Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PIC16C55X

EPROM-Based 8-Bit CMOS Microcontrollers

Devices Included in this Data Sheet:

Referred to collectively as PIC16C55X.

- PIC16C554
- PIC16C557
- PIC16C558

High Performance RISC CPU:

- Only 35 instructions to learn
- All single-cycle instructions (200 ns), except for program branches which are two-cycle
- Operating speed:
  - DC - 20 MHz clock input
  - DC - 20 ns instruction cycle

Device	Program Memory	Data Memory
PIC16C554	512	80
PIC16C557	2 K	128
PIC16C558	2 K	128

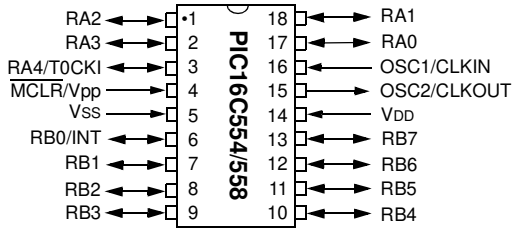
- Interrupt capability
- 16-18 special function hardware registers
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Peripheral Features:

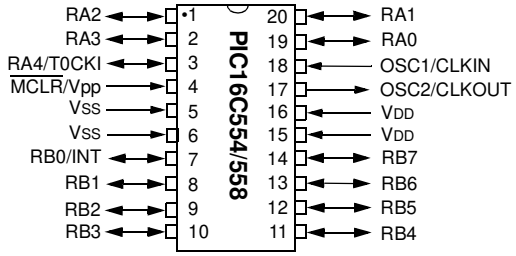
- 13-22 I/O pins with individual direction control
  - Pull-up resistors on PORTB
- High current sink/source for direct LED drive
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Diagram

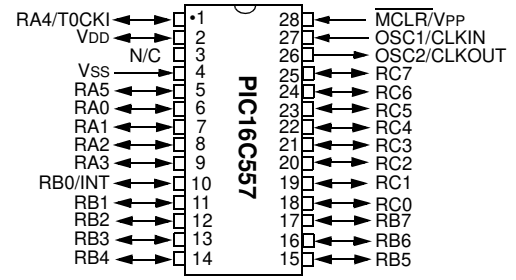
PDIP, SOIC, Windowed CERDIP



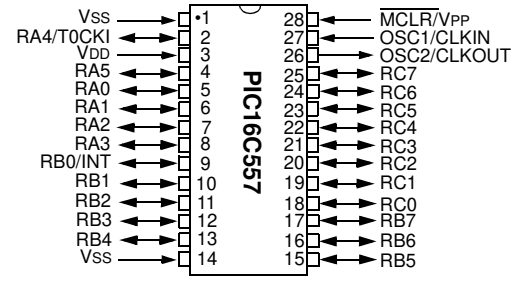
SSOP



PDIP, SOIC, Windowed CERDIP



SSOP



Special Microcontroller Features:

- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Serial in-circuit programming (via two pins)
- Four user programmable ID locations

Note: For additional information on enhancements, see Appendix A

CMOS Technology:

- Low power, high speed CMOS EPROM technology
- Fully static design
- Wide operating voltage range
  - 2.5V to 5.5V
- Commercial, Industrial and Extended temperature range
- Low power consumption
  - < 2.0 mA @ 5.0V, 4.0 MHz
  - 15 μA typical 3.0V, 32 kHz
  - < 1.0 μA typical standby current @ 3.0V

Device Differences

Device	Voltage Range	Oscillator
PIC16C554	2.5 - 5.5	**(Note 1)**
PIC16C557	2.5 - 5.5	**(Note 1)**
PIC16C558	2.5 - 5.5	**(Note 1)**

Note 1: If you change from this device to another device, please verify oscillator characteristics in your application.

Table of Contents

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

1.0 GENERAL DESCRIPTION

The PIC16C55X are 18, 20 and 28-Pin EPROM-based members of the versatile PIC16CXX family of low cost, high performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC® microcontrollers employ an advanced RISC architecture. The PIC16C55X have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single-cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16C55X microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC16C554 has 80 bytes of RAM. The PIC16C557 and PIC16C558 have 128 bytes of RAM. The PIC16C554 and PIC16C558 have 13 I/O pins and an 8-bit timer/counter with an 8-bit programmable prescaler. The PIC16C557 has 22 I/O pins and an 8-bit timer/counter with an 8-bit programmable prescaler.

PIC16C55X devices have special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for high speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake-up the chip from SLEEP through several external and internal interrupts and RESET.

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

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

Table 1-1 shows the features of the PIC16C55X mid-range microcontroller families.

A simplified block diagram of the PIC16C55X is shown in Figure 3-1.

The PIC16C55X series fit perfectly in applications ranging from motor control to low power remote sensors. The EPROM technology makes customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16C55X very versatile.

1.1 Family and Upward Compatibility

Users familiar with the family of microcontrollers will realize that this is an enhanced version of the architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for can be easily ported to PIC16C55X family of devices (Appendix B).

The PIC16C55X family fills the niche for users wanting to migrate up from the family and not needing various peripheral features of other members of the PIC16XX mid-range microcontroller family.

1.2 Development Support

The PIC16C55X family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low cost development programmer and a full-featured programmer.

TABLE 1-1: PIC16C55X FAMILY OF DEVICES

		PIC16C554	PIC16C557	PIC16C558
Clock	Maximum Frequency of Operation (MHz)	20	20	20
Memory	EPROM Program Memory (x14 words)	512	2K	2K
	Data Memory (bytes)	80	128	128
Peripherals	Timer Module(s)	TMR0	TMR0	TMR0
Features	Interrupt Sources	3	3	3
	I/O Pins	13	22	13
	Voltage Range (Volts)	2.5-5.5	2.5-5.5	2.5-5.5
	Brown-out Reset	—	—	—
	Packages	18-pin DIP, SOIC; 20-pin SSOP	28-pin DIP, SOIC; 28-pin SSOP	18-pin DIP, SOIC, SSOP
All PIC® Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C55X Family devices use serial programming with clock pin RB6 and data pin RB7.				

2.0 PIC16C55X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C55X Product Identification System section at the end of this data sheet. When placing orders, please use this page of the data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package, is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PICSTART® and PROMATE® programmers both support programming of the PIC16C55X.

2.2 One-Time Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium-to-high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices, but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined 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.

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C55X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C55X uses a Harvard architecture in which program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently from 8-bit wide data words. Instruction opcodes are 14-bit wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single-cycle (200 ns @ 20 MHz) except for program branches. The table below lists the memory (EPROM and RAM).

Device	Program Memory (EPROM)	Data Memor (RAM)
PIC16C554	512	80
PIC16C557	2 K	128
PIC16C558	2 K	128

The PIC16C554 addresses 512 x 14 on-chip program memory. The PIC16C557 and PIC16C558 addresses 2 K x 14 program memory. All program memory is internal.

The PIC16C55X can directly or indirectly address its register files or data memory. All special function registers, including the program counter, are mapped into the data memory. The PIC16C55X has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any Addressing mode. This symmetrical nature and lack of ‘special optimal situations’ make programming with the PIC16C55X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16C55X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

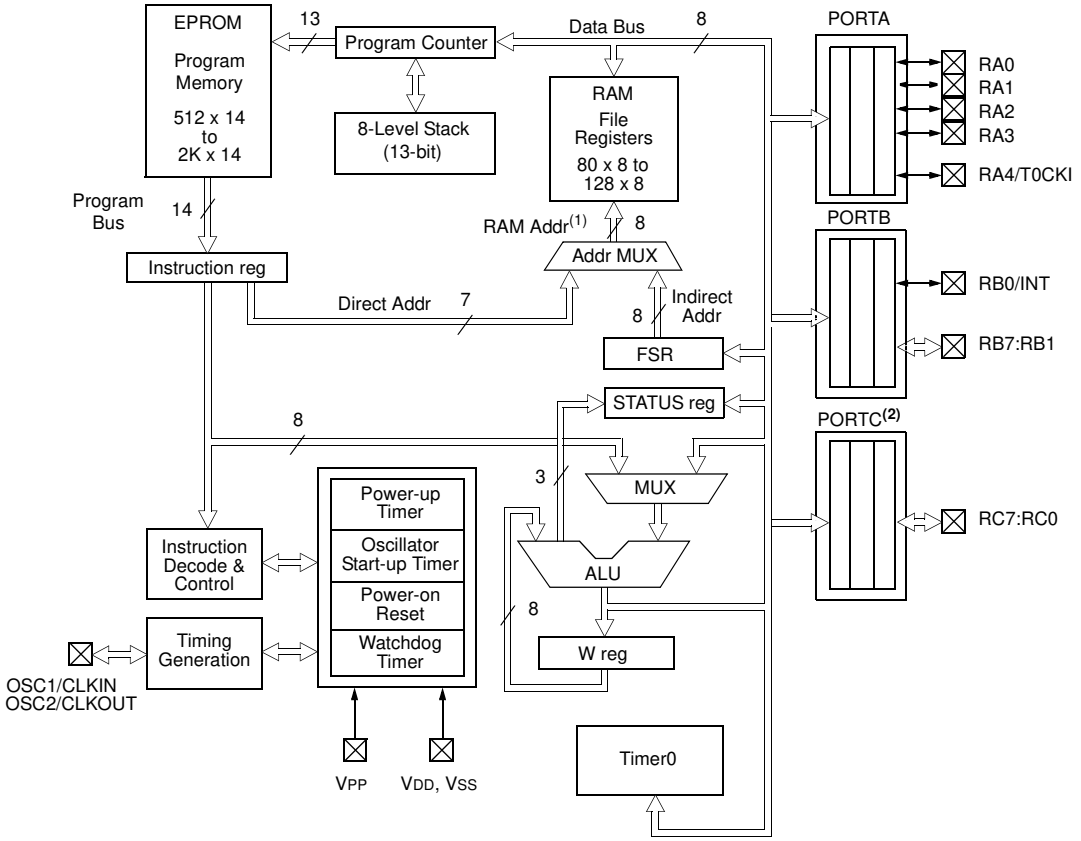
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram is shown in Figure 3-1, with a description of the device pins in Table 3-1.

FIGURE 3-1: BLOCK DIAGRAM

Device	Program Memory	Data Memory
PIC16C554	512 x 14	80 x 8
PIC16C557	2 K x 14	128 x 8
PIC16C558	2 K x 14	128 x 8



Note 1: Higher order bits are from STATUS Register.
2: PIC16C557 only.

TABLE 3-1: PIC16C55X PINOUT DESCRIPTION

Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	SOIC	SSOP			
OSC1/CLKIN	16	16	18	I	ST/CMOS	Oscillator crystal input/external clock source output.
OSC2/CLKOUT	15	15	17	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	4	4	4	I/P	ST	Master clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
RA0	17	17	19	I/O	ST	Bi-directional I/O port
RA1	18	18	20	I/O	ST	Bi-directional I/O port
RA2	1	1	1	I/O	ST	Bi-directional I/O port
RA3	2	2	2	I/O	ST	Bi-directional I/O port
RA4/T0CKI	3	3	3	I/O	ST	Bi-directional I/O port or external clock input for TMR0. Output is open drain type.
RB0/INT	6	6	7	I/O	TTL/ST(1)	Bi-directional I/O port can be software programmed for internal weak pull-up. RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up.
RB2	8	8	9	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up.
RB3	9	9	10	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up.
RB4	10	10	11	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST(2)	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST(2)	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin. Serial programming data.
RC0(3)	18	18	18	I/O	TTL	Bi-directional I/O port input buffer.
RC1(3)	19	19	19	I/O	TTL	Bi-directional I/O port input buffer.
RC2(3)	20	20	20	I/O	TTL	Bi-directional I/O port input buffer.
RC3(3)	21	21	21	I/O	TTL	Bi-directional I/O port input buffer.
RC4(3)	22	22	22	I/O	TTL	Bi-directional I/O port input buffer.
RC5(3)	23	23	23	I/O	TTL	Bi-directional I/O port input buffer.
RC6(3)	24	24	24	I/O	TTL	Bi-directional I/O port input buffer.
RC7(3)	25	25	25	I/O	TTL	Bi-directional I/O port input buffer.
VSS	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: O = Output, I/O = Input/output, P = Power, — = Not used, I = Input, ST = Schmitt Trigger input, TTL = TTL input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: PIC16C557 only.

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 3-2.

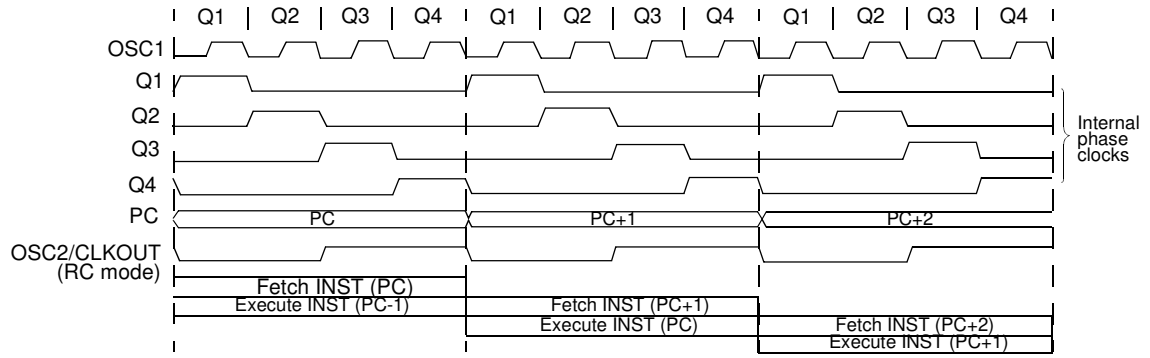
3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., `GOTO`), then two cycles are required to complete the instruction (Example 3-1).

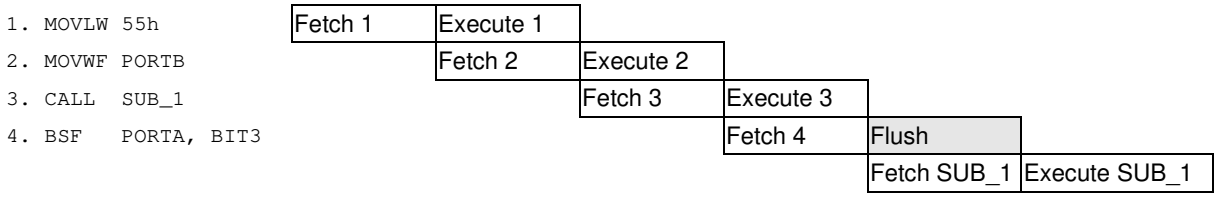
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the “Instruction Register (IR)” in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C55X has a 13-bit program counter capable of addressing an 8 K x 14 program memory space. Only the first 512 x 14 (0000h - 01FFh) for the PIC16C554 and 2K x 14 (0000h - 07FFh) for the PIC16C557 and PIC16C558 are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 512 x 14 spaces in the PIC16C554, or 2K x 14 space of the PIC16C558 and PIC16C557. The RESET vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1, Figure 4-2).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C554

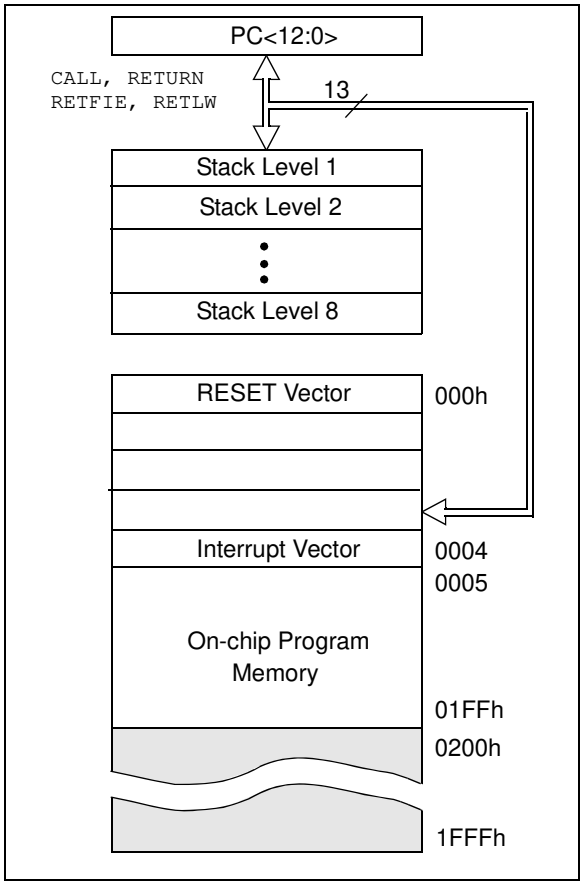
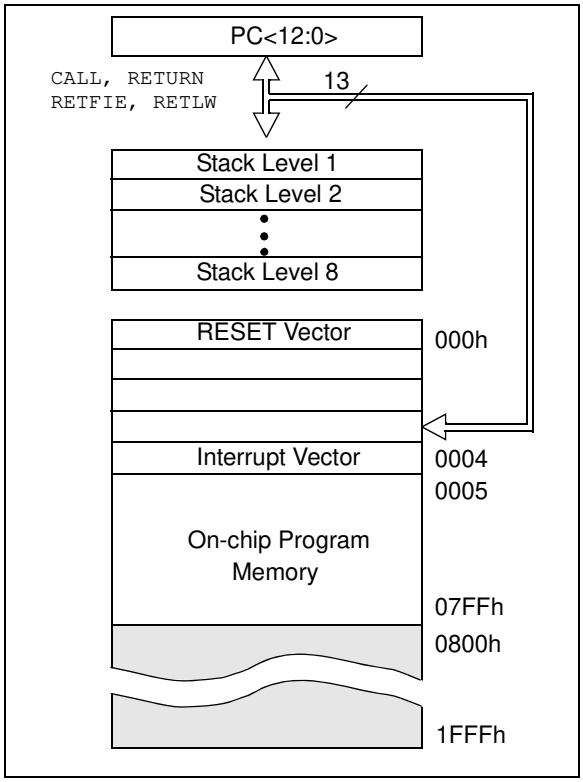


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C557 AND PIC16C558



4.2 Data Memory Organization

The data memory (Figure 4-3 through Figure 4-5) is partitioned into two banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). Bank 0 is selected when the RP0 bit (STATUS <5>) is cleared. Bank 1 is selected when the RP0 bit is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations 20-6Fh (Bank 0) on the PIC16C554 and 20-7Fh (Bank 0) and A0-BFh (Bank 1) on the PIC16C558 and PIC16C557 are General Purpose Registers implemented as static RAM. Some special purpose registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 80 x 8 in the PIC16C554 and 128 x 8 in the PIC16C557 and PIC16C558. Each can be accessed either directly or indirectly through the File Select Register, FSR (Section 4.4).

FIGURE 4-3: DATA MEMORY MAP FOR THE PIC16C554

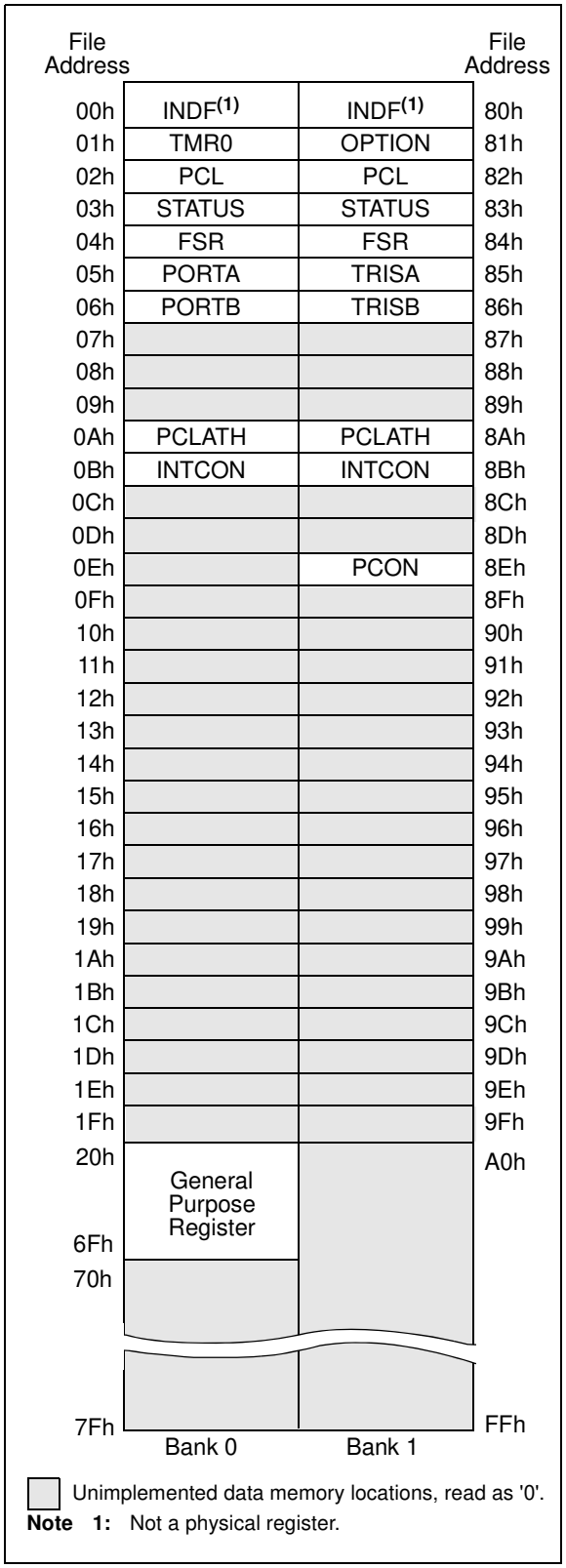


FIGURE 4-4: DATA MEMORY MAP FOR THE PIC16C557

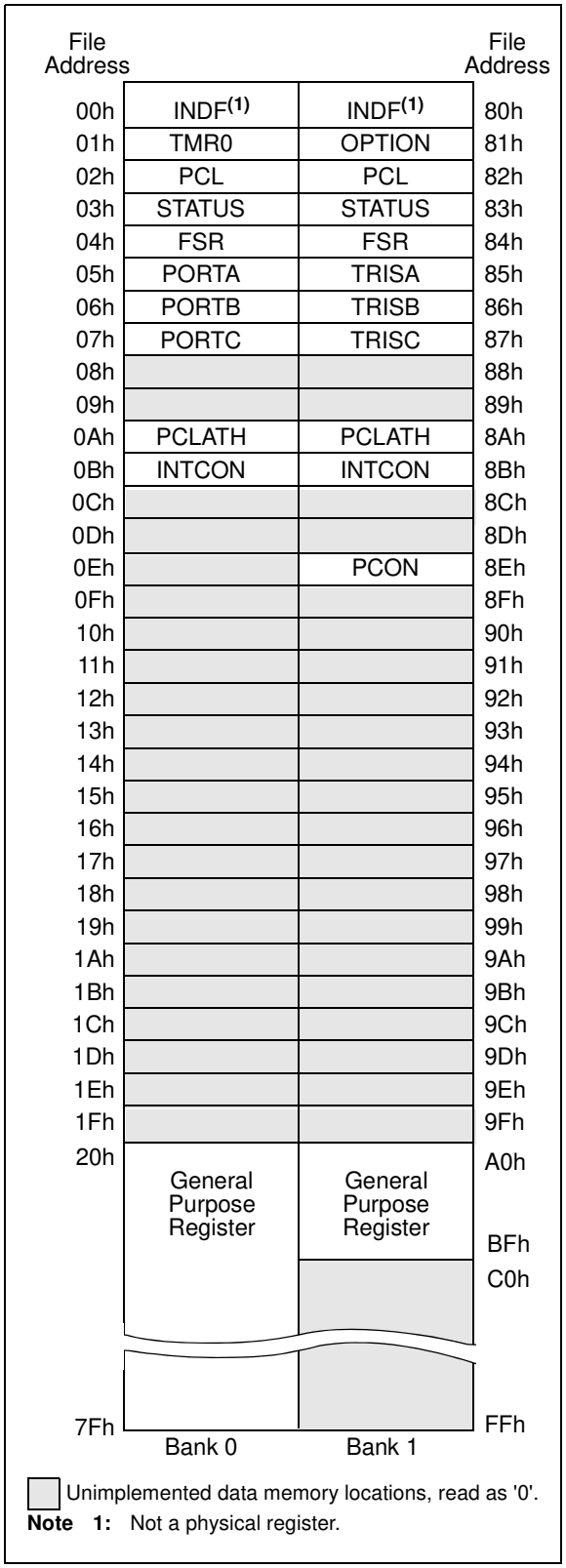
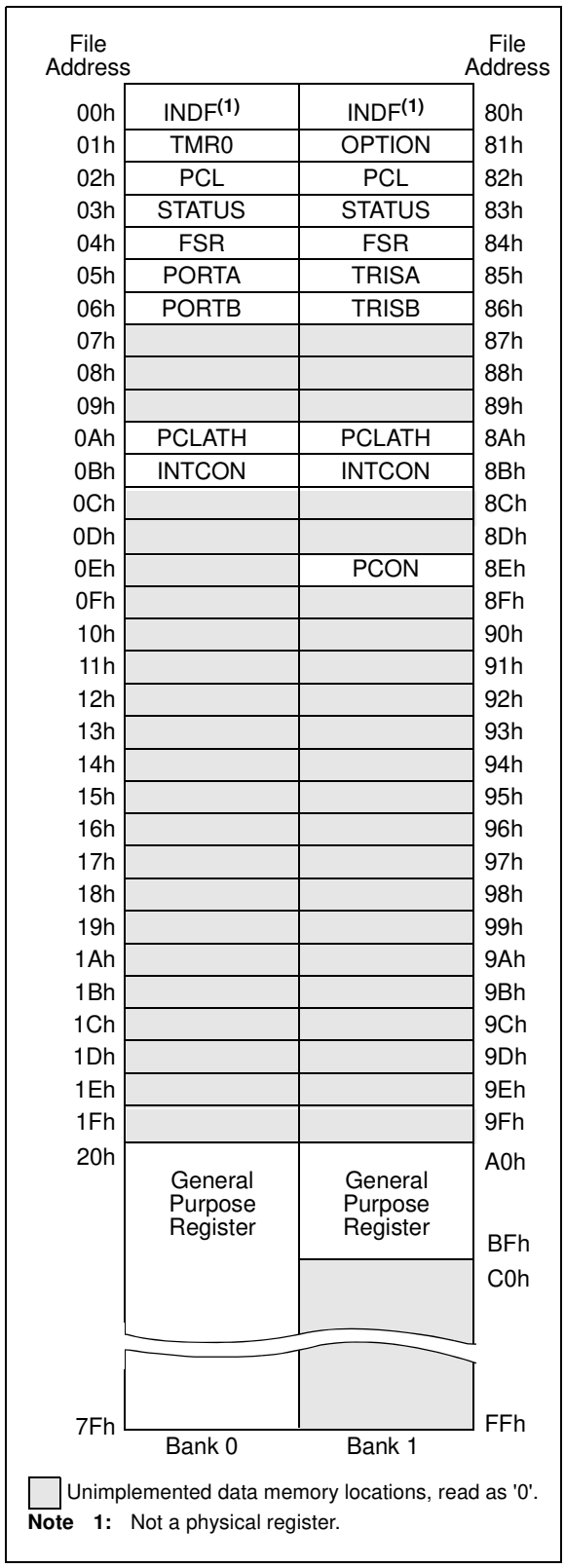


FIGURE 4-5: DATA MEMORY MAP FOR THE PIC16C558



4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (Table 4-1). These registers are static RAM.

The Special Function Registers can be classified into two sets (core and peripheral). The special function 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 of that peripheral feature.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C55X

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset	Detail on Page:
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	21
01h	TMR0	Timer0 Module's Register								xxxx xxxx	47
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	21
03h	STATUS	IRP(2)	RP1(2)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	17
04h	FSR	Indirect data memory address pointer								xxxx xxxx	21
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	---x xxxx	23
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	25
07h	PORTC(4)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	27
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	21
0Bh	INTCON	GIE	(3)	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	19
0Ch	—	Unimplemented								—	—
0Dh-1Eh	—	Unimplemented								—	—
1Fh	—	Unimplemented								—	—
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	21
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	18
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	21
83h	STATUS	—	—	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	17
84h	FSR	Indirect data memory address pointer								xxxx xxxx	21
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	23
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	25
87h	TRISC(4)	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	27
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	21
8Bh	INTCON	GIE	(3)	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	19
8Ch	—	Unimplemented								—	—
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	—	---- --0-	20
8Fh-9Eh	—	Unimplemented								—	—
9Fh	—	Unimplemented								—	—

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non Power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: IRP & RP1 bits are reserved, always maintain these bits clear.

3: Bit 6 of INTCON register is reserved for future use. Always maintain this bit as clear.

4: PIC16C557 only.

4.2.2.1 STATUS Register

The STATUS register, shown in Figure 4-2, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as the destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000uu1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions be used to alter the STATUS register because these instructions do not affect any status bits. For other instructions, not affecting any status bits, see the "Instruction Set Summary".

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C55X and should be programmed as '0'. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

2: The C and DC bits operate as a $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 4-1: STATUS REGISTER (ADDRESS 03h OR 83h)

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

bit 7 **IRP**: Register Bank Select bit (used for Indirect addressing)
`1` = Bank 2, 3 (100h - 1FFh)
`0` = Bank 0, 1 (00h - FFh)
The IRP bit is reserved on the PIC16C55X, always maintain this bit clear

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for Direct addressing)
`11` = Bank 3 (180h - 1FFh)
`10` = Bank 2 (100h - 17Fh)
`01` = Bank 1 (80h - FFh)
`00` = Bank 0 (00h - 7Fh)
Each bank is 128 bytes. The RP1 bit is reserved on the PIC16C55X, always maintain this bit clear.

bit 4 **$\overline{TO}$**: Timeout bit
`1` = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
`0` = A WDT timeout occurred

bit 3 **$\overline{PD}$**: Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for $\overline{borrow}$ the polarity is reversed)
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`,`SUBLW`,`SUBWF` instructions)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

Note 1: For $\overline{borrow}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

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

4.2.2.2 OPTION Register

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT interrupt, TMR0 and the weak pull-ups on PORTB.

Note 1: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT (PSA = 1).

REGISTER 4-2: OPTION REGISTER (ADDRESS 81H)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

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

4.2.2.3 INTCON Register

The INTCON register is a readable and writable register which contains the various enable and flag bits for all interrupt sources.

Note: Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

REGISTER 4-3: INTCON REGISTER (ADDRESS 0BH OR 8BH)

R/W-0	Reserved	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	—	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit7							bit0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

bit 6 **Reserved:** For future use. Always maintain this bit clear.

bit 5 **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **T0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit
1 = When at least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

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

4.2.2.4 PCON Register

The PCON register contains a flag bit to differentiate between a Power-on Reset, an external $\overline{MCLR}$ Reset or WDT Reset. See Section 6.3 and Section 6.4 for detailed RESET operation.

REGISTER 4-4: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	$\overline{POR}$	—
bit7							bit0

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **$\overline{POR}$**: Power-on Reset status bit
1 = No Power-on Reset occurred
0 = Power-on Reset occurred

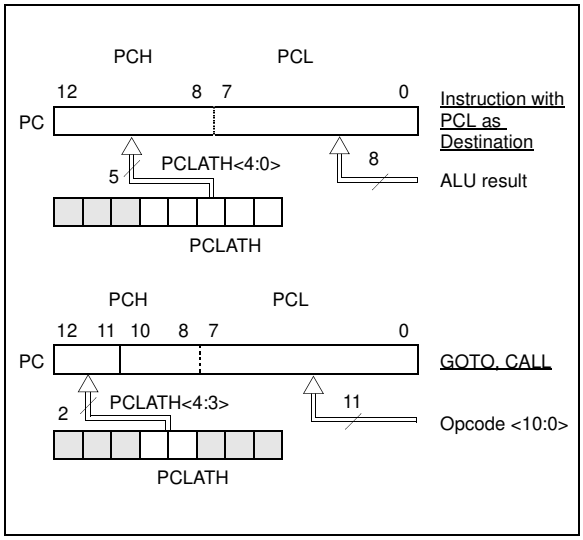
bit 0 **Unimplemented:** Read as '0'

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

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high bits (PC<12:8>) are not directly readable or writable and come from PCLATH. On any RESET, the PC is cleared. Figure 4-6 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 4-6 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 4-6: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note *"Implementing a Table Read"* (AN556).

4.3.2 STACK

The PIC16C55X family has an 8-level deep x 13-bit wide hardware stack (Figure 4-1 and Figure 4-2). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no status bits to indicate stack overflow or stack underflow conditions.

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

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

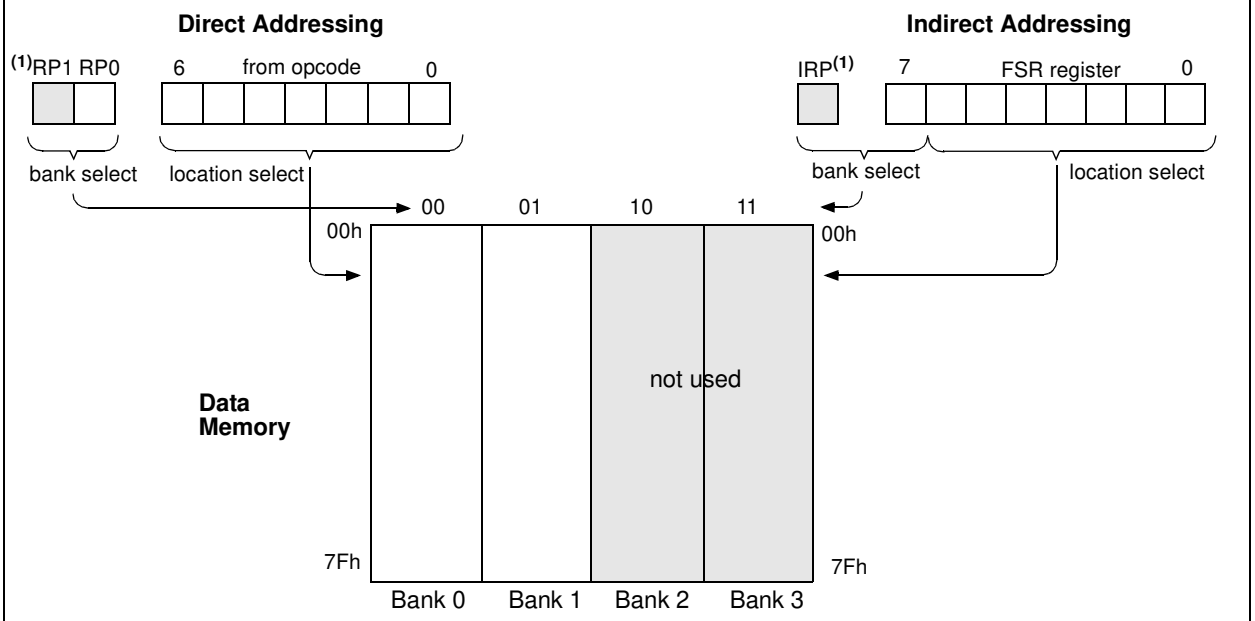
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-7. However, IRP is not used in the PIC16C55X.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
        movlw   0x20    ;initialize pointer
        movwf   FSR     ;to RAM
NEXT    clrf    INDF    ;clear INDF register
        incf    FSR     ;inc pointer
        btfss   FSR,4   ;all done?
        goto    NEXT    ;no clear next
                        ;yes continue
CONTINUE:
```

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING PIC16C55X



For memory map detail see Figure 4-3 and Figure 4-5.

Note 1: The RP1 and IRP bits are reserved, always maintain these bits clear.

5.0 I/O PORTS

The PIC16C554 and PIC16C558 have two ports, PORTA and PORTB. The PIC16C557 has three ports, PORTA, PORTB and PORTC.

5.1 PORTA and TRISA Registers

PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open-drain output. Port RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

A '1' in the TRISA register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Note 1: On RESET, the TRISA register is set to all inputs.

FIGURE 5-1: BLOCK DIAGRAM OF PORT PINS RA<3:0>

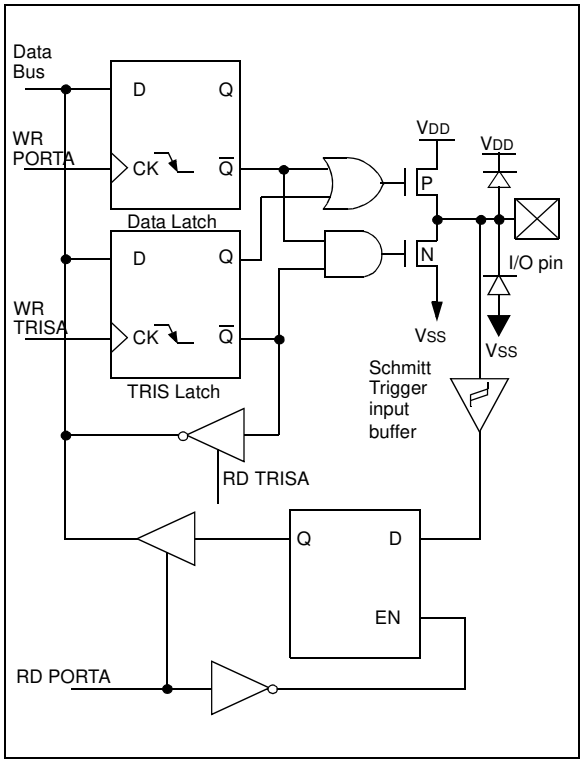


FIGURE 5-2: BLOCK DIAGRAM OF RA4 PIN

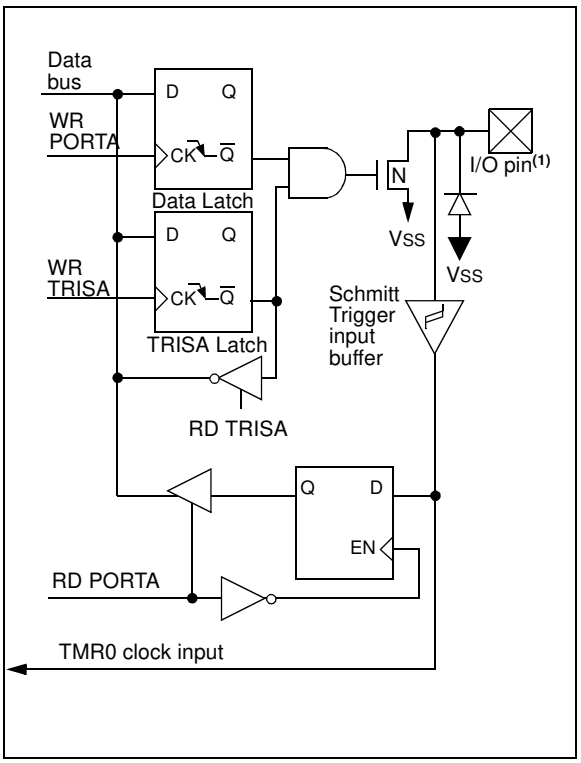


TABLE 5-1: PORTA FUNCTIONS

Name	Bit #	Buffer Type	Function
RA0	Bit 0	ST	Bi-directional I/O port.
RA1	Bit 1	ST	Bi-directional I/O port.
RA2	Bit 2	ST	Bi-directional I/O port.
RA3	Bit 3	ST	Bi-directional I/O port.
RA4/T0CKI	Bit 4	ST	Bi-directional I/O port or external clock input for TMR0. Output is open drain type.

Legend: ST = Schmitt Trigger input

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other RESETS
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: — = Unimplemented locations, read as '0', x = unknown, u = unchanged

Note 1: Shaded bits are not used by PORTA.

5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

Reading PORTB register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Each of the PORTB pins has a weak internal pull-up (≈200 μA typical). A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on Power-on Reset.

Four of PORTB's pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>). This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

- Any read or write of PORTB (this will end the mismatch condition)
- Clear flag bit RBIF

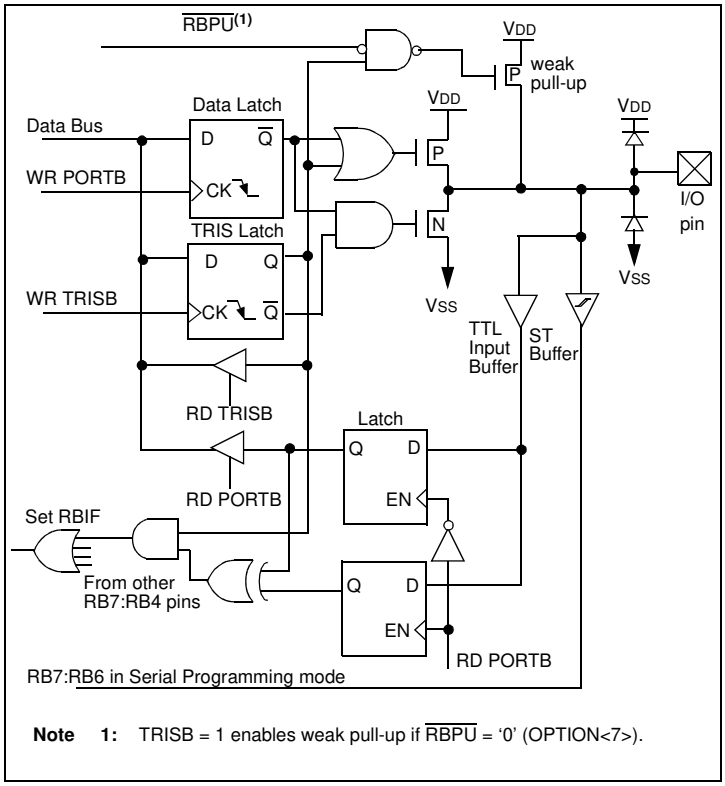
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition, and allow flag bit RBIF to be cleared.

The interrupt on mismatch feature, together with software configurable pull-ups on these four pins, allows easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552 in the Microchip *Embedded Control Handbook*.)

Note 1: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

FIGURE 5-3: BLOCK DIAGRAM OF RB7:RB4 PINS



Note 1: TRISB = 1 enables weak pull-up if $\overline{\text{RBPU}}$ = '0' (OPTION<7>).