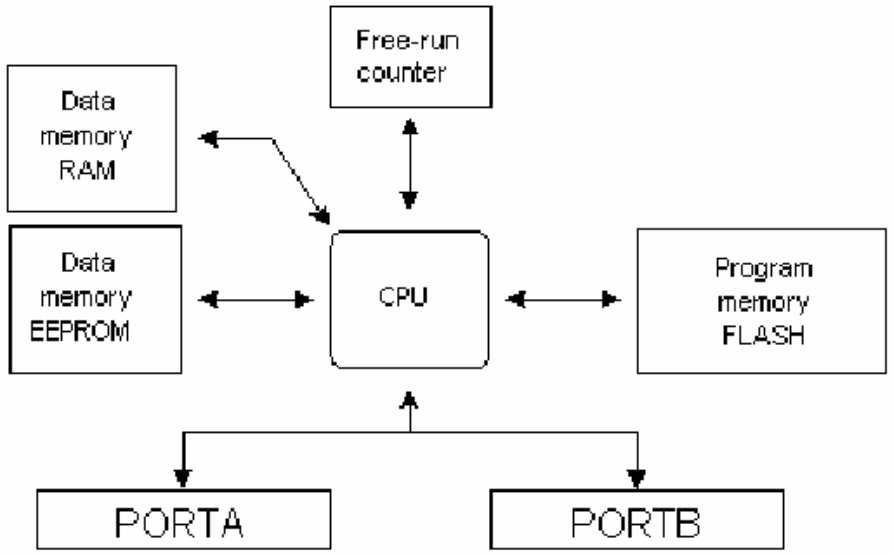
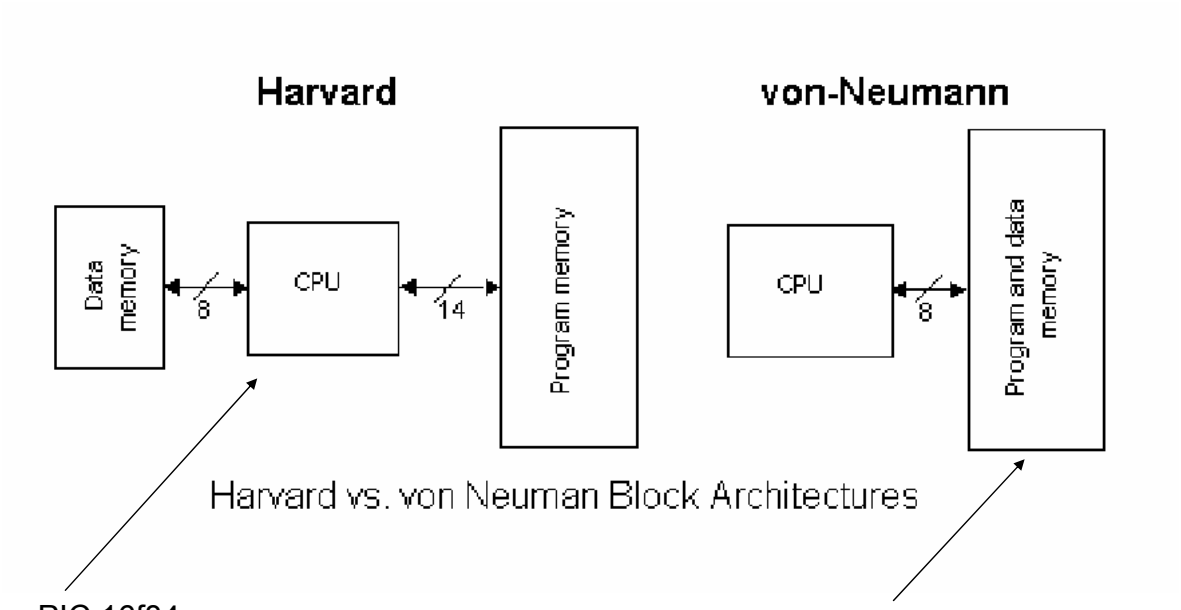
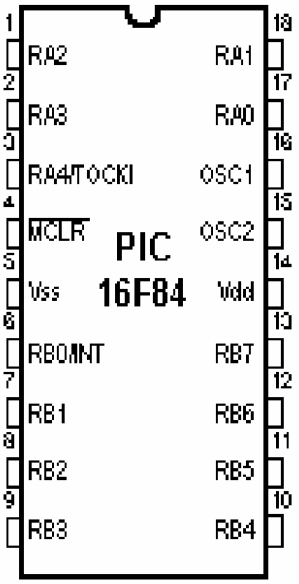
(Microcontroller)



PIC16F84 microcontroller outline


Harvard
von-Neumann
Data memory
8
CPU
14
Program memory
CPU
8
Program and data memory
Harvard vs. von Neuman Block Architectures
PIC 16f84
8086 microprocessor



Pin no.1 **RA2** Second pin on port A. Has no additional function
Pin no.2 **RA3** Third pin on port A. Has no additional function.
Pin no.3 **RA4** Fourth pin on port A. TOCK1 which functions as a timer is also found on this pin
Pin no.4 **MCLR** Reset input and Vpp programming voltage of a microcontroller
Pin no.5 **Vss** Ground of power supply.
Pin no.6 **RB0** Zero pin on port B. Interrupt input is an additional function.
Pin no.7 **RB1** First pin on port B. No additional function.
Pin no.8 **RB2** Second pin on port B. No additional function. Pin no.9 **RB3** Third pin on port B. No additional function.
Pin no.10 **RB4** Fourth pin on port B. No additional function.
Pin no.11 **RB5** Fifth pin on port B. No additional function.
Pin no.12 **RB6** Sixth pin on port B. 'Clock' line in program mode.
Pin no.13 **RB7** Seventh pin on port B. 'Data' line in program mode.
Pin no.14 **Vdd** Positive power supply pole.
Pin no.15 **OSC2** Pin assigned for connecting with an oscillator
Pin no.16 **OSC1** Pin assigned for connecting with an oscillator
Pin no.17 **RA2** Second pin on port A. No additional function
Pin no.18 **RA1** First pin on port A. No additional function.

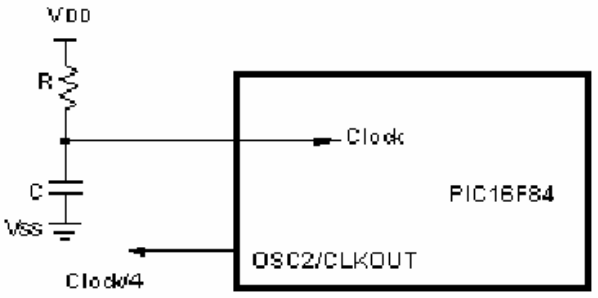
Memory organization
PIC16F84 has two separate memory blocks, one for data and the other for program. EEPROM memory and GPR registers in RAM memory make up a data block, and FLASH memory makes up a program block.

Program memory:
Program memory has been realized in FLASH technology which makes it possible to program a microcontroller many times before it's installed into a device, and even after its installment if eventual changes in program or process parameters should occur. The size of program memory is 1024 locations with 14 bits width where locations zero and four are reserved for reset and interrupt vector.

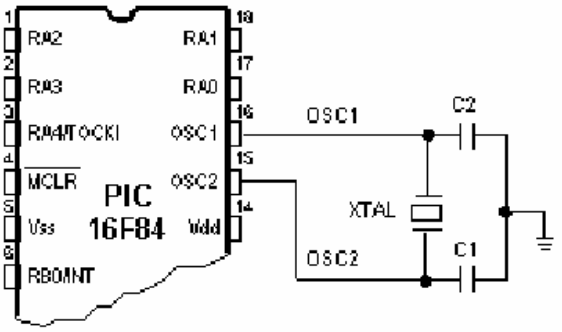
Data memory

Data memory consists of EEPROM and RAM memories. EEPROM memory consists of 64 eight bit locations whose contents is not lost during loosing of power supply. EEPROM is not directly addressible, but is accessed indirectly through EEADR and EEDATA registers. As EEPROM memory usually serves for storing important parameters (for example, of a given temperature in temperature regulators) , there is a strict procedure for writing in EEPROM which must be followed in order to avoid accidental writing. RAM memory for data occupies space on a memory map from location 0x0C to 0x4F which comes to 68 locations. Locations of RAM memory are also called GPR registers which is an abbreviation for General Purpose Registers. GPR registers can be accessed regardless of which bank is selected at the moment.

Osilator: RC, Crystal,..

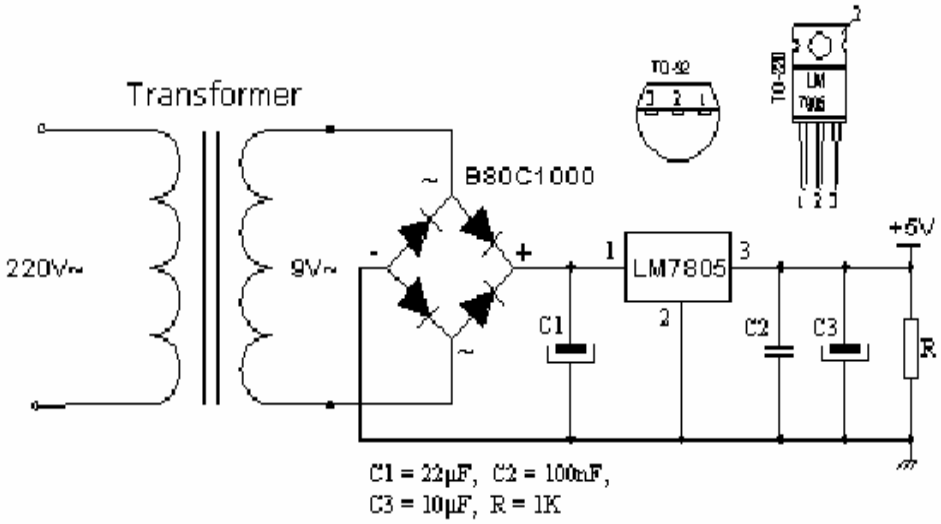


Note:This pin can be configured as input/output pin



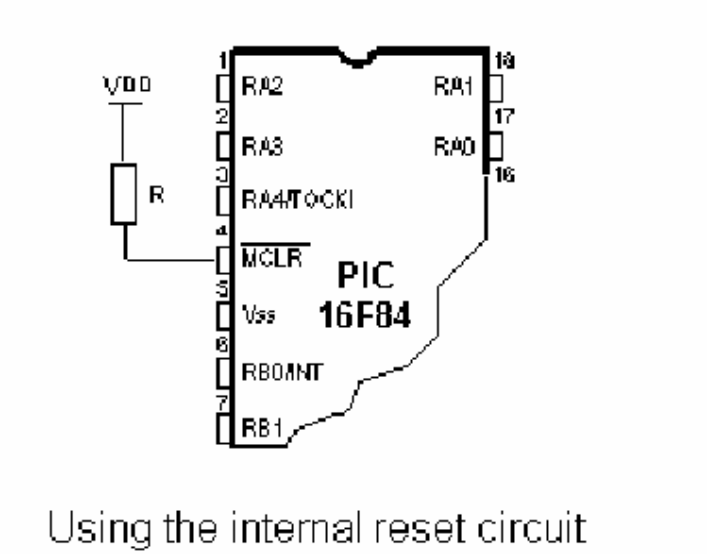
Connecting the quartz oscillator to give clock to a microcontroller

Simple Power supply



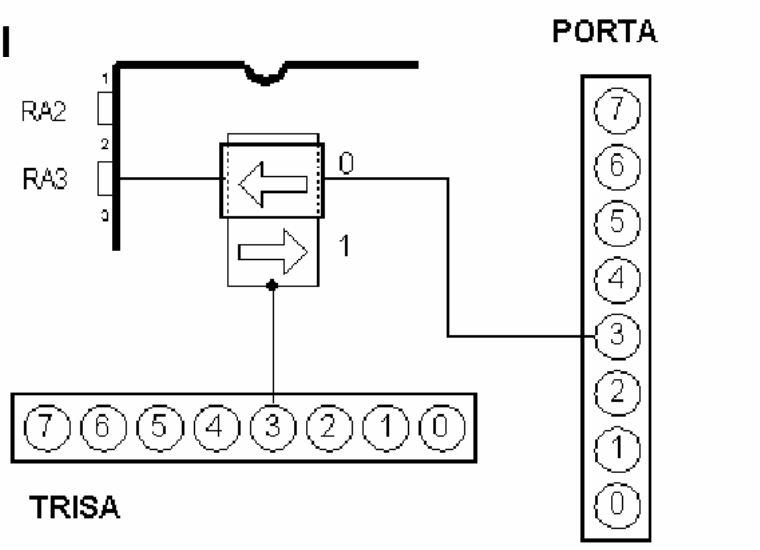
2.2 Reset

Reset is used for putting the microcontroller into a 'known' condition. That practically means that microcontroller can behave rather inaccurately under certain undesirable conditions. In order to continue its proper functioning it has to be reset, meaning all registers would be placed in a starting position. Reset is not only used when microcontroller doesn't behave the way we want it to, but can also be used when trying out a device as an interrupt in program execution, or to get a microcontroller ready when reading in a program.



Using the internal reset circuit

I/O selection (Bidirectional ports)



Relationship between TRISA and PORTA register

STATUS Register

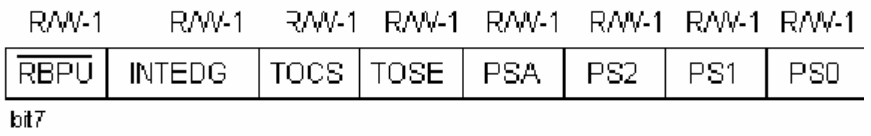
R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							

Legend:
R = Readable bit **W** = Writable bit
U = Unimplemented bit, read as '0' - n = Value at power-on reset

STATUS Register

bit 0 **C** (Carry) TransferBit that is affected by operations of addition, subtraction and shifting. 1= transfer occured from the highest resulting bit 0=transfer did not occur
C bit is affected by ADDWF, ADDLW, SUBLW, SUBWF instructions.
bit 1 **DC** (Digit Carry) DC TransferBit affected by operations of addition, subtraction and shifting. Unlike C bit, this bit represents transfer from the fourth resulting place. It is set by addition when occurs carry from bit3 to bit4, or by subtraction when occurs borrow from bit4 to bit3, or by shifting in both direction. 1=transfer occured on the fourth bit according to the order of the result0=transfer did not occurDC bit is affected by ADDWF, ADDLW, SUBLW, SUBWF instructions.
bit 2 **Z** (Zero bit) Indication of a zero resultThis bit is set when the result of an executed arithmetic or logic operation is zero. 1=result equals zero0=result does not equal zero
bit 3 **PD** (Power-down bit)Bit which is set whenever power supply is brought to a microcontroller as it starts running, after each regular reset and after execution of instruction CLRWDT. Instruction SLEEP resets it when microcontroller falls into low consumption/usage regime. Its repeated setting is possible via reset or by turning the supply on, or off . Setting can be triggered also by a signal on RB0/INT pin, change on RB port, completion of writing in internal DATA EEPROM, and by a watchdog, too.1=after supply has been turned on0= executing SLEEP instruction
bit 4 **TO** Time-out ; Watchdog overflow.Bit is set after turning on the supply and execution of CLRWDT and SLEEP instructions. Bit is reset when watchdog gets to the end signaling that something is not right.1=overflow did not occur0=overflow did occur
bit6:5 **RP1:RP0** (Register Bank Select bits) These two bits are upper part of the address for direct addressing. Since instructions which address the memory directly have only seven bits, they need one more bit in order to address all 256 bytes which is how many bytes PIC16F84 has. RP1 bit is not used, but is left for some future expansions of this microcontroller.01=first bank00=zero bank
bit 7 **IRP** (Register Bank Select bit) Bit whose role is to be an eighth bit for indirect addressing of internal RAM.1=bank 2 and 30=bank 0 and 1 (from 00h to FFh)STATUS register contains arithmetic status ALU (C, DC, Z), RESET status (TO, PD) and bits for selecting of memory bank (IRP, RP1, RP0). Considering that selection of memory bank is controlled through this register, it has to be present in each bank. Memory bank will be discussed in more detail in Memory organization chapter. STATUS register can be a destination for any instruction, with any other register. If STATUS register is a destination for instructions which affect Z, DC or C bits, then writing to these three bits is not possible.

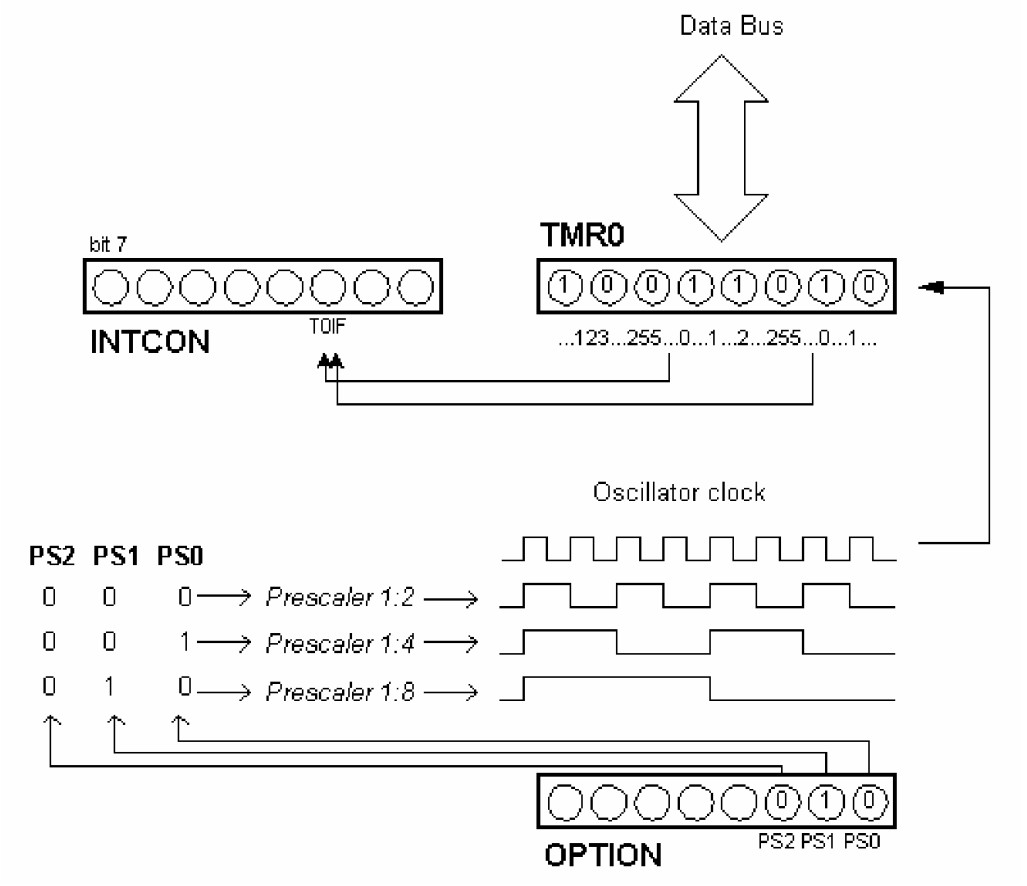
OPTION register (Timer)



Bits	TMR0	WDT
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

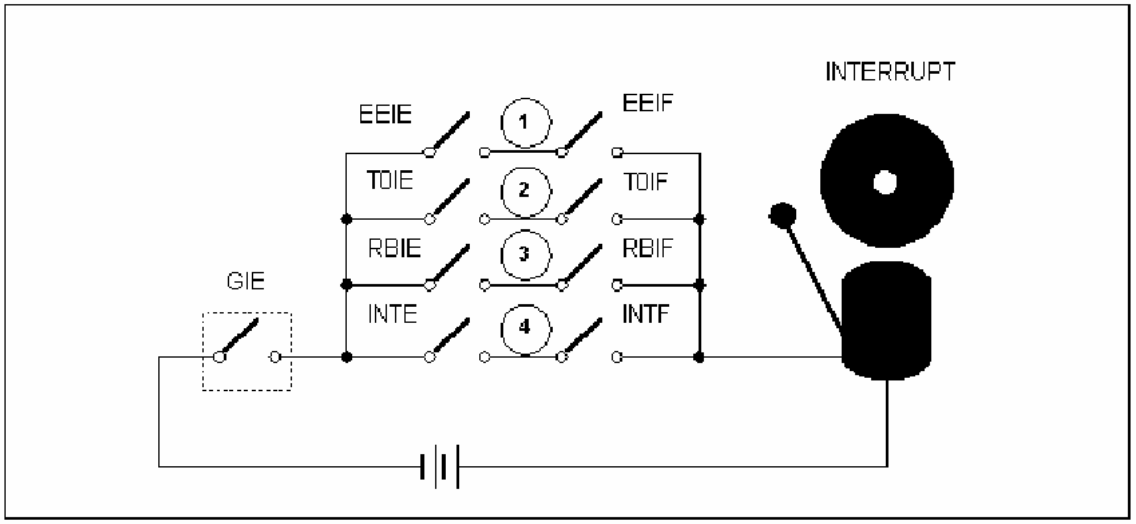
bit 0:2 **PS0, PS1, PS2** (Prescaler Rate Select bit) These three bits define prescaler rate select bit. What a prescaler is and how these bits can affect the work of a microcontroller will be explained in section on TMR0.
bit 3 **PSA** (Prescaler Assignment bit)Bit which assigns prescaler between TMR0 and watchdog.1=prescaler is assigned to watchdog0=prescaler is assigned to a free-run timer TMR0
bit 4 **T0SE** (TMR0 Source Edge Select bit)If it is allowed to trigger TMR0 by impulses from the pin RA4/T0CKI, this bit determines whether this will be to the falling or rising edge of a signal.1=falling edge0=rising edge
bit 5 **TOCS** (TMR0 Clock Source Select bit)This pin enables free-run timer to increment its state either from internal oscillator on every ¼ oscillator clock, or through external impulses on RA4/T0CKI pin.1=external impulses0=1/4 internal clock
bit 6 I**NTEDG** (Interrupt Edge Select bit)If interrupt is enabled possible this bit will determine the edge at which an interrupt will be activated on pin RB0/INT.1=rising edge0=falling edge
bit 7 **RBPU** (PORTB Pull-up Enable bit) This bit turns on and off internal 'pull-up' resistors on port B.1= "pull-up" resistors turned off 0= "pull-up" resistors turned on

Timer


Data Bus
TMR0
bit 7
1 0 0 1 1 0 1 0
INTCON
TOIF
...123...255...0...1...2...255...0...1...
Oscillator clock
PS2 PS1 PS0
0 0 0 ⟶ Prescaler 1:2 ⟶
0 0 1 ⟶ Prescaler 1:4 ⟶
0 1 0 ⟶ Prescaler 1:8 ⟶
0 1 0
OPTION
PS2 PS1 PS0

İnterrupt Register (INTCON Register)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF



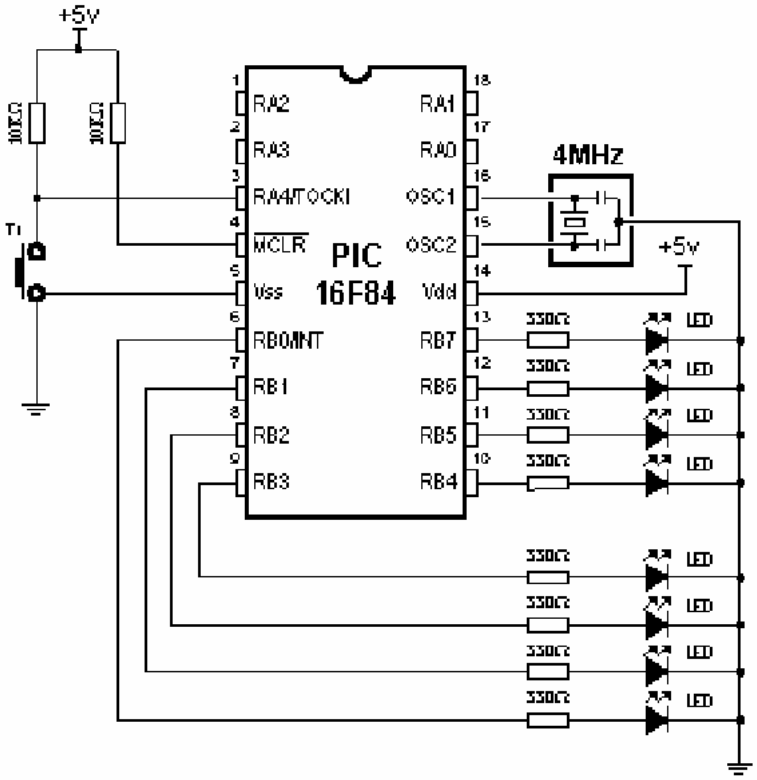
Simplified outline of PIC16F84 microcontroller interrupt

İnterrupt Register (INTCON Register)

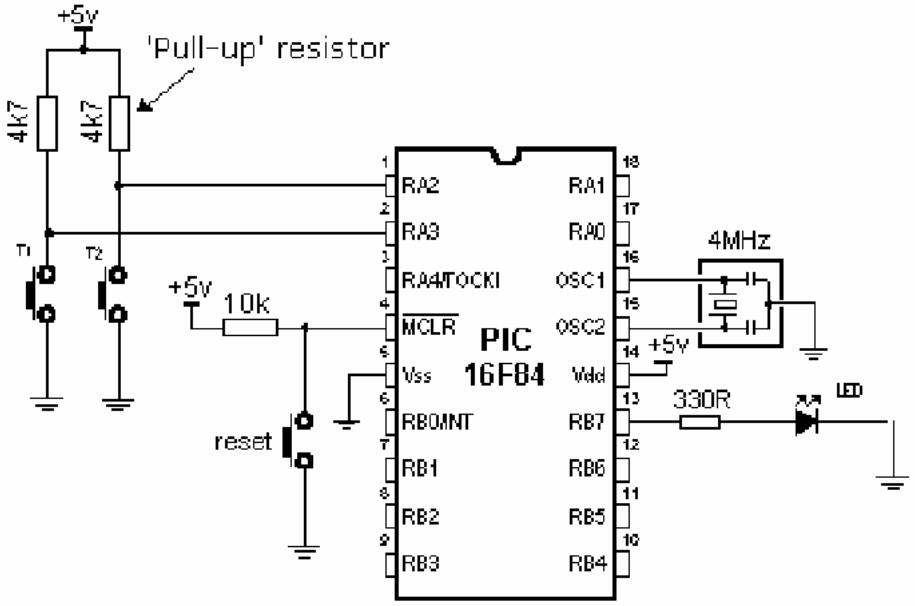
bit 0 **RBIF** (RB Port Change Interrupt Flag bit) Bit which informs about changes on pins 4, 5, 6 and 7 of port B.1=at least one pin has changed its status0=no change occured on any of the pins
bit 1 **INTF** (INT External Interrupt Flag bit) External interrupt occured.1=interrupt occured0=interrupt did not occurIf a rising or falling edge was detected on pin RB0/INT, (which is defined with bit INTEDG in OPTION register), bit INTF is set. Bit must be cleared in interrupt subprogram in order to detect the next interrupt.
bit 2 **T0IF** (TMR0 Overflow Interrupt Flag bit) Overflow of counter TMR0.1= counter changed its status from FFh to 00h0=overflow did not occurBit must be cleared in program in order for an interrupt to be detected
.bit 3 **RBIE** (RB port change Interrupt Enable bit) Enables interrupts to occur at the change of status of pins 4, 5, 6, and 7 of port B. 1= enables interrupts at the change of status0=interrupts disabled at the change of statusIf RBIE and RBIF were simultaneously set, an interrupt would occur.
bit 4 **INTE** (INT External Interrupt Enable bit) Bit which enables external interrupt from pin RB0/INT.1=external interrupt enabled0=external interrupt disabledIf INTE and INTF were set simultaneously, an interrupt would occur
.bit 5 **T0IE** (TMR0 Overflow Interrupt Enable bit) Bit which enables interrupts during counter TMR0 overflow.1=interrupt enabled0=interrupt disabledIf T0IE and T0IF were set simultaneously, interrupt would occur.
Bit 6 **EEIE** (EEPROM Write Complete Interrupt Enable bit) Bit which enables an interrupt at the end of a writing routine to EEPROM1=interrupt enabled0=interrupt disabledIf EEIE and EEIF (which is in EECON1 register) were set simultaneously , an interrupt would occur.
Bit 7 **GIE** (Global Interrupt Enable bit) Bit which enables or disables all interrupts.1=all interrupts are enabled0=all interrupts are disabledPIC16F84 has four interrupt sources:1. Termination of writing data to EEPROM2. TMR0 interrupt caused by timer overflow3. Interrupt during alteration on RB4, RB5, RB6 and RB7 pins of port B.4. External interrupt from RB0/INT pin of microcontroller

Interrupt initialization

```
clrf INTCON              ; all interrupts disabled
movlw B'00010000'        ; external interrupt only is enabled
bsf INTCON, GIE          ; occurrence of interrupts allowed
```



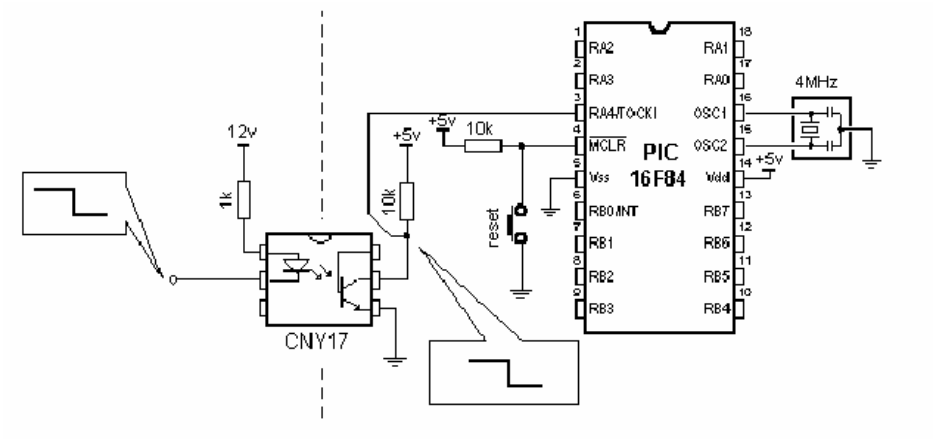
Connecting LED diodes to PORTB microcontroller



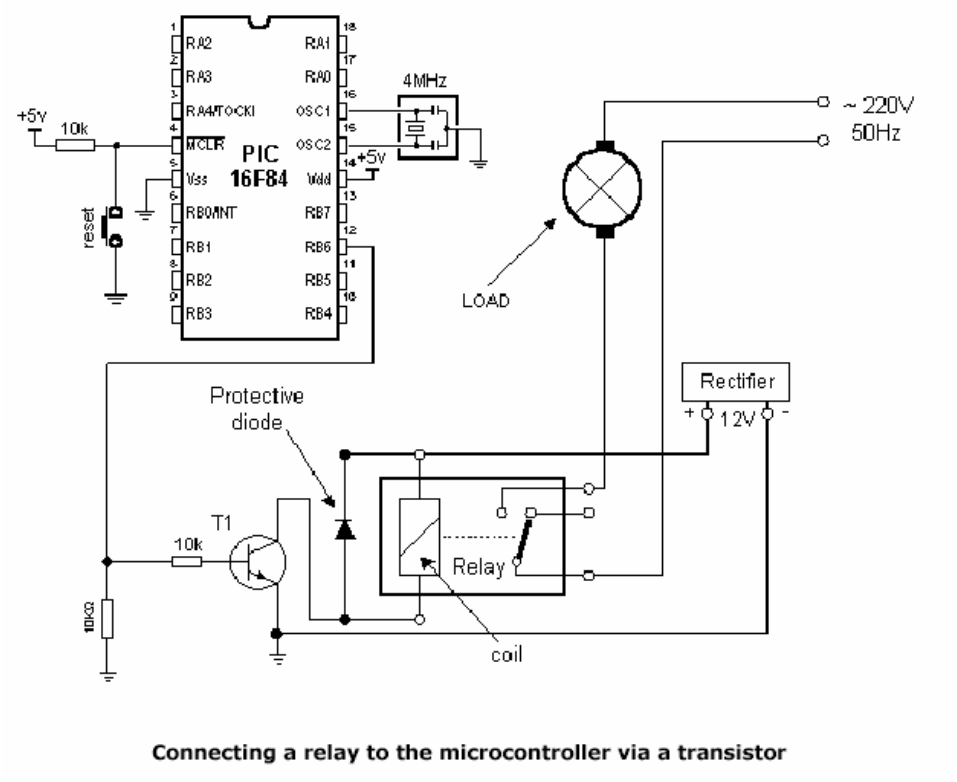
Example of connecting keys to microcontroller pins.

Optocoupler on an input line (İzolation)

The way it works is simple: when a signal arrives, the LED within the optocoupler is turned on, and it shines on the base of a photo-transistor within the same case. When the transistor is activated, the voltage between collector and emitter falls to 0.5V or less and the microcontroller sees this as a logic zero on its RA4 pin. The example below is a counter, used for counting products on production line, determining motor speed, counting the number of revolutions of an axis etc. Let the sensor be a micro-switch. Each time the switch is closed, the LED is illuminated. The LED 'transfers' the signal to the phototransistor and the operation of the photo-transistor delivers a LOW to input RA4 of a microcontroller. A program in the microcontroller will be needed to prevent false counting and an indicator connected to any of the outputs of the microcontroller will shows the current state of the counter.



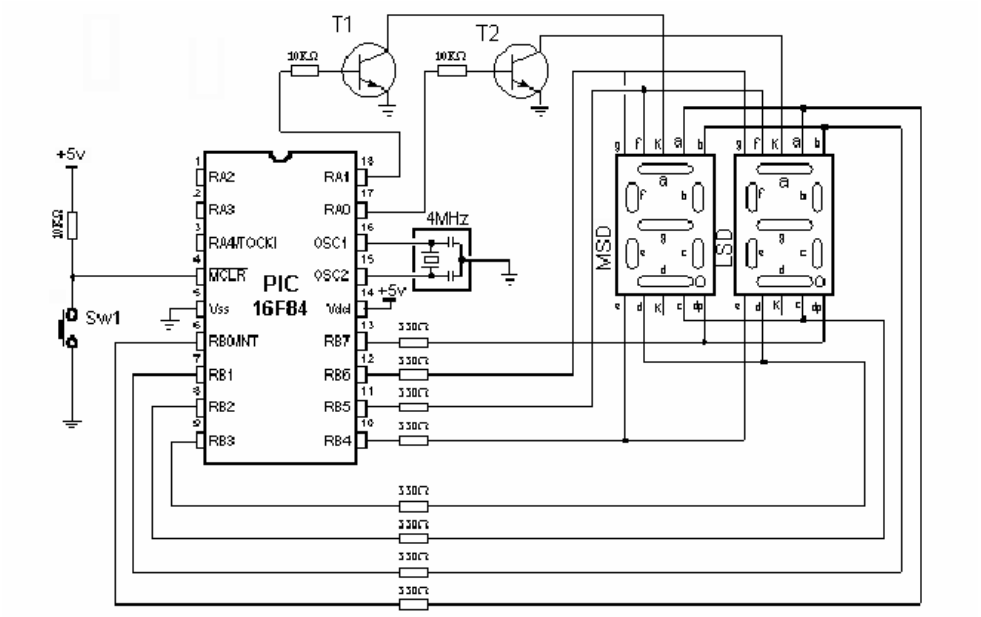
(Relay control)



Connecting a relay to the microcontroller via a transistor

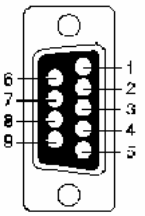
Seven-Segment Display (multiplexing)

The segments in a 7-segment display are arranged to form a single digit from 0 to F as shown in the animation:



Connecting a microcontroller to 7-segment displays in multiplex mode

Serial Communication



1. CD (Carrier Detect)
2. RXD (Receive Data)
3. TXD (Transmit Data)
4. DTR (Data terminal Ready)
5. GND (Ground)
6. DSR (Data Set Ready)
7. RTS (Request To Send)
8. CTS (Clear To Send)
9. RI (Ring Indicator)