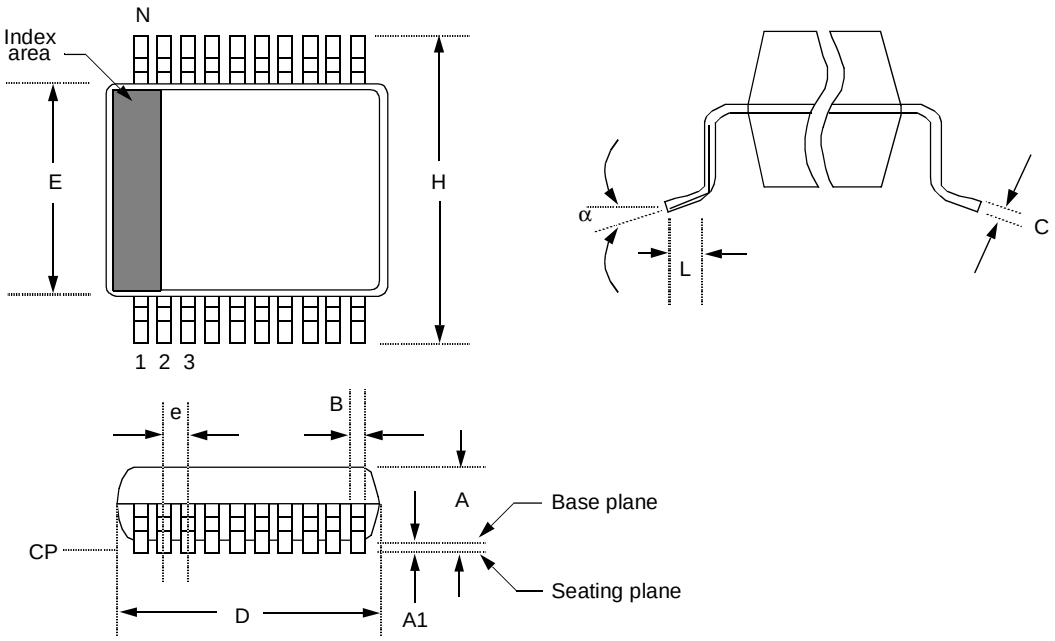
For convenience, the development tools are packaged into comprehensive systems as listed in Table 21-2.

TABLE 21-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator, PRO MATE Programmer, Assembler, Software Simulator, Samples and your choice of Target Probe.
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator
4.	PICSEE-85A Introduction Design Kit	Kit contains the programming adaptor (item 3), a PICMASTER interface board and microcontroller samples of the MTA85XXX component devices
5.	PICSEESTART-85A Development Kit	Kit contains a PICSTART System (item 2) and PICSEE-85A Introduction Design Lit (item 4)
6.	PICSEE-85A PRO MATE Header	SSOP PRO MATE header for the MTA85XXX products. No programming adaptor needed

22.0 PACKAGING DIAGRAMS AND DIMENSIONS

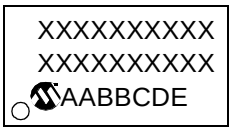
22.1 20-Lead Plastic Surface Mount (SSOP - 209 mil Body 5.30mm)



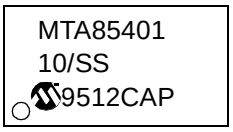
Package Group: Plastic SSOP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	1.730	1.990		0.068	0.078	
A1	0.050	0.210		0.002	0.008	
B	0.250	0.380		0.010	0.015	
C	0.130	0.220		0.005	0.009	
D	7.070	7.330		0.278	0.289	
E	5.200	5.380		0.205	0.212	
e	0.650	0.650	**Reference**	0.026	0.026	**Reference**
H	7.650	7.900		0.301	0.311	
L	0.550	0.950		0.022	0.037	
N	20	20		20	20	
CP	-	0.102		-	0.004	

23.0 PACKAGE MARKING INFORMATION

20-Lead SSOP



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	D_2	Mask revision number for EEPROM
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

INDEX

Table of Figures

Table of Tables

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS**.

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

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 area code (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: MTA85XXX Literature Number: DS40115C

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?

NOTES:

NOTES:

NOTES:

MTA85XXX Product Identification System

To order or to obtain information (e.g., on pricing or delivery), please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)		
Package:	SS	=	209 mill SSOP
Temperature Range:	I	=	-40°C to +85°C (S for tape/reel)
Frequency Range:	04	=	4 MHz (MTA854XX and 858XX)
	10	=	10 MHz (MTA858XX only)
Device:	MTA85401 :Refer to Table 1-1		

Examples:

a) MTA85402 - 04/SS 301 = Commercial temperature, SSOP package, 4 MHz, QTP pattern #301

b) MTA85811 - 10I/SS = Industrial temperature, SSOP package, 10 MHz

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 404 640-0034 Fax: 404 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued)

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology Inc.
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology Taiwan
Singapore Branch
200 Middle Road
#10-03 Prime Centre
Singapore 0718
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire
SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

MICROCHIP

Printed in the USA, 8/95