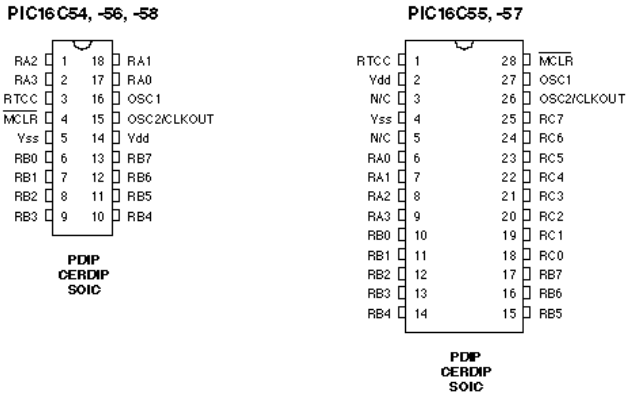
Condensed PIC Data Sheets

BLANK PAGE

PIC16C5x Devices

PIC16C5x Pin-Outs

The following diagrams show the 18-pin and 28-pin PIC16C5x pin-outs:



Pin	Function
RA0 - RA3	I/O Port A
RB0 - RB7	I/O Port B
RC0 - RC7	I/O Port C (only on 28-pin PIC's)
RTCC	Real-time clock/counter input
MCLR	Master clear (reset)
OSC1	Oscillator input
OSC2/CLKOUT	Oscillator output (OSC/4)
Vdd	Power supply
Vss	Ground
N/C	No connection

PIC16C5x Devices

PIC16C5x Microcontrollers

The table below shows the various PIC16C5x devices available:

PART #	ERASE	(E)PROM	RAM	I/O	SUPPLY	OSC.	FREQ.
PIC16C54-RC/P	No	512 x 12	32 x 8	12	3.00 - 6.25 V	RC	DC - 4 MHz
PIC16C54-XT/P	No	512 x 12	32 x 8	12	3.00 - 6.25 V	XTAL	DC - 4 MHz
PIC16C54-HS/P	No	512 x 12	32 x 8	12	4.50 - 5.50 V	XTAL	DC - 20 MHz
PIC16C54-LP/P	No	512 x 12	32 x 8	12	2.50 - 6.25 V	XTAL	DC - 40 kHz
PIC16C54/JW	Yes	512 x 12	32 x 8	12	3.00 - 5.50 V	RC,XTAL	DC - 20 MHz
PIC16C55-RC/P	No	512 x 12	32 x 8	20	3.00 - 6.25 V	RC	DC - 4 MHz
PIC16C55-XT/P	No	512 x 12	32 x 8	20	3.00 - 6.25 V	XTAL	DC - 4 MHz
PIC16C55-HS/P	No	512 x 12	32 x 8	20	4.50 - 5.50 V	XTAL	DC - 20 MHz
PIC16C55-LP/P	No	512 x 12	32 x 8	20	2.50 - 6.25 V	XTAL	DC - 40 kHz
PIC16C55/JW	Yes	512 x 12	32 x 8	20	3.00 - 5.50 V	RC,XTAL	DC - 20 MHz
PIC16C56-RC/P	No	1K x 12	32 x 8	12	3.00 - 6.25 V	RC	DC - 4 MHz
PIC16C56-XT/P	No	1K x 12	32 x 8	12	3.00 - 6.25 V	XTAL	DC - 4 MHz
PIC16C56-HS/P	No	1K x 12	32 x 8	12	4.50 - 5.50 V	XTAL	DC - 20 MHz
PIC16C56-LP/P	No	1K x 12	32 x 8	12	2.50 - 6.25 V	XTAL	DC - 40 kHz
PIC16C56/JW	Yes	1K x 12	32 x 8	12	3.00 - 5.50 V	RC,XTAL	DC - 20 MHz
PIC16C57-RC/P	No	2K x 12	80 x 8	20	3.00 - 6.25 V	RC	DC - 4 MHz
PIC16C57-XT/P	No	2K x 12	80 x 8	20	3.00 - 6.25 V	XTAL	DC - 4 MHz
PIC16C57-HS/P	No	2K x 12	80 x 8	20	4.50 - 5.50 V	XTAL	DC - 20 MHz
PIC16C57-LP/P	No	2K x 12	80 x 8	20	2.50 - 6.25 V	XTAL	DC - 40 kHz
PIC16C57/JW	Yes	2K x 12	80 x 8	20	3.00 - 5.50 V	RC,XTAL	DC - 20 MHz
PIC16LC58A-04/P	No	2K x 12	80 x 8	20	2.50 - 6.25 V	RC,XTAL	DC - 4 MHz
PIC16C58A-04/P	No	2K x 12	80 x 8	20	3.00 - 6.25 V	XTAL	DC - 4 MHz
PIC16C58A-20/P	No	2K x 12	80 x 8	20	3.00 - 6.25 V	XTAL	DC - 20 MHz
PIC16C58A/JW	Yes	2K x 12	80 x 8	20	3.00 - 6.25 V	RC,XTAL	DC - 20 MHz

Peripheral Features

In addition to the obvious features shown above, the PIC's have a number of not-so-obvious features and qualities that are important:

- **Fully static operation,** allowing you to stop the oscillator and then restart where you left off, with all registers intact. This is useful in applications where power conservation is important.
- **Individual I/O pins are programmable as inputs or outputs.**
  Sink current: 25 mA per pin, 50 mA per port
  Source current: 20 mA per pin, 40 mA per port
- **2-level hardware stack.**

Peripheral Features *(continued)*

- **Real-Time Clock/Counter (RTCC).** The RTCC is an 8-bit counter, which can be driven by the RTCC pin or by the PIC's internal instruction clock (OSC/4). If the external pin is used, the counter can be set to increment on low-to-high or high-to-low transitions.

  Normally, the RTCC is driven directly by either source. For higher count values, though, the prescaler can be used to effectively increase the RTCC to 16 bits.

  The RTCC signal source and trigger edge are determined by bits in the Option register (see next section for register descriptions).

  See the Microchip PIC16C5x data sheet for details concerning timing characteristics for the RTCC's external input.

- **Watchdog Timer (WDT).** When enabled, the watchdog timer is used to reset the PIC if the program has "crashed." Normally, a CLR WDT instruction in the main loop of your program would prevent the watchdog timer from ever timing out and resetting the PIC. However, if the program was not executing properly, the watchdog timer would reset the PIC.

  The watchdog timer works from an internal oscillator, allowing it to run even if the PIC's main oscillator has stopped. The watchdog timer's normal time-out period is 18 ms, but can be increased to several seconds by using the post-scaler.

- **Prescaler/Post-scaler.** This 8-bit counter can be assigned to the RTCC (as a prescaler) or the watchdog timer (as a post-scaler). For simplicity, this counter is normally referred to as the "prescaler," even when it's used as a post-scaler.

  When assigned to the RTCC, the prescaler is placed between the RTCC and its clock source. The clock signal which would normally increment the RTCC, increments the prescaler. When the prescaler overflows, the RTCC is incremented. Increment ratios from 1:2 - 1:256 can be used, effectively giving you a 16-bit RTCC.

  When assigned to the watchdog timer, the prescaler is placed between the watchdog timer and the PIC's reset circuit. The watchdog timer signal which would normally reset the PIC,

Peripheral Features *(continued)*

increments the prescaler. When the prescaler overflows, the PIC is reset. Delay ratios from 1:1 - 1:128 can be used, allowing the watchdog period to be set from 18 ms to several seconds.

The prescaler setup is determined by bits in the Option register *(see the next section for register descriptions)*.

- **Code-protect fuse.** This is a special fuse that can be blown during programming; once the fuse is blown, the PIC's EPROM cannot be read.
- **Power saving sleep mode.** For applications where low power consumption is important, the PIC has a sleep mode. This mode is entered by executing a SLEEP instruction, which shuts down the oscillator. I/O pins maintain whatever state they had when the sleep mode was entered.

  To awaken the PIC from sleep mode, a reset must be performed, either from the MCLR pin or from a watchdog timer time-out.

  Once the PIC is running, the "PD" and "TO" bits in the Status register can be read to determine 1) if the PIC was powered up or awakened from sleep, and 2) if the wake-up was caused by an external reset or by the watchdog timer.
- **Programmable oscillator type.** The PIC can run with any of four oscillator types, as shown below:

LP:	Low power crystal	(DC - 40 KHz)
RC:	Resistor & capacitor	(DC - 4 MHz, ± 13-39 percent)
XT:	Crystal or resonator	(100 KHz - 4 MHz)
HS:	High speed crystal	(4 - 20 MHz)

  The oscillator type is determined by two EPROM bits, which are normally programmed at the factory. The RC and erasable PIC's, however, can be user-programmed for any of the oscillator types. Keep in mind, though, that RC-type PIC's are only tested for use with RC oscillators.

PIC16C5x Devices

Internal Architecture

Internally, the PIC is based on a register file concept with separate busses and memories for data and instructions (sometimes called "Harvard architecture"). The data bus and memory (RAM) are 8-bits wide, while the program bus and memory (EPROM) are 12-bits wide. All PIC instructions and their operands fit into a single 12-bit word, resulting in smaller code and faster execution. PIC programs are typically 33-50 percent smaller than programs written for 8-bit processors. And most instructions execute in a single instruction cycle (4 clock cycles); instructions that affect the program counter take an extra instruction cycle, for a total of 8 clock cycles. To further increase speed, the PIC uses overlapping instruction fetch and execution cycles; while one instruction is executed, the following instruction is being read from program memory. Because of its efficiency, the PIC can deliver 5 MIPS execution with a clock frequency of 20 MHz.

PIC16C5x Registers

The following table shows the various PIC16C5x registers; the function of each register is described in the following pages.

Register	Function
00h	Indirect addressing register
01h	Real-time clock/counter (RTCC)
02h	Program counter (PC)
-	Stack registers (2)
03h	Status register
04h	File select register (FSR)
05h	I/O Port A
06h	I/O Port B
07h	I/O Port C
-	TRISA
-	TRISB
-	TRISC
-	W register
-	Option register
08h - 0Fh	General purpose registers
10h - 1Fh	General purpose registers (4 banks in PIC16C57)

PIC16C5x Registers *(continued)*

Quick reference register map:

	BANK 0	BANK 1*	BANK 2*	BANK 3*
	W register	W register	W register	W register
	Stack (2)	Stack (2)	Stack (2)	Stack (2)
	Option register	Option register	Option register	Option register
	TRISA	TRISA	TRISA	TRISA
	TRISB	TRISB	TRISB	TRISB
	TRISC	TRISC	TRISC	TRISC
00h	Indirect addr.	Indirect addr.	Indirect addr.	Indirect addr.
01h	RTCC	RTCC	RTCC	RTCC
02h	PC	PC	PC	PC
03h	STATUS	STATUS	STATUS	STATUS
04h	FSR	FSR	FSR	FSR
05h	PORT A	PORT A	PORT A	PORT A
06h	PORT B	PORT B	PORT B	PORT B
07h	PORT C	PORT C	PORT C	PORT C
08h 0Fh	8 general purpose registers (RAM)	reads/writes registers 08h - 0Fh	reads/writes registers 08h - 0Fh	reads/writes registers 08h - 0Fh
10h 1Fh	16 general purpose registers (RAM)	30h 16 general purpose registers (RAM)* 3Fh	50h 16 general purpose registers (RAM)* 5Fh	70h 16 general purpose registers (RAM)* 7Fh
	(00h - 1Fh)	(20h - 3Fh)*	(40h - 5Fh)*	(60h - 7Fh)*

* Available on PIC16C57 only.

PIC16C5x Registers *(continued)*

The following text describes the function of each register. For some of the registers, you'll notice the designation "xxh" following the register name. This indicates the address of the register. Registers with no address cannot be addressed directly.

- **Indirect Addressing Register (00h).** This register doesn't actually exist. Naming register 00h in an instruction causes the PIC to read the register *pointed to* by register 04h (file select register). For example, the instruction "`ADD 00h,#05`" will *not* add five to register 00h; instead, it will add five to whatever register is *pointed to* by the address in register 04h.

  If register 00h itself is read through register 04h (04h contains "00h"), 00h will be returned. If register 00h is written to through register 04h, the PIC will execute a NOP.

- **Real-Time Clock/Counter (01h: RTCC).** This is the location of the RTCC (see previous section for a description). Although its contents may change in response to a clock signal, the RTCC register may be read and written just as any other register.

- **Program Counter (02h: PC).** The program counter holds the address for the instruction currently being executed. The program counter and its associated two-level stack are 9-11 bits wide, depending on the EPROM size of the PIC being used.

  Certain instructions affect the program counter, as shown below:

  GOTO (Microchip) and **JMP** (Parallax) load the lower 9 bits of the program counter. In the PIC16C56 and '57, which have more than 512 words of EPROM, the upper two bits of the program counter are loaded with the page select bits from the status register. The Parallax instruction set includes a convenient instruction, **LJMP**, which sets the page select bits before executing the jump.

  CALL loads the lower 8 bits of the program counter and clears the ninth bit. The program counter + 1 is pushed into the stack. In the '56 and '57, the upper two bits of the program counter are loaded with the page select bits from the status register. The Parallax instruction set includes **LCALL**, which

PIC16C5x Registers *(continued)*

sets the page select bits before executing the jump.

RETLW (Microchip) and **RETW** (Parallax) load the program counter with the address most recently pushed on the stack by a CALL instruction.

Instructions which load a computed value into the program counter, such as **JMP PC+W**, load the value into the lower 8 bits. The ninth bit of the program counter is cleared. In the PIC16C56 and '57, the upper two bits of the program counter are loaded with the page select bits from the status register.

It should be noted that because the ninth bit of the program counter is cleared by CALL instructions and computed value instructions (such as JMP PC+W), all subroutine calls and computed jumps must have their destination in the first 256 locations of any page (each page is 512 words).

As you may have noticed when reading the JMP and CALL paragraphs above, the program counter may not be loaded as expected when using a PIC with more than 512 words of EPROM. This is because the upper two bits of the program counter are loaded with the page select bits from the status register. If your program continues into the second page of memory and executes a JMP without having properly set the page select bits, execution may jump to another page (probably not what you want). To avoid this mistake, make sure to set the page select bits for the correct page. Or, use the Parallax "long" instructions, which do this for you (long instructions are LCALL, LJMP, and LSET).

- **Stack.** The stack is a pair of registers which are used for calling and returning from subroutines. The stack is affected by two instructions:

  When a **CALL** is executed, the first stack register is copied into the second register, then the program counter + 1 (the return address) is loaded into the first register. The original contents of the second register are lost. Finally, the program counter is loaded with the subroutine address, at which point execution continues.

PIC16C5x Registers *(continued)*

When a **RETLW** (Microchip) or **RETW** (Parallax) is executed, the first stack register is copied into the program counter, then the second register is copied into the first register. Execution continues at the address loaded from the first stack register.

- **Status Register (03h).** This register contains the status of the PIC's arithmetic logic unit (ALU), the reset status, and the page select bits for PIC's with more than 512 words of EPROM.

  The function of each bit in the status register is shown below:

Bit	Function
0	**Carry bit (C).** Set if an addition or subtraction causes an overflow from the most significant bit of the resultant (bit 7). Subtraction is included because it's executed by adding the two's complement. Also used by rotate instructions, which rotate the contents of a register and copy the low or high order bit of the register into the carry bit.
1	**Digit carry bit (DC).** Set if an addition or subtraction causes an overflow from the 4th low order bit (bit 3). Digit carry indicates that more than one hex digit (4 bits) was necessary to accomodate the result.
2	**Zero bit (Z).** Set if the result of an arithmetic or logic operation is zero.
3	**Power-down bit (PD).** Set during power-up or by a CLR WDT (clear watchdog) instruction. Cleared by a SLEEP instruction.
4	**Time-out bit (TO).** Set during power-up, by CLR WDT, or by SLEEP. Cleared by a watchdog time-out.
5-6	**Page select bits (PA0, PA1).** In the '54 and '55, these are unused. In the '56, bit 5 selects program page 0 or 1 (bit 6 is unused). In the '57, both bits select page 0, 1, 2, or 3. Each page is 512 words long.

PIC16C5x Registers *(continued)*

Bit	Function
7	**Unused bit (PA2).** Reserved by Microchip for future use.

The following table shows how various events affect the power-down and time-out bits:

Event	PD	TO
Power-up	1	1
Watchdog time-out	x	0
SLEEP instruction	0	1
CLR WDT instruction	1	1

Lastly, this table shows the status of the power-down and time-out bits after a reset:

Cause of Reset	PD	TO
Watchdog time-out (not during sleep)	1	0
Watchdog time-out (during sleep)	0	0
External reset (not during sleep)	x	x
External reset (during sleep)	0	1
Normal power-up	1	1

- **File Select Register (04h: FSR).** This register serves a dual purpose: it selects the register for indirect addressing, and it selects the current register bank in the PIC16C57.

  In all PIC16C5x devices, bits 0-4 select one of 32 registers in the current bank (only the '57 has more than one bank).

  In the PIC16C57, bits 5-6 select one of four register banks. Each bank has 32 registers. However, reading or writing the lower 16 registers in **any** bank will access registers 00h - 0Fh (bank 0). Only the upper 16 registers in each bank are unique. This results in one bank of 32 registers, plus three banks of 16 registers, for a total of 80 registers.

PIC16C5x Registers *(continued)*

The following table shows the function of each FSR bit:

Bit	Function
0-4	Available in all PIC's; select one of 32 registers in the current bank (only the '57 has more than 1 bank).
5-6	PIC16C57 only; select one of four register banks.
7	Read-only bit; always reads as "1".

This table shows the registers available in the various PIC's:

Registers	Description
00h - 07h	Special registers, bank 0 (RTCC, PC, etc.)
08h - 0Fh	General purpose registers, bank 0
10h - 1Fh	General purpose registers, upper half of bank 0
———	**End of register memory in '54, '55, and '56.**
20h - 2Fh	Reads/writes registers 00h - 0Fh
30h - 3Fh	General purpose registers, upper half of bank 1
40h - 4Fh	Reads/writes registers 00h - 0Fh
50h - 5Fh	General purpose registers, upper half of bank 2
60h - 6Fh	Reads/writes registers 00h - 0Fh
70h - 7Fh	General purpose registers, upper half of bank 3

The register selected in the FSR can be accessed in the indirect addressing mode *(see description of indirect addressing register, earlier in this section).*

In the PIC16C57, all registers can be accessed through indirect addressing. However, to access registers above 1Fh using direct instructions, you must set the page select bits (5-6) to the appropriate page, then read or write the corresponding register in

PIC16C5x Registers *(continued)*

bank 0. For instance, to load register 30h with #A5h, you would execute the following instructions:

```
MOV   04h,#00100000b    ;Select bank 1

MOV   10h,#A5h          ;Load register 10h in
                        ;bank 1 (register 30h)
                        ;with #A5h
```

- **I/O Port A (05h: RA or Port A).** 4-bit I/O port. This register is used to read and write I/O Port A. This register can be read and written just as any other register. However, read instructions always read the I/O pins, regardless of whether the pins are programmed as inputs or outputs.

  The upper 4 bits are unused and read as 0's.

- **I/O Port B (06h: RB or Port B).** 8-bit I/O port.

- **I/O Port C (07h: RC or Port C).** 8-bit I/O port, only available on 28-pin PIC's. On 18-pin PIC's, this register can be used for storage.

- **TRISA (TRI-State A).** This is the data direction register for Port A. Bits in this register which are set to "1" cause the corresponding bits in Port A to become inputs (the pins go into high impendence mode, allowing them to be driven by an external source). Bits which are cleared to "0" cause the corresponding bits in Port A to become outputs.

  The data direction registers are not directly addressable; to change their contents, you can use either of these instructions:

`TRIS port_fr`	(Microchip) Copies W into the data direction register for *port_fr*, where *port_fr* is 05h-07h.
`MOV !port_fr,#literal`	(Parallax) Copies *literal* into the data direction register for *port_fr*, where *port_fr* is 05h-07h.

PIC16C5x Registers *(continued)*

- **TRISB.** Data direction register for Port B.
- **TRISC.** Data direction register for Port C.
- **W** (working register). The W register holds the second operand in two-operand instructions and is used in internal data transfer; much like the "accumulator" in other processors.
- **Option Register.** This register defines the prescaler ratio, prescaler assignment, RTCC trigger edge, and RTCC signal source. The function of each bit in the option register is shown below:

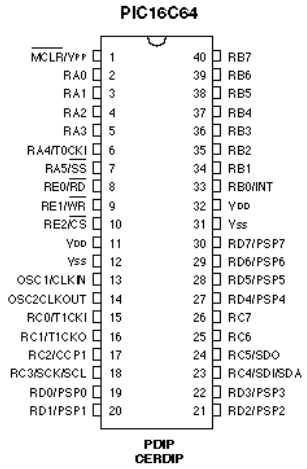
Bit	Function
0-2	**Prescaler ratio.** These 3 bits determine the prescaler input-to-output ratio. When using the prescaler with the RTCC, the seven possible ratios are 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256. When using the prescaler with the watchdog timer, the ratios are 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128. For example, let's say that the prescaler is assigned to the watchdog timer. To increase the watchdog time-out period to 64 times its normal length, the prescaler ratio would be set to 110b; this yields a watchdog period of approx. 1 second (64 x 18 ms).
3	**Prescaler assignment.** This bit determines whether the prescaler is assigned to the RTCC ("0") or to the watchdog timer ("1").
4	**RTCC trigger edge.** This bit determines whether the RTCC increments on a low-to-high ("0") or high-to-low ("1") transition on the RTCC pin.
5	**RTCC signal source.** This bit determines whether the RTCC is driven by the PIC's internal instruction clock ("0") or by the RTCC pin ("1").

- **General purpose registers.** These registers may be used by your program for storage of variables, data, etc.

BLANK PAGE

PIC16C64 Pin-Out

The following diagram shows the PIC16C64 pin-out:



Pin	Function
RA0 - RA5	I/O Port A
RB0 - RB7	I/O Port B
RC0 - RC7	I/O Port C
RD0 - RD7	I/O Port D
RE0 - RE2	I/O Port E
CCP1	Capture1 input,Compare1 output, PWM1 output
PSP0 - PSP7	Parallel slave port
$\overline{RD}$	Parallel slave port read control
$\overline{WR}$	Parallel slave port write control
$\overline{CS}$	Parallel slave port select control
$\overline{SS}$	Slave select for synch. serial
SCK/SCL	SPI and I^2C Synch. serial clock
SDI/SDA	SPI data in or I^2C data I/O
SDO	SPI data out
T0CKI	Timer0 input
T1CKI	Timer1 input
T1CKO	Timer1 output

(continued)

PIC16C64 Pin-Out *(continued)*

Pin	Function
$\overline{\text{INT}}$	External interrupt input
$\overline{\text{MCLR}}$	Master clear (reset)
OSC1/CLKIN	Oscillator input
OSC2/CLKOUT	Oscillator output (OSC/4)
V_{DD}	Power supply
V_{SS}	Ground
V_{PP}	Programming voltage input

The table below shows the various PIC16C64s available:

Part #	Erasable	Program	Registers	I/O	Power	Osc. Type	Frequency
PIC16C64-04	No	2K x 14	128 x 8	33	4.0 - 6.0 V	RC,XTAL	DC - 4 MHz
PIC16C64-20	No	2K x 14	128 x 8	33	4.0 - 6.0 V	RC,XTAL	DC - 20 MHz
PIC16C64-20/JW	Yes	2K x 14	128 x 8	33	4.0 - 6.0 V	RC,XTAL	DC - 20 MHz
PIC16LC64-04	No	2K x 14	128 x 8	33	2.5 - 6.0 V	RC,XTAL	DC - 4 MHz

New and Modified Features

The PIC16C64 builds upon the earlier PIC16C5x microcontrollers. It is the first 40-pin device in the PIC16Cxx family. The following overview shows how the PIC16C64 differs from the original PIC165x:

- **2K program space implemented in EPROM.**
- **128 general-purpose registers implemented in SRAM.**
- **Five I/O ports totalling 33 pins.**
- **Interrupts are possible from eight sources:** External interrupt, TMR0 overflow, TMR1 overflow, TMR2, CCP1 (Capture and Compare modes), synchronous serial port, and microprocessor port read/write.
- **Synchronous serial port with two modes: 3-wire SPI or I^2C.**

PIC16C64

New and Modified Features *(continued)*

- **Port B modifications:** Software-controlled pull-ups have been added, and an interrupt can ocurr when any of four pins changes state.
- **An 8-level hardware stack:** Allows deeper nesting of subroutines
- **A 14-bit instruction word:** Provides larger page sizes for program memory (2K) and RAM (128 bytes).
- **New timers:** Two new CPU timers control delays on power-up and wake-up. These timers are the oscillator start-up timer (OST) and power-up timer (PWRT). Three new timer/counter modules are available (TMR0, TMR1, TMR2). TMR0 replaces RTCC, and is identical in function. In asynchronous mode, TMR1 can run during sleep. Both TMR1 and TMR2 can function as a PWM time base.
- **New I/O pins on Port A:** Two additional I/O pins are available as bit 4 and bit 5 of PORTA. RA4 is a Schmitt trigger input and an open collector output. All other PortA pins are TTL input and CMOS output.
- **Status register changes:** Program page select bits (bit 5-6) have been replaced by register page select bits. Bit 6 and bit 7 are not used by the PIC16C64.
- **File select register changes:** The FSR has been increased to 8 bits, and bits 5-6 no longer function as register page select. Register page selection is now done in the Status register.
- **High byte added to PC:** A high byte has been added to the program counter to handle program memory paging.
- **Page select bits removed:** Program memory page select bits in the Status register (PA0 - PA2) have been removed.

  Note: The 2K program space can be utilized without worrying about page boundaries.
- **Reset vector moved to 0000h.**
- **Interrupt vector added at 0004h.**

New and Modified Features *(continued)*

- **The following directives and instructions have been added:**

Microchip Instruction	Parallax Equivalent
sublw	**mov w,#literal-w** Move #literal-W into W.
addlw	**add w,#literal** Add #literal to W.
return	**ret** Return, *without affecting W*.
retfie	**reti** Return from interrupt.

- **The following directives and instructions do not function with the PIC16C64:**

Microchip Instruction	Parallax Equivalent
-	**reset** The reset vector was moved to address 0000h, so the RESET directive was dropped. Setup the reset vector by making a JMP at 0000h to the start of your program; address 0005h is the first possible location for your program.
-	**ljmp, lcall, lset** Since there is no paged program memory, long instructions are unnecessary.
-	**addb fr,bit** Add bit into file register.
-	**subb fr,bit** Subtract bit from file register.

New and Modified Features *(continued)*

- **The following directives and instructions do not function with the PIC16C64** *(continued):*

Microchip Instruction	Parallax Equivalent
-	**neg fr** Negate file register.
-	**mov !port_fr, w**
-	**mov !port_fr, #literal**
-	**mov !port_fr, fr**
-	**mov !option, w**
-	**mov !option, #literal**
-	**mov !option, fr**

PIC16C64 Registers

The following table identifies the PIC16C64 registers. A quick reference map is on page 8, and the registers are described in detail beginning on page 9.

Register	Function
-	W register
00h	Indirect address register† (INDF)
01h	Timer0
02h	Program counter low byte (PCL)
-	Stack registers (8)
03h	Status register
04h	File select register (FSR)
05h	I/O Port A
06h	I/O Port B
07h	I/O Port C
08h	I/O Port D
09h	I/O Port E
0Ah	Program counter latch high (PCLATH)
0Bh	Interrupt control register (INTCON)
0Ch	Contains flag bit for peripheral interrupts (PIR1)
0Dh	Reserved
0Eh	Timer1 low byte (TMR1L)
0Fh	Timer1 high byte (TMR1H)
10h	Timer1 control register (T1CON)
11h	Timer2
12h	Timer2 control register (T2CON)
13h	SSP†† receive/transmit buffer (SSPBUF)
14h	SSP control register (SSPCON)
15h	Capture/compare/pwm low byte (CCPR1L)
16h	Capture/compare/pwm high byte (CCPR1H)
17h	Capture/compare/pwm control register (CCP1CON)
18h - 1Fh	Reserved
20h - 7Fh	General purpose registers
	End of Page 0 register memory

† Not a physical register
†† Synchronous serial port

PIC16C64 Registers *(continued)*

Register	Function
———	Start of Page 1 register memory
80h	Indirect address register†
81h	Option register
82h	Program counter low byte (PCL)
83h	Status register
84h	File select register (FSR)
85h	TRISA
86h	TRISB
87h	TRISC
88h	TRISD
89h	TRISE
8Ah	Program counter latch high (PCLATH)
8Bh	Interrupt control register (INTCON)
8Ch	Peripheral interrupt enable bits (PIE1)
8Dh	Reserved
8Eh	Contains reset cause flag (PCON)
8Fh - 91h	Reserved
92h	Timer2 period (PR2)
93h	SSP address register–I²C mode (SSPADD)
94h	SSP status register (SSPSTAT)
95h - 9Fh	Reserved
A0h - BFh	General purpose registers
C0h - FFh	Reserved

† Not a physical register

PIC16C64 Quick Reference Register Map

	PAGE 0	PAGE 1	
	W register	W register	
	Stack (8)	Stack (8)	
00h	INDIRECT	INDIRECT	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	PORTC	TRISC	87h
08h	PORTD	TRISD	88h
09h	PORTE	TRISE	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh	TMR1L	PCON	8Eh
0Fh	TMR1H		8Fh
10h	T1CON		90h
11h	TMR2		91h
12h	T2CON	PR2	92h
13h	SSPBUF	SSPADD	93h
14h	SSPCON	SSPSTAT	94h
15h	CCPR1L		95h
16h	CCPR1H		
17h	CCP1CON		
18h			
1Fh			9Fh
20h	General Purpose Registers	General Purpose	A0h
			BFh
			C0h
7Fh			FF

PIC16C64 Register Description

- **W** (working register): The W register holds the second operand in two-operand instructions and is used for internal data transfer. The W register is similar to the accumulator in other processors.
- **Stack.** The stack is comprised of eight registers which are used for calling and returning from subroutines. The program counter is pushed onto the stack when a CALL is executed or an interrupt is acknowledged. The stack is popped in the event of a RET, RETW, or RETI instruction.
- **Indirect Addressing Registers (00h & 80h).** These registers don't actually exist. Naming register 00h or 80h in an instruction causes the PIC to read the register *pointed to* by register 04h (file select register). Register 04h should contain the address of a register to read or write.
- **Timer0 (01h: TMR0).** TMR0 replaces RTCC, and is similar in function. TMR0 is an 8-bit overflow counter. The clock is driven externally through the TMR0 pin, or internally by OSC/4. When driven externally, the TMR0 module can be selected to increment on either low-to-high or high-to-low transitions. TMR0 has a programmable prescaler option, assigned through the OPTION register. The maximum external clock frequency is 50 MHz. See the Microchip PIC16C64 data sheet for details concerning external timing characteristics for TRM0.
- **Program Counter Low Byte (02h: PCL).** The program counter holds the address for the instruction currently being executed. The program counter and its associated eight-level stack are 13 bits wide, with PCL holding the lower 8 bits and PCH holding the upper 5 bits. PCH is automatically loaded when a JMP or CALL is performed. Although PCH is not addressable, your program can load PCH through the PCLATH register (see register f0A, later in this section).
- **Status register (03h).** This register contains the status of the arithmetic logic unit (ALU), the reset status, and the page select bits for register memory (not program memory, as in PIC16C5x devices).

PIC16C64 Register Description *(continued)*

The function of each bit in the **STATUS** register is shown below:

Bit	Label Name and Function
0	**Carry bit (C).** Set if an addition or subtraction causes an overflow from the most significant bit of the resultant (bit 7). Subtraction is included because it is executed by adding the two's complement. Also used by rotate instructions, which rotate the contents of a register and copy the low or high order bit of the register into the carry bit.
1	**Digit carry bit (DC).** Set if an addition or subtraction causes an overflow from the 4th low order bit (bit 3). Digit carry indicates that more than one hex digit (4 bits) was necessary to accommodate the result.
2	**Zero bit (Z).** Set if the result of an arithmetic or logic operation is zero.
3	**Power-down bit (PD).** Set during power-up or by a CLRWDT (clear watchdog) instruction. Cleared by a SLEEP instruction.
4	**Time-out bit (TO).** Set during power-up, by CLRWDT, or by SLEEP. Cleared by a watchdog timer time-out. The following table shows how various events affect the STATUS register power-down(PD) and time-out (TO) bits:

Event	PD	TO
Power-up	1	1
Watchdog time-out	x	0
SLEEP instruction	0	1
CLRWDT instruction	1	1

The following table shows the status of the power-down and time-out bits after a reset:

PIC16C64 Register Description *(continued)*

Cause of Reset	PD	TO
Watchdog time-out (not during sleep)	1	0
Watchdog time-out (during sleep)	0	0
External reset (not during sleep)	x	x
External reset (during sleep)	0	1
Normal power-up	1	1

5-6 **Register page select bits (RP0, RP1).** These bits determine which register page is selected for direct addressing operations. Each page is 128 bytes long, so only RP0 is valid in the PIC16C64. Future PICs may require RP1, so using this bit for general storage may prevent upward compatibility.

7 **Page select bit for indirect addressing (IRP).** The IRP bit is not used. Future PICs may require IRP, so using this bit for general storage may prevent upward compatibility.

- **File Select Register (04h: FSR).** This register selects the register for indirect addressing. Bits 0-7 select 1 of 256 registers in the current bank. As described in the PIC16C5x register descriptions, a read or write to register 00h will access the register pointed to by the FSR.
- **I/O Port A (05h: RA or Port A).** 6-bit I/O port. This register is for reading and writing I/O Port A. It can be read and written just as any other register. Read instructions always read the I/O pins, however, regardless of whether the pins are programmed as inputs or outputs.

  Bit 4 is a Schmitt trigger input and an open-collector output and shares its pin with the TMR0 clock input.
- **I/O Port B (06h: RB or Port B).** 8-bit I/O port. Each of the Port B pins has a weak internal pull-up resistor (~100 µA). A pin's pull-up is turned off if the pin is configured as an output, and a single bit in the OPTION register can turn off all the pull-ups. The pull-up resistors are disabled on power-on reset.

PIC16C64 Register Description *(continued)*

On bits 4-7, Port B has an interrupt on change feature that can generate an interrupt if any of the pins changes state. Any pin configured as an output is excluded from the interrupt feature.

This interrupt can awaken the chip from sleep. Along with the internal pull-ups, the interrupt from sleep feature makes it easy to interface to a keypad and have wake-up on key press.

- **I/O Port C (07h: RC or Port C).** 8-bit I/O port. All pins have a Schmitt trigger input buffer. Port C is multiplexed with peripheral functions as shown in the following table:

Pin	Option	Description
RC0	T1CKI	TMR1 clock input or TMR1 oscillator input
RC1	T1CKO	TMR1 oscillator output
RC2	CCP1	Capture1 input, Compare1 output, or PWM1 output
RC3	SCK/SCL	Synch. serial clock for SPI and I²C modes
RC4	SDI/SDA	SPI Data in or I²C data I/O
RC5	SDO	SPI Data out
RC6		No multiplexed functions
RC7		No multiplexed functions

- **I/O Port D (08h: RD or Port D).** 8-bit I/O port. All pins have a Schmitt trigger input buffer when configured as a general-purpose I/O port. Port D can also be configured as an 8-bit parallel slave port for interfacing with another microprocessor, in which case the inputs are TTL.
- **I/O Port E (09h: RE or Port E).** 3-bit I/O port. Each pin can be configured as a Schmitt trigger input or as an output. When PSPMODE bit 5 and bits 0-2 are set, Port E is a TTL control input for the parallel slave port (PORTD).
- **Program Counter Latch High (0Ah & 8Ah: PCLATH).** This accesses the 5 high bits of the program counter (PCH). Unlike the lower 8 bits (PCL), the 5 high bits are not directly addressable. Instead, they are stored in PCLATH for later use. When the program counter is loaded with a new value during a JMP, CALL, or a write to PCL, the high bits are loaded from PCLATH.

PIC16C64 Register Description *(continued)*

- **Interrupt Control Register (0Bh: INTCON).** This register is for enabling interrupts and for determining the cause of an interrupt. The function of each bit is given below:

Bit	Label Name and Function
0	**Port B interrupt (RBIF).** This bit is set if an interrupt was the result of a transition on any of the upper four bits of Port B. It remains set until cleared by software.
1	**External interrupt (INTF).** Set if an interrupt was caused by a transition on the external INT pin. Must be cleared by software.
2	**Timer0 overflow interrupt (RTIF).** Set if an interrupt was caused by an overflow in TMR0 (RTCC). Must be cleared by software.
3	**Port B interrupt enable (RBIE).** Determines whether the Port B interrupt is enabled (“1”) or disabled (“0”).
4	**External interrupt enable (INTE).** Determines whether the external interrupt is enabled (“1”) or disabled (“0”).
5	**Timer interrupt enable (RTIE).** Determines whether the TMR0 (RTCC) interrupt is enabled (“1”) or disabled (“0”).
6	**Peripheral interrupt enable (PEIE).** Determines whether unmasked peripheral interrupts are enabled (“1”) or disabled (“0”).
7	**Global interrupt enable (GIE).** Clearing this bit disables all interrupts. Setting this bit allows all interrupts that are individually enabled in bits 3-6.

Note: Register PIR1 (0Ch) contains the peripheral interrupt flags, and register PIE1 contains the peripheral interrupt enable bits. Each interrupt flag is set independently of its mask bit or the GIE bit.

PIC16C64 Register Description *(continued)*

- **Peripheral Interrupt Register (0Ch: PIR1).** This register contains interrupt flag bits. Interrupt flags are set by interrupt events, regardless of whether or not some or all interrupts are disabled. Before enabling interrupts, clear the interrupt flag to prevent a jump to the interrupt service routine.

Bit	Label Name and Function
0	**Timer1 interrupt flag (TMR1IF).** When set, TMR1 has overflowed, and TMR1IF must be cleared in software. When clear, TMR1 has not overflowed.
1	**Timer2 interrupt flag (TMR2IF).** When set, TMR2 has overflowed, and TMR2IF must be cleared in software. When clear, TMR2 has not overflowed.
2	**Capture1/Compare1/PWM1 interrupt flag (CCP1IF).** In Capture mode, if CCP1IF is set, a TMR1 capture has occurred. In Compare mode, if CCP1IF is set, a TMR1 compare match has occurred. Both set conditions must be cleared in software. A clear setting in either mode means that the corresponding condition did not occur. The PWM Mode for CCP1IF is not used.
3	**Synchronous serial port interrupt flag (SSPIF).** If set, a transmit or receive has been completed. If clear, the port is waiting to transmit or receive.
4-6	Unimplemented and read as “0”.
7	**Parallel slave port read/write interrupt flag (PSPIF).** If set, a read or a write operation has taken place. This condition must be cleared in software. If clear, no read or write has occurred.

PIC16C64 Register Description *(continued)*

- **Timer1 (OEh, OFh: TMR1L, TMR1H).** These are the low and high bytes that comprise the 16-bit Timer1. The clock source can be either internal (OSC/4) or external. When the clock source is external, TMR1 can run either synchronously or asynchronously. In asynchronous mode, TMR1 can run during sleep. When used with CCP1, TMR1 functions synchronously as a time base for 16-bit compare or capture.

- **Timer1 Control Register (10h: T1CON).**

Bit	Label Name and Function
0	**Timer1 run bit (TMR1ON).** Set enables TMR1. Clear stops TMR1.
1	**Timer1 clock select (TMR1CS).** If set, use external clock, low-to-high transition, through TCKI pin. If clear, use internal clock (OSC/4).
2	**Timer1 external clock input synchronization control (T1INSYNC).** If TMR1CS is set, then if T1INSYNC is set , do not synchronize external clock input, otherwise if clear, synchronize external clock input. If TMR1CS is clear, then T1INSYNC is ignored.
3	**Timer1 oscillator enable/disable (T1OSCEN).** If set, then the oscillator is enabled. If clear, the oscillator is shut off, reducing power use.
4-5	**Timer1 input clock prescale select register (T1CKPS0, T1CKPS1).**

Bits 5-4	Prescale Value
11	8
10	4
01	2
00	1

6-7	Unimplemented and read as “0”.

PIC16C64 Register Description *(continued)*

- **Timer2 (11h: TMR2).** TMR2 is an 8-bit timer with a programmable prescalar and postscalar (programmable rates are: 1:1, 1:4, 1:16), and an 8-bit period register (PR2). With the postscalar, TMR2 can match PR2 a specified number of times before generating an interrupt. TMR2 can also be used with Capture/Compare/PWM and with the synchronous serial port.

- **TMR2 Control Register (12h: T2CON).**

Bit	Label Name and Function
0-1	**Timer2 clock prescaler select (T2CKPS0, T2CKPS1).**

Bits 1-0	Prescale Value
11	16
10	16
01	4
00	1

2	**Timer2 on/off control (TMR2ON).** When set, TMR2 is on. When clear, TMR2 is off.
3-6	**Timer2 output postscale select (TOUTPS0, TOUTPS1, TOUTPS2, TOUTPS3).** The 4-bit postscaler range is 0000b - 1111b (0-15).
7	Unimplemented and read as "0".

- **Synchronous Serial Port Buffer Register (13h: SSPBUF).** The 8-bit SSPBUF holds data transferred from the synchronous serial port shift register (SSPSR).

PIC16C64 Register Description *(continued)*

- **Synchronous Serial Port Control Register (14h: SSPCON).**

Bit	Label Name and Function
0-3	**Synchronous serial port mode select (SSPM0, SSPM1, SSPM2, SSPM3):**

Bits 3-0	Mode
0000	SPI master mode, clock=OSC/4
0001	SPI master mode, clock=OSC/16
0010	SPI master mode, clock=OSC/64
0011	SPI master mode, clock=TMR2 output/2
0100	SPI slave mode, clock=SCK pin. $\overline{SS}$ pin control enabled.
0101	SPI slave mode, clock=SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ is available for I/O.
0110	I^2C slave mode, 7-bit address.
0111	I^2C slave mode, 10-bit address.
1011	I^2C master mode support enabled (slave idle).
1110	I^2C slave mode, 7-bit address with master mode support enabled.
1111	I^2C slave mode, 10-bit address with master mode support enabled.

4 **Clock polarity select (CKP).**

In SPI mode: If this bit is set, the transmit occurs on high-to-low transitions, and the clock's idle state is high. If clear, the transmit occurs on low-to-high transitions, and the clock's idle state is low.

In I^2C mode: This bit is for SCK release control. If set, the clock is enabled. If clear, the clock is stretched by holding it low. (This ensures enough time to set up data.)

PIC16C64 Register Description *(continued)*

5 **Synchronous serial port enable (SSPEN).** In either SPI or I²C modes, the affected pins must be configured as input or output.

In SPI mode: If this bit is set, enables serial port and configures SCK, SDO and SDI as serial port pins.

In I²C mode: If set, this bit enables the serial port and configures SDA and SCL pins as serial port pins. If clear, this bit disables the serial port, and configures the SDA and SCL pins as I/O ports.

6 **Receive overflow flag (SSPOV).** A new byte is received while SSPBUF still contains previous data. In either SPI or I²C modes, SSPOV must be cleared in software.

In SPI mode: If this bit is true in slave mode, overflow has occurred, and the data in SSPSR is lost. SSPBUF must be read to avoid overflow, both when transmitting and receiving. In master mode, no overflow occurs because each transmission or reception is started by writing to SSPBUF.

In I²C mode: If true, an overflow has occurred when receiving data. When transmitting data, SSPOV doesn't matter.

7 **Write collision detect (WCOL).** If true, the SSPBUF register is written to while still transmitting. This flag must be cleared in software. If clear, no collision has occurred.

- **Compare Register (15h, 16h: CCPR1L, CCPR1H).** The compare register consists of two bytes, CCPR1L and CCPR1H. CCPR1 is used with capture, compare, and pwm modes.

PIC16C64 Register Description *(continued)*

Capture Mode: The RC2/CCP1 pin should be configured for input through its TRIS register. Then CCPR1 captures TMR1 when a state transition occurs on the RC2/CCP1 pin. The desired state transition (high-to-low or low-to-high) is set with the CCP1M bits in the CCP1CON register, described below. If a capture occurs before a previous capture is read, the previous capture is overwritten. When a capture occurs, the interrupt request flag CCP1IF in the PIR register is set, which must be cleared by software. When the capture mode is changed, a capture interrupt may be raised, so CCP1IF should be cleared by the user after switching modes.

Compare Mode: The CCPR1 register becomes a 16-bit period register for TMR1 when compare mode is selected.

PWM Mode: In PWM mode, the user writes the 8-bit duty cycle into the low byte (CCPR1L). The high byte (CCPR1H) becomes the slave buffer to CCPR1L. When the PWM1 output is set, CCPR1L is transferred to CCPR1H. Clearing the CPP1CON register forces the RC2/CCP1 PWM output latch low. This is not the I/O data latch.

- **Capture/Compare/PWM Control Register (17h: CCP1CON).**

Bit	Label Name and Function
0-3	**CCP1 mode select (CCP1M0, CCP1M1, CCP1M2, CCP1M3).**

Bits 3-0	Mode
0000	Capture/Compare/PWM off
0100	Capture mode, high-to-low transition
0101	Capture mode, low-to-high transition
0110	Capture mode, every 4th low-to-high transition
0111	Capture mode, every 16th low-to-high transition

PIC16C64 Register Description *(continued)*

Bits 3-0	Mode
1000	Compare mode, set output on match
1001	Compare mode, clear output on match
1010	Compare mode, raise software interrupt, setting CCP1IF bit.
1011	Compare mode, trigger special event
11??	PWM mode

4-5 **Two low-order bits (CCP1X, CCP1Y).** Used only for PWM. Write these two bits in high resolution (10-bit) mode, or keep them constant at "0" when only 8-bit resolution is desired.

6-7 Unimplemented, read as "0".

- **General Purpose Registers (20h - 7Fh).** These registers are for use by your program.
- **Option Register (81h):** This register sets prescaler options, TMR0 settings, external interrupt trigger edge, and PORTB pull-up status. The function of each bit is shown below:

Bit	Label Name and Function
0-2	**Prescaler ratio (PS0, PS1, PS2).** These 3 bits determine the prescaler input-to-output ratio. When using the prescaler with TMR0, the seven possible ratios are 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256. When using the prescaler with the watchdog timer, the ratios are 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128. For example, assign the prescaler to the watchdog timer. To increase the watchdog time-out period to 64 times its normal length, set the prescaler ratio to 110b; this yields a watchdog period of approximately 1 second (64 x 18 ms).

PIC16C64 Register Description *(continued)*

Bit	Label Name and Function
3	**Prescaler assignment (PSA).** This bit determines whether the prescaler is assigned to TMR0 (“0”) or to the watchdog timer (“1”).
4	**Timer0 trigger edge (RTE).** This bit determines whether TMR0 increments on a low-to-high (“0”) or high-to-low (“1”) transition on the T0CKI pin.
5	**Timer0 signal source (RTS).** This bit determines whether TMR0 is driven by the PIC's internal instruction clock (“0”) or by the T0CKI pin (“1”).
6	**External interrupt trigger edge (INTEDG).** Determines whether an interrupt will be caused by a high-to-low (“0”) or low-to-high (“1”) transition on the external INT pin.
7	**PORTB pull-up status (RBPU).** If this bit is clear, PORTB pull-up resistors are enabled on pins that are inputs. If this bit is set, all Port B pull-ups are disabled.

- **Program Counter Low Byte (82h: PCL).** Same as register 02h.
- **Status Register (83h).** Same as register 03h.
- **File Select Register (84h: FSR).** Same as register 04h.
- **Data Direction Register for Port A (85h: TRISA).** This is the data direction register for Port A. In the Microchip data book, this register is referred to as “Tri-State A”, hence “TRISA” as an abbreviation. Data direction registers in the PIC16C64 are addressable, unlike their counterparts in PIC16C5x devices.

  Bits in this register which are set to “1” cause the corresponding bits in Port A to become inputs (the pins go into high impedance mode, allowing them to be driven by an external source). Bits which are cleared to “0” cause the corresponding bits in Port A to become outputs.

PIC16C64 Register Description *(continued)*

- **Data Direction Register for Port B (86h: TRISB).** This is the data direction register for Port B. The bits are set similarly to TRISA.
- **Data Direction Register for Port C (87h: TRISC).** This is the data direction register for Port C. The bits are set similarly to TRISA.
- **Data Direction Register for Port D (88h: TRISD).** This is the data direction register for Port D. The bits are set similarly to TRISA.
- **Data Direction Register or Parallel Slave Port Mode Register for Port E (89h: TRISE).** The following table describes the TRISE register:

Bit	Label Name and Function
0	**Data Direction Bit (TRISE0).** This bit is set similarly to TRISA when Port E is used for general-purpose I/O. When Port E is used for parallel slave port control (PSPMODE), This bit must be set.
1	**Data Direction Bit (TRISE1).** This bit is set similarly to TRISA when Port E is used for general-purpose I/O. When Port E is used for parallel slave port control (PSPMODE), this bit must be set.
2	**Data Direction Bit (TRISE2).** This bit is set similarly to TRISA when Port E is used for general-purpose I/O. When Port E is used for parallel slave port control (PSPMODE), this bit must be set.
3	Not used. Read as “0”.
4	**Parallel Slave Port Mode (PSPMODE).** This bit selects the parallel slave port mode for ports A and E when set, and bits 0 - 2 (above) must be set. When clear, Port E is a general-purpose I/O port.

PIC16C64 Register Description *(continued)*

Bit	Label Name and Function
5	**Input Buffer Overflow (IBOV).** This bit indicates whether or not an input buffer overflow has occurred. If set, a write occurred before previous input was read. This bit must be cleared in software. If clear, overflow has not occurred.
6	**Output Buffer Full (OBF).** This bit indicates whether or not the output buffer is full. If set, the output buffer holds a previously written word. If clear, the output buffer has been read.
7	**Input Buffer Full (IBF).** This bit indicates whether or not the input buffer is full. If set, a word has been received and is waiting to be read by the CPU. If clear, no word has been received.

- **Program Counter Latch High (8Ah: PCLATH).** Same as register 0Ah.
- **Interrupt Control Register (8Bh: INTCON).** Same as register 0Bh.
- **Enable Bits for Peripheral Interrupts (8Ch: PIE1):**

Bit	Label Name and Function
0	**Timer1 interrupt enable bit (TMR1IE).** If set, enables TMR1IF interrupt. If clear, disables TMR1IF interrupt.
1	**Timer2 interrupt enable bit (TMR2IE).** If set, enables TMR2IF interrupt. If clear, disables TMR2IF interrupt.
2	**CCP1 interrupt enable bit (CCP1IE).** If set, enables CCP1IF interrupt. If clear, disables CCP1IF interrupt.
3	**Synchronous serial port interrupt enable bit (SSPIE).** If set, enables SSPIF interrupt. If clear, disables SSPIF interrupt.

PIC16C64 Register Description *(continued)*

Bit	Label Name and Function
4-6	Unimplemented and read as "0".
7	**Parallel slave port interrupt enable (PSPIE).** If set, enables PSPIF interrupt. If clear, disables interrupt.

- **PCON Register (8Eh):** Flags for differentiating between a power-on reset, an external MCLR reset, and a WDT reset:

Bit	Label Name and Function
0	Reserved. Should be programmed as "1".
1	**Power-on-reset flag (POR).** If set, no power-on-reset. If clear, a power-on-reset has occurred. Must be set by software after a power-on-reset.
2-7	Unimplemented. Read as "0".

- **Timer2 Period Register (92h: PR2).** Read/write register for setting period of TMR2.
- **SSP (I^2C) Slave Address Register (93h: SSPADD).** In 10-bit mode, write the high byte of the address (1111 0 A9 A8 0). After the high byte address match, load the low byte address (A7 - A0).
- **Synchronous Serial Port Status Register (94h: SSPSTAT):**

Bit	Label Name and Function
0	**Buffer full (BF).** *Receive Mode:* If set, reception is complete and SSPBUF is full. If clear, reception is not complete, and SSPBUF is empty. *Transmit Mode:* If set, transmission is occurring, and SSPBUF is full. If clear, transmission is complete and SSPBUF is empty.

PIC16C64 Register Description *(continued)*

Bit	Label Name and Function
1	**Update address (UA).** 10-bit I²C slave mode only. If set, the user needs to update the address in the SSPADD register. If clear, the address is current.
2	**Read/Write bit information (R_W).** I²C mode only. This bit holds the R/W bit information received following the last address match. This bit is only valid during the transmission, and can be used to determine whether transmission or reception is in progress. If set, a read is in progress. If clear, a write is in progress.
3	**Start bit (S).** I²C mode only. Cleared when the SSP module is disabled (SSPEN cleared). If set, indicates that a start bit has been detected last. This bit is clear on reset. If clear, the start bit was not detected last.
4	**Stop bit (P).** I²C mode only. Cleared when the SSP module is disabled (SSPEN is cleared). If set, indicates that a stop bit was detected last. If clear, a stop bit was not detected last.
5	**Data/Address bit (D_A).** I²C mode only. If set, indicates that the last byte received was data. If clear, indicates that the last byte received was an address.
6-7	Unimplemented and read as "0".

- **General Purpose Registers (A0h - BFh).** These registers are for use by your program.

PIC16C64

Where to Find Additional information

Many of the features in the PIC16C64 are common to the earlier PIC16C5x family. Refer to the PIC16C5x section earlier in this handbook for PIC16C5x device information.

The following topics are covered in the PIC16C64 databook from Microchip Technology:

- Architectural Overview
- I/O Programming Considerations: Bidirectional I/O ports and Parallel Slave Port
- Timer Modules and Prescaler
- Capture/Compare/PWM Modes
- Synchronous Serial Port: SPI and I^2C
- Special CPU Features: fuses, oscillators, resets, and interrupts
- Electrical and Timing Characteristics

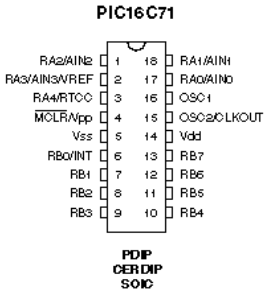
BLANK PAGE

BLANK PAGE

PIC16C71

PIC16C71 Pin-Out

The following diagram shows the PIC16C71 pin-out:



Pin	Function
RA0 - RA4	I/O Port A
RB0 - RB7	I/O Port B
AIN0 - AIN3	Analog inputs
VREF	External reference for A/D
INT	External interrupt input
RTCC	Real-time clock/counter input
MCLR	Master clear (reset)
OSC1	Oscillator input
OSC2/CLKOUT	Oscillator output (OSC/4)
Vdd	Power supply
Vss	Ground
Vpp	Programming voltage input

PIC16C71

PIC16C71 Microcontrollers

The table below shows the various PIC16C71's available:

PART #	ERASE	(E)PROM	RAM	I/O	SUPPLY	OSC.	FREQ.
PIC16C71-04	No	1K x 14	36 x 8	13	4.0 - 6.0 V	RC,XTAL	DC - 4 MHz
PIC16C71-16	No	1K x 14	36 x 8	13	4.0 - 6.0 V	RC,XTAL	DC - 16 MHz
PIC16C71/JW	Yes	1K x 14	36 x 8	13	4.0 - 6.0 V	RC,XTAL	DC - 16 MHz
PIC16LC71-04	No	1K x 14	36 x 8	13	3.0 - 6.0 V	RC,XTAL	DC - 4 MHz

New and Modified Features

Although the PIC16C71 is much like devices in the PIC16C5x family, it has a number of important differences:

- **Interrupts possible from four sources:** external pin, RTCC timer, A/D conversion, and change on four Port B pins.
- **Port B modifications.** Software controlled pull-up's have been added, along with the ability to generate an interrupt when any of four pins changes state.
- **8-bit analog-to-digital convertor.** Four channel multiplexed A/D convertor has 20µS conversion time, ±1 LSB accuracy, and built-in sample and hold.
- **8-level hardware stack.** Allows deeper nesting of subroutines.
- **14-bit instruction word.** Provides larger page sizes for program memory (2K) and RAM (128 bytes).
- **New timers.** Two new timers have been added to avoid unnecessary delays on power-up and wake-up. These timers are the oscillator start-up timer (OST) and power-up timer (PWRT).
- **New I/O pin.** An additional I/O pin has been added as bit 4 of Port A. This is physically implemented on pin 3 (RTCC/RA4).
- **Status register changes.** Program page select bits (bit 5-6) have been replaced by *register* page select bits.

New and Modified Features *(continued)*

- **File select register changes.** The FSR has been increased to 8 bits, and bits 5-6 no longer function as register page select (register page selection is now done in the Status register).
- **High byte added to PC.** A high byte has been added to the program counter to handle program memory paging.
- **Page select bits removed.** Program memory page select bits in the Status register (PA0 - PA2) have been removed.

  The part's 1K of program space can be utilized without worrying about page boundaries.
- **Register 07h is unimplemented.** This register cannot be used for storage, as it can in 18-pin PIC16C5x devices.
- **Reset vector changed to 0000h.**
- **Interrupt vector added at 0004h.**
- **Four new instructions have been added to the assembler:**

Microchip Instruction	Parallax Equivalent
sublw	**mov w,#literal-w** (move #literal-W into W)
addlw	**add w,#literal** (add #literal to W)
retfie	**reti** (return from interrupt)
return	**ret** (return from subroutine, *without affecting W*)

New and Modified Features *(continued)*

- **Five directives & instructions have been removed:**

Microchip Instruction	Parallax Equivalent
-	**ljmp, lcall, lset** (the PIC16C71 does not have paged program memory, so long linstructions are not necessary)
-	**reset** (in the '71, the reset vector has been moved to address 0000h, so the reset directive was dropped. To have your program start at another location, you must insert a jmp at 0000h)
-	**addb fr,bit** (add bit into file register)
-	**subb fr,bit** (subtract bit from file register)
-	**neg fr** (negate file register)

PIC16C71 Registers

The following table shows the various registers in the PIC16C71; the function of each register is described in the following pages.

Register	Function
-	W register
00h	Indirect addressing register
01h	Real-time clock/counter (RTCC)
02h	Program counter low byte (PCL)
-	Stack registers (8)
03h	Status register
04h	File select register (FSR)
05h	I/O Port A
06h	I/O Port B
07h	Not implemented
08h	A/D control & status register (ADCON0)
09h	A/D result (ADRES)
0Ah	Program counter latch high (PCLATH)
0Bh	Interrupt control register (INTCON)
0Ch - 2Fh	General purpose registers
30h - 7Fh	Not implemented
	End of page 0 register memory
80h	Indirect addressing register
81h	Option register
82h	Program counter low byte (PCL)
83h	Status register
84h	File select register (FSR)
85h	TRISA
86h	TRISB
87h	Not implemented
88h	A/D control register (ADCON1)
89h	A/D result (ADRES)
8Ah	Program counter latch high (PCLATH)
8Bh	Interrupt control register (INTCON)
8Ch - AFh	Reads/writes registers 0Ch - 2Fh
B0h - FFh	Not implemented

PIC16C71 Registers *(continued)*

Quick reference register map:

	PAGE 0	PAGE 1	
	W register	W register	
	Stack (8)	Stack (8)	
00h	Indirect addr.	Indirect addr.	80h
01h	RTCC	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORT A	TRISA	85h
06h	PORT B	TRISB	86h
07h			87h
08h	ADCON0	ADCON1	88h
09h	ADRES	ADRES	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	36 general purpose registers (RAM)	reads/writes registers 0Ch - 2Fh	8Ch
2Fh			AFh
30h			B0h
7Fh			FFh

PIC16C71 Registers *(continued)*

The following text describes the function of each *new or modified* register. Refer to the PIC16C5x section for registers not shown here.

For some of the registers, you'll notice the designation "xxh" following the register name. This indicates the address of the register. Registers with no address cannot be addressed directly.

- **Program Counter Low Byte (02h: PCL).** The program counter holds the address for the instruction currently being executed. The program counter and its associated eight-level stack are 13 bits wide, with PCL holding the lower 8 bits and PCH holding the upper 5 bits. PCH is automatically loaded when a JMP or CALL is performed. Although PCH is not addressable, your program can load PCH through the PCLATH register (see register f0A, later in this section).
- **Stack.** The stack is comprised of eight registers which are used for calling and returning from subroutines. The program counter is pushed onto the stack when a CALL is executed or an interrupt is acknowledged. The stack is popped in the event of a RET, RETW, or RETI instruction.
- **Status Register (03h).** This register contains the status of the PIC's arithmetic logic unit (ALU), the reset status, and the page select bits for *register* memory (not program memory, as in PIC16C5x devices).

  The function of each bit in the status register is shown below:

Bit	Function
0	**Carry bit (C).** Set if an addition or subtraction causes an overflow from the most significant bit of the resultant (bit 7). Subtraction is included because it's executed by adding the two's complement. Also used by rotate instructions, which rotate the contents of a register and copy the low or high order bit of the register into the carry bit.

PIC16C71 Registers *(continued)*

Bit	Function
1	**Digit carry bit (DC).** Set if an addition or subtraction causes an overflow from the 4th low order bit (bit 3). Digit carry indicates that more than one hex digit (4 bits) was necessary to accommodate the result.
2	**Zero bit (Z).** Set if the result of an arithmetic or logic operation is zero.
3	**Power-down bit (PD).** Set during power-up or by a CLR WDT (clear watchdog) instruction. Cleared by a SLEEP instruction.
4	**Time-out bit (TO).** Set during power-up, by CLR WDT, or by SLEEP. Cleared by a watchdog timer time-out.
5-6	**Register page select bits for direct addressing (RP0, RP1).** These bits determine which register page is selected for direct addressing operations. Each page is 128 bytes long, so only RP0 is valid in a PIC16C71. RP1 can be used for storage, but may have an actual use in future PIC's.
7	**Register page select bit for indirect addressing (IRP).** This bit determines which register page is selected for indirect addressing operations. Since each "indirect" page is 256 bytes long, however, it is not valid in the PIC16C71. IRP can be used for storage, but may have an actual use in future PIC's.

PIC16C71 Registers *(continued)*

The following table shows how various events affect the power-down and time-out bits:

Event	PD	TO
Power-up	1	1
Watchdog time-out	x	0
SLEEP instruction	0	1
CLR WDT instruction	1	1

Lastly, this table shows the status of the power-down and time-out bits after a reset:

Cause of Reset	PD	TO
Watchdog time-out (not during sleep)	1	0
Watchdog time-out (during sleep)	0	0
External reset (not during sleep)	x	x
External reset (during sleep)	0	1
Normal power-up	1	1

- **File Select Register (04h: FSR).** This register selects the register for indirect addressing. Bits 0-7 select 1 of 256 registers in the current bank (the '71 only has one bank). As described earlier, a read or write to register 00h will access the register pointed to by the contents of the FSR.
- **I/O Port A (05h).** 5-bit I/O port. This register is used to read and write I/O Port A. It can be read and written just as any other register. However, read instructions always read the I/O pins, regardless of whether the pins are programmed as inputs or outputs.

  Bits 0-3 (pins 17, 18, 1 & 2) share their pins with analog inputs AIN0 - AIN3. Bit 3 further shares its pin with the analog reference input.

  Bit 4 (pin 3) has an open-collector output and shares its pin with the RTCC input.

PIC16C71 Registers *(continued)*

Two bits in ADCON1 (f88) determine whether bits 0-3 are digital or analog pins. Upon power-on reset, bits 0-3 are configured as analog inputs.

- **I/O Port B (06h).** 8-bit I/O port. Each of the Port B pins has a weak internal pull-up resistor (~250 µA). A pin's pull-up is turned off if the pin is configured as an output, and a single bit in the Option register can turn off all the pull-ups. The pull-up resistors are disabled on power-on reset.

  On bits 4-7 (pins 10-13), Port B has an interrupt on change feature that can generate an interrupt if any of the pins changes state. Any pin configured as an output is excluded.

  This interrupt can wake up the chip from sleep. Along with the internal pull-ups, the interrupt from sleep feature makes it easy to interface to a keypad and have wake-up on key press.

- **A/D Control and Status Register (08h: ADCON0).** This register controls the PIC16C71's analog-to-digital convertor; it also reports the status of the A/D. The functions of the various bits are shown below:

Bit	Function
0	**A/D enable.** This bit determines whether or not the A/D convertor is on or off. When turned off, the A/D convertor consumes no current.
1	**A/D interrupt.** Set if an interrupt was the result of an A/D conversion. Must be cleared by software.
2	**Go/done bit.** Must be set to "1" to start a conversion. Cleared by hardware when conversion is complete.
3-4	**Analog channel select.** These bits determine which input pin will be used for the A/D convertor. "00" selects AIN0, "01" selects AIN1, "10" selects AIN2, and "11" selects AIN3.
5	**General purpose bit.** Can be used for storage.

PIC16C71 Registers *(continued)*

Bit	Function
6-7	**A/D conversion clock source.** These bits determine the clock source for the A/D convertor. The clock can be derived from either the OSC1 input or from the A/D's on-chip RC oscillator, as follows:

Bits	Clock Source
00	2 x Tosc
01	8 x Tosc
10	32 x Tosc
11	RC oscillator (2-6 µs, 4 µs typical)

Whatever source you choose, each clock period must be at least 2 µs. Thus, if you're using a fast crystal for OSC1, you will need to set the A/D clock source bits to "01" or "10". And if the OSC1 clock is 20 MHz, then you would only have the option of using the A/D's RC oscillator, since 20 MHz cannot be divided down far enough to give the minimum 2 µs necessary.

The A/D's RC oscillator can vary considerably with voltage and temperature (2-6 µs).

Total conversion time for a single sample is ten clock periods (20 µs minimum).

- **A/D Conversion Result (09h: ADRES).** This register holds the result of the most recent conversion.
- **Program Counter Latch High (0Ah: PCLATH).** This register is used to access the 5 high bits of the program counter (PCH). Unlike the lower 8 bits (PCL), the 5 high bits are not directly addresable. Instead, they are stored in PCLATH for later use. When the program counter is loaded with a new value during a JMP, CALL, or a write to PCL, the high bits are loaded from PCLATH.

PIC16C71 Registers *(continued)*

- **Interrupt Control Register (0Bh: INTCON).** This register is used to enable interrupts and to determine what caused an interrupt. The function of each bit is given below:

Bit	Function
0	**Port B interrupt.** This bit is set if an interrupt was the result of a transition on any of the upper four bits of Port B. It will remain set until cleared by software.
1	**External interrupt.** Set if an interrupt was caused by a transition on the external INT pin. Must be cleared by software.
2	**RTCC overflow interrupt.** Set if an interrupt was caused by an overflow in the real-time clock/counter. Must be cleared by software.
3	**Port B interrupt enable.** Determines whether the Port B interrupt is enabled (“1”) or disabled (“0”).
4	**External interrupt enable.** Determines whether the external interrupt is enabled (“1”) or disabled (“0”).
5	**RTCC interrupt enable.** Determines whether the RTCC interrupt is enabled (“1”) or disabled (“0”).
6	**A/D interrupt enable.** Determines whether the A/D conversion complete interrupt is enabled (“1”) or disabled (“0”).
7	**Global interrupt enable.** Clearing this bit disables all interrupts. Setting this bit allows all interrupts that are individually enabled in bits 3-6.

- **Indirect Addressing Register (80h).** This register has the same function as register 00h.

PIC Registers *(continued)*

- **Option register (81h: OPTION).** This register is used to set prescaler options, RTCC settings, external interrupt trigger edge, and Port B pull-up status. The function of each bit is shown below:

Bit	Function
0-2	**Prescaler ratio.** These 3 bits determine the prescaler input-to-output ratio. When using the prescaler with the RTCC, the seven possible ratios are 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256. When using the prescaler with the watchdog timer, the ratios are 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128. For example, let's say that the prescaler is assigned to the watchdog timer. To increase the watchdog time-out period to 64 times its normal length, the prescaler ratio would be set to 110b; this yields a watchdog period of approx. 1 second (64 x 18 ms).
3	**Prescaler assignment.** This bit determines whether the prescaler is assigned to the RTCC ("0") or to the watchdog timer ("1").
4	**RTCC trigger edge.** This bit determines whether the RTCC increments on a low-to-high ("0") or high-to-low ("1") transition on the RTCC pin.
5	**RTCC signal source.** This bit determines whether the RTCC is driven by the PIC's internal instruction clock ("0") or by the RTCC pin ("1").
6	**External interrupt trigger edge.** Determines whether an interrupt will be caused by a high-to-low ("0") or low-to-high ("1") transition on the external INT pin.
7	**Port B pull-up status.** If this bit is clear, Port B pull-up resistors are enabled on pins that are inputs. If this bit is set, all Port B pull-ups are disabled.

PIC16C71 Registers *(continued)*

- **Program Counter Low Byte (82h: PCL).** Same as register 02h.
- **Status Register (83h).** Same as register 03h.
- **File Select Register (84h: FSR).** Same as register 04h.
- **Data Direction Register for Port A (85h: TRISA).** This is the data direction register for Port A. In the Microchip data book, this register is referred to as "Tri-State A", hence "TRISA" as an abbreviation.

  Bits in this register which are set to "1" cause the corresponding bits in Port A to become inputs (the pins go into high impedance mode, allowing them to be driven by an external source). Bits which are cleared to "0" cause the corresponding bits in Port A to become outputs.
- **Data Direction Register for Port B (86h: TRISB).** This is the data direction register for Port B.

  Data direction registers in the PIC16C71 are addressable, unlike their counterparts in PIC16C5x devices.
- **A/D Control Register (88h: ADCON1).** This register controls which pins will be used as analog inputs and whether the A/D's reference will come from Vdd or from the external VREF pin. The bits that control these functions are shown below:

Bits 1,0	RA0	RA1	RA2	RA3	Ref.
0 0	analog	analog	analog	analog	Vdd
0 1	analog	analog	analog	ref.	RA3
1 0	analog	analog	digital	digital	Vdd
1 1	digital	digital	digital	digital	-

- **A/D Conversion Result (89h: ADRES).** Same as register 09h.
- **Program Counter Latch High (8Ah).** Same as register 0Ah.
- **Interrupt Control Register (8Bh: INTCON).** Same as register 0Bh.

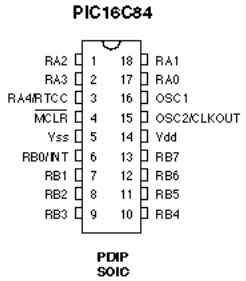
BLANK PAGE

BLANK PAGE

PIC16C84

PIC16C84 Pin-Out

The following diagram shows the PIC16C84 pin-out:



Pin	Function
RA0 - RA4	I/O Port A
RB0 - RB7	I/O Port B
INT	External interrupt input
RTCC	Real-time clock/counter input
$\overline{\text{MCLR}}$	Master clear (reset)
OSC1	Oscillator input
OSC2/CLKOUT	Oscillator output (OSC/4)
Vdd	Power supply
Vss	Ground

PIC16C84

PIC16C84 Microcontrollers

The table below shows the various PIC16C84's available:

Part #	Erasable	Program	Registers	I/O	Power	Osc. Type	Frequency
PIC16C84-04	Yes	1K x 14	100 x 8*	13	4.0 - 6.0 V	RC,XTAL	DC - 4 MHz
PIC16C84-10	Yes	1K x 14	100 x 8*	13	4.0 - 6.0 V	RC,XTAL	DC - 10 MHz
PIC16LC84-04	Yes	1K x 14	100 x 8*	13	2.0 - 6.0 V	RC,XTAL	DC - 4 MHz

* 36 registers in RAM, 64 registers in EEPROM.

New and Modified Features

The PIC16C84 shares many features with the older PIC16C5x and PIC16C71 devices. In fact, with a few notable exceptions, the '84 is a virtual copy of the '71. The following text describes the important differences between the '84 and the original '5x series:

- **1K program space and 64 registers implemented in EEPROM**.
- **Interrupts possible from four sources**: external pin, RTCC timer, EEPROM write complete, and change on four Port B pins.
- **Port B modifications**. Software controlled pull-up's have been added, along with the ability to generate an interrupt when any of four pins changes state.
- **8-level hardware stack**. Allows deeper nesting of subroutines.
- **14-bit instruction word**. Provides larger page sizes for program memory (2K) and RAM (128 bytes).
- **New timers**. Two new timers have been added to control delays on power-up and wake-up. These timers are the oscillator start-up timer (OST) and power-up timer (PWRT).
- **New I/O pin**. An additional I/O pin has been added as bit 4 of Port A. This is physically implemented on pin 3 (RTCC/RA4).
- **Status register changes**. Program page select bits (bit 5-6) have been replaced by *register* page select bits.

New and Modified Features *(continued)*

- **File select register changes**. The FSR has been increased to 8 bits, and bits 5-6 no longer function as register page select (register page selection is now done in the Status register).
- **High byte added to PC**. A high byte has been added to the program counter to handle program memory paging.
- **Page select bits removed**. Program memory page select bits in the Status register (PA0 - PA2) have been removed.

  The part's 1K of program space can be utilized without worrying about page boundaries.
- **Register 07h is unimplemented**. This register cannot be used for storage, as it can in 18-pin PIC16C5x devices.
- **Reset vector moved to 0000h**.
- **Interrupt vector added at 0004h.**
- **Six directives & instructions have been added to the assembler**:

Microchip Instruction	Parallax Equivalent
-	**pwrt_on, pwrt_off** Used in the device directive to control the 84's power-up timer.
sublw	**mov w,#literal-w** Move #literal-w into w.
addlw	**add w,#literal** Add #literal to w.
return	**ret** Return, *without affecting w*.
retfie	**reti** Return from interrupt.
option	**mov !option,data** Load option register with data.

New and Modified Features *(continued)*

- **Five directives & instructions have been removed**:

Microchip Instruction	Parallax Equivalent
-	**reset** In the '84, the reset vector has been moved to address 0000h, so the reset directive was dropped. The reset vector is now set up by establishing a jmp at 0000h to the start of your program; address 0005h is the first possible location for your program.
-	**ljmp, lcall, lset** The '84 does not have paged program memory, so long linstructions are unnecessary.
-	**addb fr,bit** Add bit into file register.
-	**subb fr,bit** Subtract bit from file register.
-	**neg fr** Negate file register.

PIC16C84 Registers

The following table shows the various registers in the PIC16C84; the function of each register is described in the following pages.

Register	Function
-	W register
00h	Indirect addressing register
01h	Real-time clock/counter (RTCC)
02h	Program counter low byte (PCL)
-	Stack registers (8)
03h	Status register
04h	File select register (FSR)
05h	I/O Port A
06h	I/O Port B
07h	Not implemented
08h	EEPROM data register (EEDATA)
09h	EEPROM address register (EEADR)
0Ah	Program counter latch high (PCLATH)
0Bh	Interrupt control register (INTCON)
0Ch - 2Fh	General purpose registers
30h - 7Fh	Not implemented
	End of page 0 register memory
80h	Indirect addressing register
81h	Option register
82h	Program counter low byte (PCL)
83h	Status register
84h	File select register (FSR)
85h	TRISA
86h	TRISB
87h	Not implemented
88h	EEPROM control register 1 (EECON1)
89h	EEPROM control register 2 (EECON2)
8Ah	Program counter latch high (PCLATH)
8Bh	Interrupt control register (INTCON)
8Ch - AFh	Reads/writes registers 0Ch - 2Fh
B0h - FFh	Not implemented

PIC16C84 Registers *(continued)*

Quick reference register map:

	PAGE 0	PAGE 1	
	W register	W register	
	Stack (8)	Stack (8)	
00h	Indirect addr.	Indirect addr.	80h
01h	RTCC	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORT A	TRISA	85h
06h	PORT B	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	36 general purpose registers (RAM)	reads/writes registers 0Ch - 2Fh	8Ch
2Fh			AFh
30h			B0h
7Fh			FFh

PIC16C84 Registers *(continued)*

The following text describes the function of each *new or modified* register. Refer to the PIC16C5x section for registers not shown here.

For some of the registers, you'll notice the designation "xxh" following the register name. This indicates the address of the register. Registers with no address cannot be addressed directly.

- **Program Counter Low Byte (02h: PCL).** The program counter holds the address for the instruction currently being executed. The program counter and its associated eight-level stack are 13 bits wide, with PCL holding the lower 8 bits and PCH holding the upper 5 bits. PCH is automatically loaded when a JMP or CALL is performed. Although PCH is not addressable, your program can load PCH through the PCLATH register (see register f0A, later in this section).
- **Stack.** The stack is comprised of eight registers which are used for calling and returning from subroutines. The program counter is pushed onto the stack when a CALL is executed or an interrupt is acknowledged. The stack is popped in the event of a RET, RETW, or RETI instruction.
- **Status Register (03h).** This register contains the status of the arithmetic logic unit (ALU), the reset status, and the page select bits for *register* memory (not program memory, as in PIC16C5x devices).

  The function of each bit in the status register is shown below:

Bit	Function
0	**Carry bit (C).** Set if an addition or subtraction causes an overflow from the most significant bit of the resultant (bit 7). Subtraction is included because it's executed by adding the two's complement. Also used by rotate instructions, which rotate the contents of a register and copy the low or high order bit of the register into the carry bit.

PIC16C84 Registers *(continued)*

Bit	Function
1	**Digit carry bit (DC).** Set if an addition or subtraction causes an overflow from the 4th low order bit (bit 3). Digit carry indicates that more than one hex digit (4 bits) was necessary to accommodate the result.
2	**Zero bit (Z).** Set if the result of an arithmetic or logic operation is zero.
3	**Power-down bit (PD).** Set during power-up or by a CLR WDT (clear watchdog) instruction. Cleared by a SLEEP instruction.
4	**Time-out bit (TO).** Set during power-up, by CLR WDT, or by SLEEP. Cleared by a watchdog timer time-out.
5-6	**Register page select bits for direct addressing (RP0, RP1).** These bits determine which register page is selected for direct addressing operations. Each page is 128 bytes long, so only RP0 is valid in a PIC16C71. RP1 can be used for storage, but may have an actual use in future PIC's.
7	**Register page select bit for indirect addressing (IRP).** This bit determines which register page is selected for indirect addressing operations. Since each indirect page is 256 bytes long, however, it is not useful in the PIC16C84. IRP can be used for storage, but may have an actual use in future PIC's.

PIC16C84 Registers *(continued)*

The following table shows how various events affect the power-down and time-out bits:

Event	PD	TO
Power-up	1	1
Watchdog time-out	x	0
SLEEP instruction	0	1
CLR WDT instruction	1	1

Lastly, this table shows the status of the power-down and time-out bits after a reset:

Cause of Reset	PD	TO
Watchdog time-out (not during sleep)	1	0
Watchdog time-out (during sleep)	0	0
External reset (not during sleep)	x	x
External reset (during sleep)	0	1
Normal power-up	1	1

- **File Select Register (04h: FSR).** This register selects the register for indirect addressing. Bits 0-7 select 1 of 256 registers in the current bank (the '84 only has one bank). As described in the PIC16C5x register descriptions, a read or write to register 00h will access the register pointed to by the FSR.
- **I/O Port A (05h).** 5-bit I/O port. This register is used to read and write I/O Port A. It can be read and written just as any other register. However, read instructions always read the I/O pins, regardless of whether the pins are programmed as inputs or outputs.

  Bit 4 (pin 3) has an open-collector output and shares its pin with the RTCC input.

PIC16C71 Registers *(continued)*

- **I/O Port B (06h).** 8-bit I/O port. Each of the Port B pins has a weak internal pull-up resistor (~250 µA). A pin's pull-up is turned off if the pin is configured as an output, and a single bit in the Option register can turn off all the pull-ups. The pull-up resistors are disabled on power-on reset.

  On bits 4-7 (pins 10-13), Port B has an interrupt on change feature that can generate an interrupt if any of the pins changes state. Any pin configured as an output is excluded from the interrupt feature.

  This interrupt can wake up the chip from sleep. Along with the internal pull-ups, the interrupt from sleep feature makes it easy to interface to a keypad and have wake-up on key press.

- **EEPROM Data Register (08h: EEDATA).** This register is used to read and write the 64 EEPROM registers.

- **EEPROM Address Register (09h: EEADR).** This register is used to set the address of the EEPROM register that will be read or written.

- **Program Counter Latch High (0Ah: PCLATH).** This register is used to access the 5 high bits of the program counter (PCH). Unlike the lower 8 bits (PCL), the 5 high bits are not directly addresable. Instead, they are stored in PCLATH for later use. When the program counter is loaded with a new value during a JMP, CALL, or a write to PCL, the high bits are loaded from PCLATH.

PIC16C84 Registers *(continued)*

- **Interrupt Control Register (0Bh: INTCON).** This register is used to enable interrupts and to determine what caused an interrupt. The function of each bit is given below:

Bit	Function
0	**Port B interrupt.** This bit is set if an interrupt was the result of a transition on any of the upper four bits of Port B. It will remain set until cleared by software.
1	**External interrupt.** Set if an interrupt was caused by a transition on the external INT pin. Must be cleared by software.
2	**RTCC overflow interrupt.** Set if an interrupt was caused by an overflow in the real-time clock/counter. Must be cleared by software.
3	**Port B interrupt enable.** Determines whether the Port B interrupt is enabled (“1”) or disabled (“0”).
4	**External interrupt enable.** Determines whether the external interrupt is enabled (“1”) or disabled (“0”).
5	**RTCC interrupt enable.** Determines whether the RTCC interrupt is enabled (“1”) or disabled (“0”).
6	**EEPROM interrupt enable.** Determines whether the EEPROM write complete interrupt is enabled (“1”) or disabled (“0”).
7	**Global interrupt enable.** Clearing this bit disables all interrupts. Setting this bit allows all interrupts that are individually enabled in bits 3-6.

- **Indirect Addressing Register (80h).** This register has the same function as register 00h.

PIC16C84 Registers *(continued)*

- **Option Register (81h: OPTION).** This register is used to set prescaler options, RTCC settings, external interrupt trigger edge, and Port B pull-up status. The function of each bit is shown below:

Bit	Function
0-2	**Prescaler ratio.** These 3 bits determine the prescaler input-to-output ratio. When using the prescaler with the RTCC, the seven possible ratios are 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256. When using the prescaler with the watchdog timer, the ratios are 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128. For example, let's say that the prescaler is assigned to the watchdog timer. To increase the watchdog time-out period to 64 times its normal length, the prescaler ratio would be set to 110b; this yields a watchdog period of approx. 1 second (64 x 18 ms).
3	**Prescaler assignment.** This bit determines whether the prescaler is assigned to the RTCC ("0") or to the watchdog timer ("1").
4	**RTCC trigger edge.** This bit determines whether the RTCC increments on a low-to-high ("0") or high-to-low ("1") transition on the RTCC pin.
5	**RTCC signal source.** This bit determines whether the RTCC is driven by the PIC's internal instruction clock ("0") or by the RTCC pin ("1").
6	**External interrupt trigger edge.** Determines whether an interrupt will be caused by a high-to-low ("0") or low-to-high ("1") transition on the external INT pin.
7	**Port B pull-up status.** If this bit is clear, Port B pull-up resistors are enabled on pins that are inputs. If this bit is set, all Port B pull-ups are disabled.

PIC16C84 Registers *(continued)*

- **Program Counter Low Byte (82h: PCL).** Same as register 02h.
- **Status Register (83h).** Same as register 03h.
- **File Select Register (84h: FSR).** Same as register 04h.
- **Data Direction Register for Port A (85h: TRISA).** This is the data direction register for Port A. In the Microchip data book, this register is referred to as "Tri-State A", hence "TRISA" as an abbreviation.

  Bits in this register which are set to "1" cause the corresponding bits in Port A to become inputs (the pins go into high impedance mode, allowing them to be driven by an external source). Bits which are cleared to "0" cause the corresponding bits in Port A to become outputs.
- **Data Direction Register for Port B (86h: TRISB).** This is the data direction register for Port B.

  Data direction registers in the PIC16C84 are addressable, unlike their counterparts in PIC16C5x devices.
- **EEPROM Control Register 1 (88h: EECON1).** This register is used to control the reading and writing of the 64 EEPROM registers. It's also used to determine if the last write cycle was completed. The function of each bit is given below:

Bit	Function
0	**Read control bit.** Setting this bit starts an EEPROM read cycle. The bit is automatically cleared after the read.
1	**Write control bit.** Setting this bit starts an EEPROM write cycle. The bit is automatically cleared after the write.
2	**Write enable bit.** If this bit is set, the EEPROM can be written; if it's clear, the EEPROM can only be read.

PIC16C84 Registers *(continued)*

Bit	Function
3	**Write error flag.** This bit is automatically set if an EEPROM write cycle is prematurely terminated by an MCLR reset (during sleep or normal operation) or by a watchdog reset (during normal operation).
4	**Write completion interrupt flag.** Set when an EEPROM write cycle is completed. Must be cleared in software. Corresponding interrupt enable bit is in INTCON register.

- **EEPROM Control Register 2 (89h: EECON2).** This register is not physically implemented, and therefore cannot be read. However, EECON2 must be written to whenever an EEPROM write cycle is performed. See the EEPROM read/write notes following the register descriptions.
- **Program Counter Latch High (8Ah).** Same as register 0Ah.
- **Interrupt Control Register (8Bh: INTCON).** Same as register 0Bh.

Reading & Writing EEPROM Registers

Separate from the PIC16C84's 1K of program EEPROM space, there exist 64 bytes of EEPROM that can be used by your program as nonvolatile data storage.

This memory is not mapped in the normal register space. Instead, it is accessed through two registers: EEDATA (holds data to be read/written) and EEADR (holds address of EEPROM location). Additionally, two special registers control reading and writing of the EEPROM.

Reading from the EEPROM. To read an EEPROM location, your program must write the address to the EEADR register and then set bit 0 of EECON1 (read control bit). The desired data can be read from EEDATA in the next instruction. The following code shows how to read the EEPROM:

Reading & Writing EEPROM Registers *(continued)*

```
mov   EEADR,#00h   ;Select first EEPROM location
setb  STATUS.5     ;Select register page 1
setb  EECON1.0     ;Start read cycle
clrb  STATUS.5     ;Select register page 0
mov   w,EEDATA     ;Read data
```

Writing to the EEPROM. To write an EEPROM location, your program must write the address to the EEADR register and the data to the EEDATA register. Then your program must set bit 2 of EECON1 (write enable), write a sequence of two bytes to EECON2, and then set bit 1 of EECON1 (write control bit). The write operation takes approximately 10 ms. The following code shows how to write the EEPROM:

```
mov   EEADR,#00h   ;Select first EEPROM location
mov   EEDATA,#41h  ;Store ASCII value for 'A'
setb  STATUS.5     ;Select register page 1
setb  EECON1.2     ;Set write enable bit
mov   EECON2,#55h  ;Write special byte #1
mov   EECON2,#AAh  ;Write special byte #2
setb  EECON1.1     ;Start write cycle
jb    EECON1.1,$   ;Wait until write bit clears
clrb  STATUS.5     ;Select register page 0
```