

Features of 8051 microcontroller are given below.

Eight bit CPU

On chip clock oscillator

4Kbytes of internal program memory (code memory) *[ROM]*

128 bytes of internal data memory *[RAM]*

64 Kbytes of external program memory address space.

64 Kbytes of external data memory address space.

32 bi directional I/O lines (can be used as four 8 bit ports or 32 individually addressable I/O

lines)

Two 16 Bit Timer/Counter :T0, T1

Full Duplex serial data receiver/transmitter

Four Register banks with 8 registers in each bank.

Sixteen bit Program counter (PC) and a data pointer (DPTR)

8 Bit Program Status Word (PSW)

8 Bit Stack Pointer

Five vector interrupt structure (RESET not considered as an interrupt.)

8051 CPU consists of 8 bit ALU with associated registers like accumulator ‘A’ , B register,

PSW, SP, 16 bit program counter, stack pointer.

ALU can perform arithmetic and logic functions on 8 bit variables.

8051 has 128 bytes of internal RAM which is divided into

o Working registers [00 – 1F]

o Bit addressable memory area [20 – 2F]

o General purpose memory area (Scratch pad memory) [30-7F]

8051 has 4 K Bytes of internal ROM. The address space is from 0000 to 0FFFh. If the

program size is more than 4 K Bytes 8051 will fetch the code automatically from external

memory.

Accumulator is an 8 bit register widely used for all arithmetic and logical operations.

Accumulator is also used to transfer data between external memory. B register is used along

with Accumulator for multiplication and division. A and B registers together is also called

MATH registers.

PSW (Program Status Word). This is an 8 bit register which contains the arithmetic status of

ALU and the bank select bits of register banks.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CY | AC | F0 | RS1 | RS0 | OV |  | Ps |

CY - carry flag

AC - auxiliary carry flag

F0 - available to the user for general purpose

RS1,RS0 - register bank select bits

OV - overflow

P - parity

Stack Pointer (SP) – it contains the address of the data item on the top of the stack. Stack

may reside anywhere on the internal RAM. On reset, SP is initialized to 07 so that the default

stack will start from address 08 onwards.

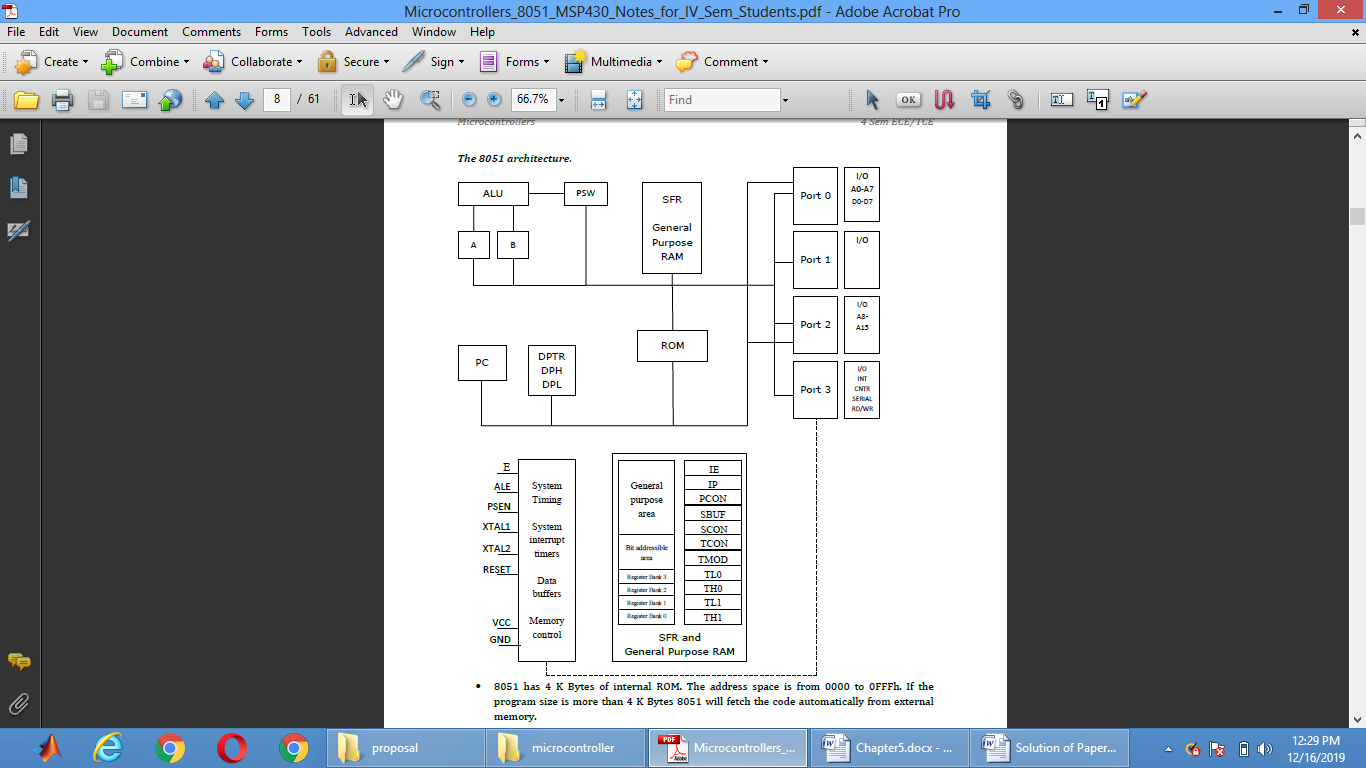
Data Pointer (DPTR) – DPH (Data pointer higher byte), DPL (Data pointer lower byte). This

is a 16 bit register which is used to furnish address information for internal and external

program memory and for external data memory.

Program Counter (PC) – 16 bit PC contains the address of next instruction to be executed.

On reset PC will set to 0000. After fetching every instruction PC will increment by one.



**Register Modes:**

8051 addressing modes are classified as follows.

1. Immediate addressing.

2. Register addressing.

3. Direct addressing.

4. Indirect addressing.

5. Indexed addressing.

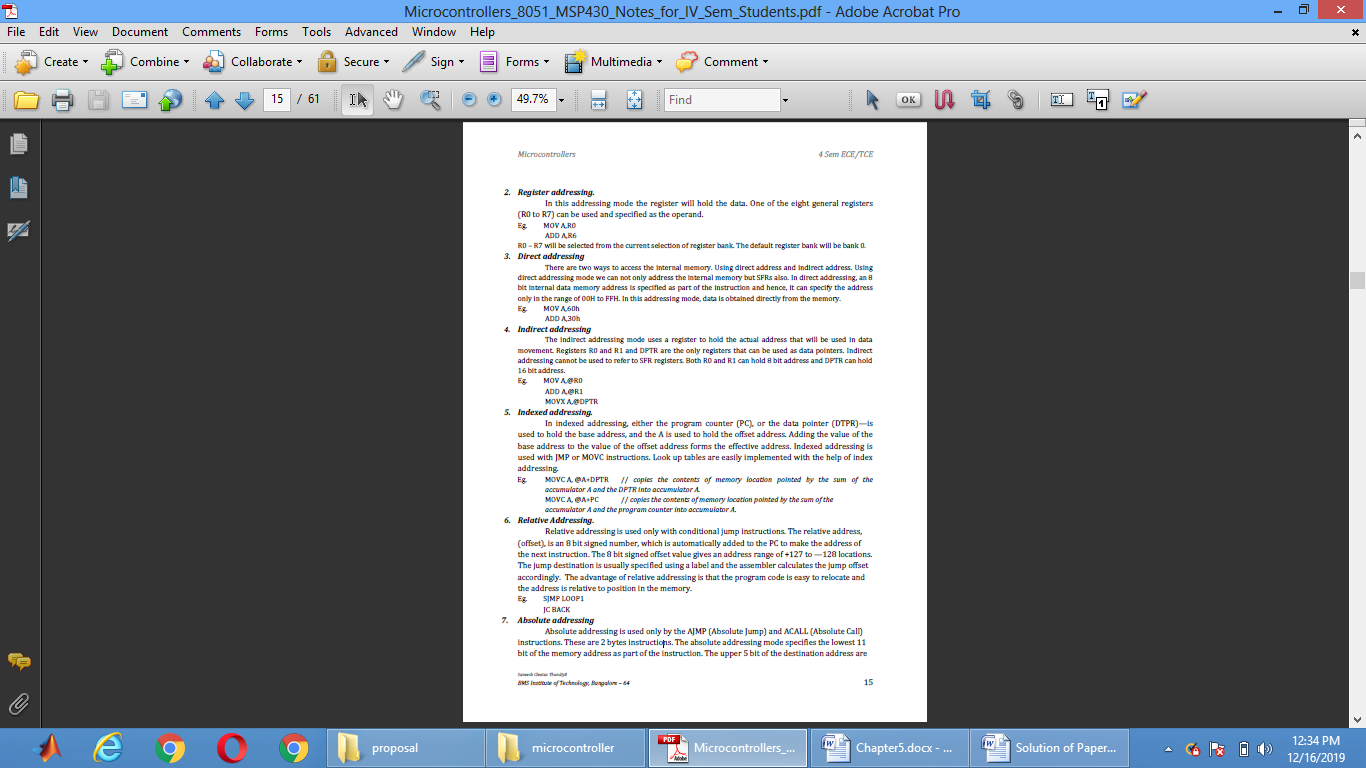
***1. Immediate addressing.***

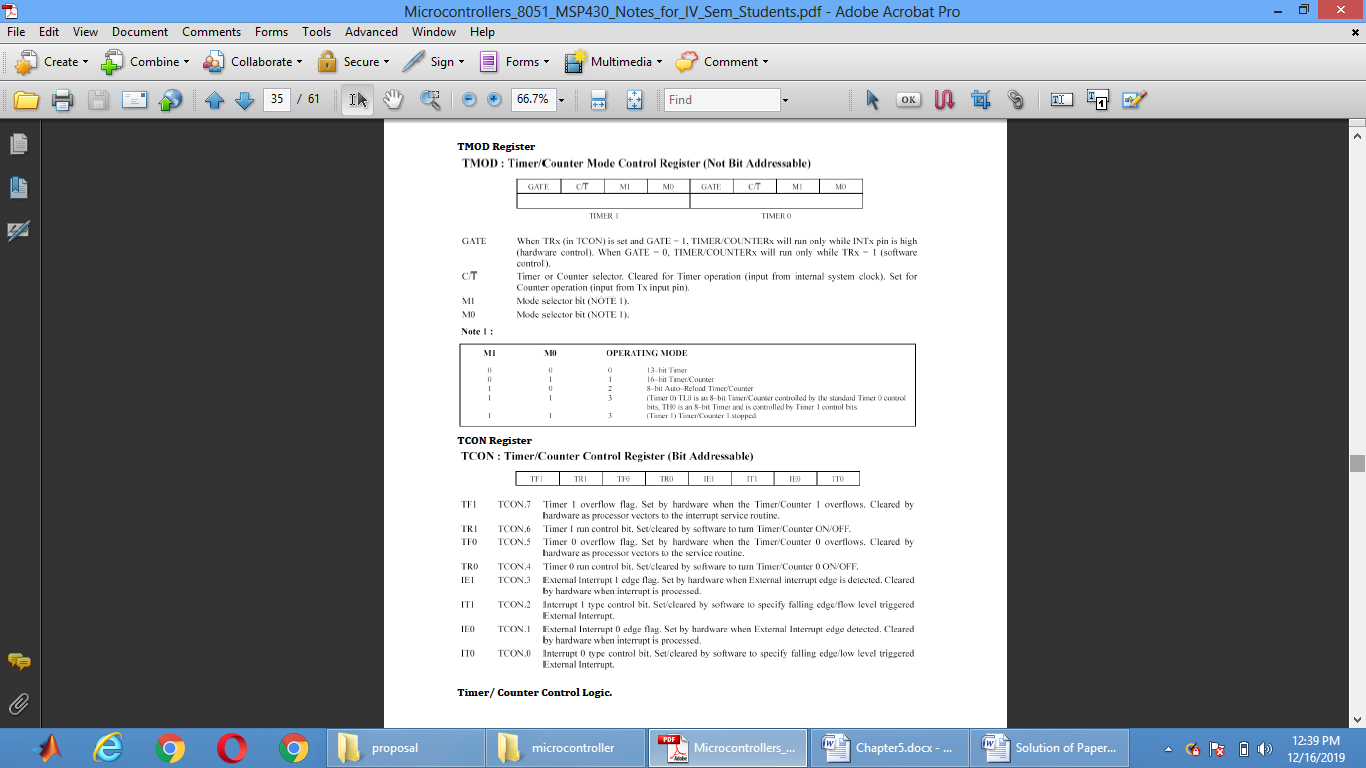
In this addressing mode the data is provided as a part of instruction itself. In other words

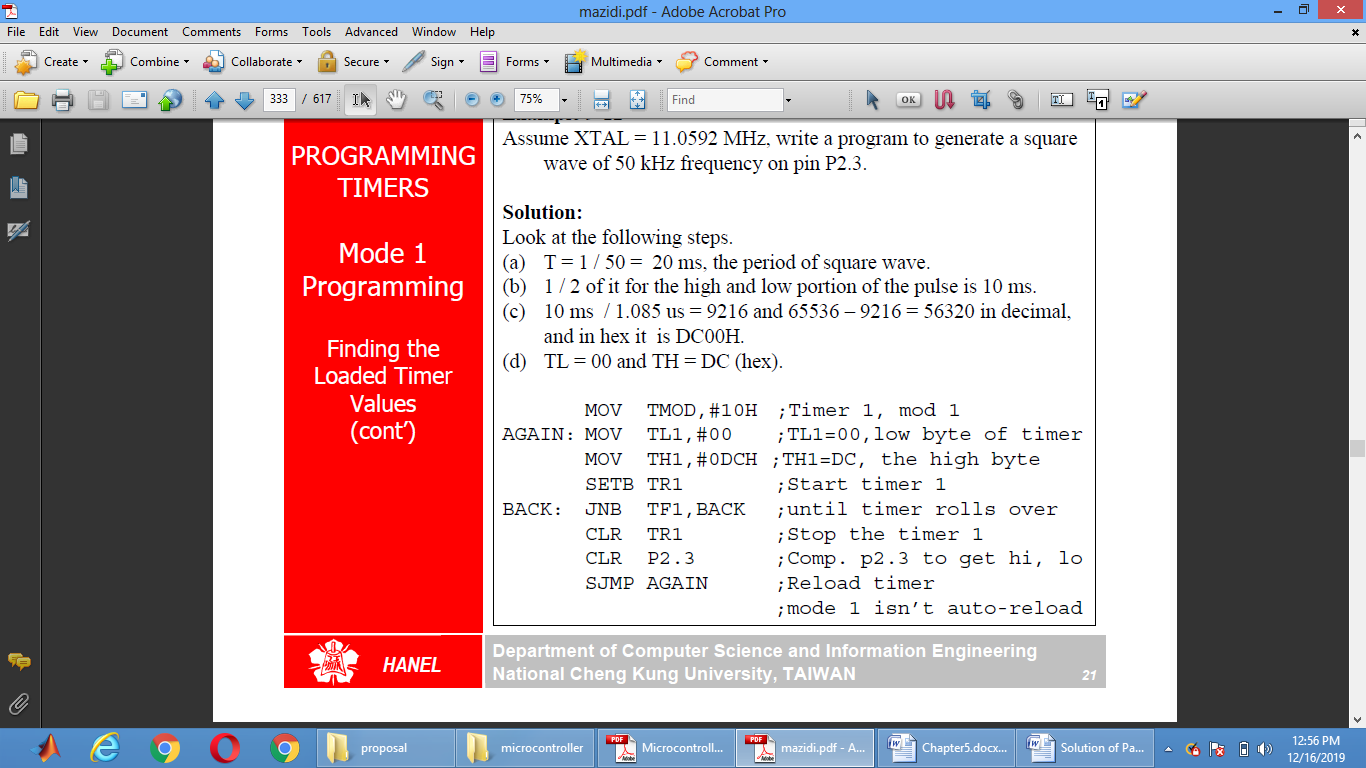
data immediately follows the instruction.

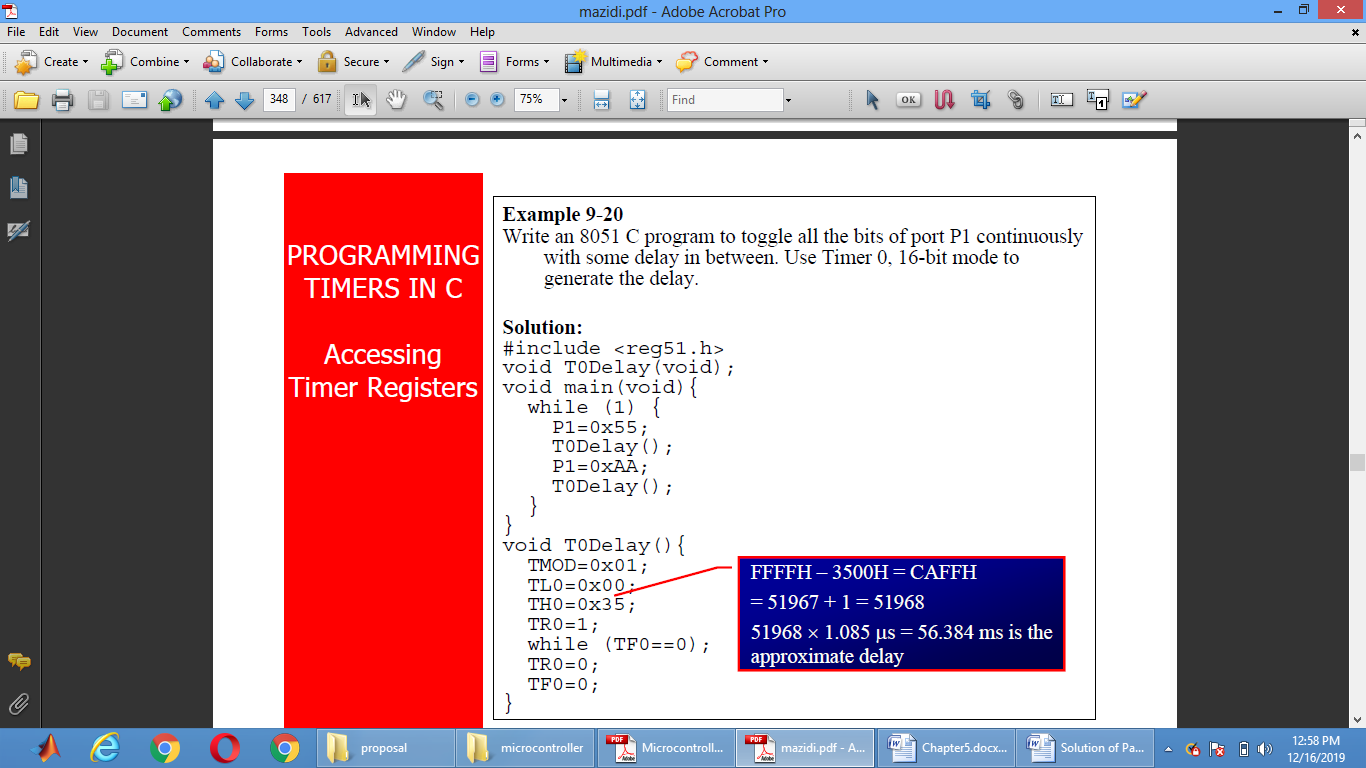
Eg. MOV A,#30H

ADD A, #83 # Symbol indicates the data is immediate









Interrrupt

An interrupt is an external or internal event that interrupts the microcontroller to inform it that a

device needs its service

A single microcontroller can serve several devices by two ways

Interrupts

* Whenever any device needs its service, the device notifies the microcontroller by sending it an interrupt signal
* Upon receiving an interrupt signal, the microcontroller interrupts whatever it is doing

and serves the device

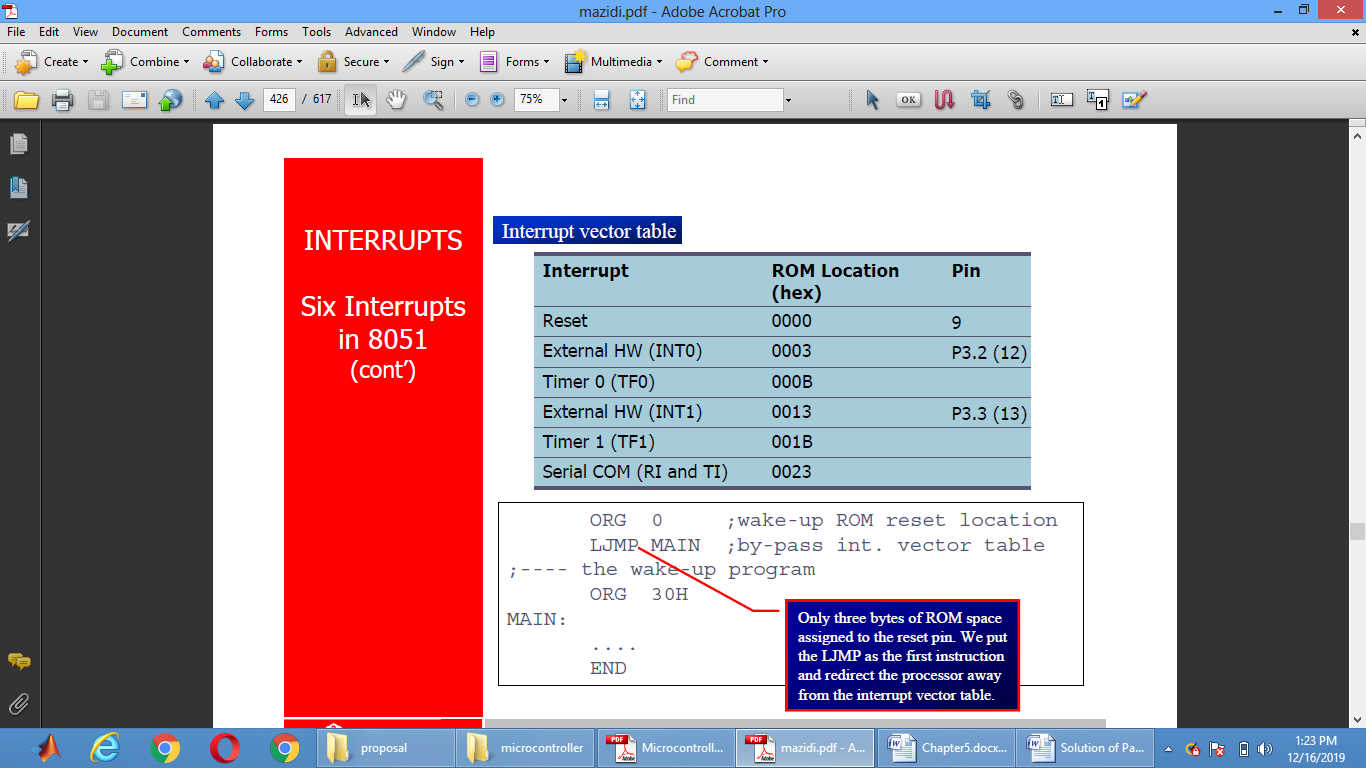
* The program which is associated with the interrupt is called the interrupt service routine

(ISR) or interrupt handler.

Polling

* The microcontroller continuously monitors the status of a given device
* When the conditions met, it performs the service
* After that, it moves on to monitor the next device until every one is serviced.

Interrupt Vector

* Six interrupts are allocated as follows
* Reset – power-up reset
* Two interrupts are set aside for the timers: one for timer 0 and one for timer 1
* Two interrupts are set aside for hardware external interrupts
  + P3.2 and P3.3 are for the external hardware
* interrupts INT0 (or EX1), and INT1 (or EX2)
* Serial communication has a single interrupt that belongs to both receive and transfer
* 

A switch is connected to pin P1.2. Write an 8051 C program to monitor SW and create the following frequencies on pin P1.7:

SW=0: 500Hz

SW=1: 750Hz, use Timer 0, mode 1 for both of them.

**Solution:**

#include <reg51.h>

sbit mybit=P1^5;

sbit SW=P1^7;

void T0M1Delay(unsigned char);

void main(void){

SW=1;

while (1) {

mybit=~mybit;

if (SW==0)

T0M1Delay(0);

else

T0M1Delay(1);

}

}

void T0M1Delay(unsigned char c){

TMOD=0x01;

if (c==0) {

TL0=0x67;

TH0=0xFC;

}

else {

TL0=0x9A;

TH0=0xFD;

}

TR0=1;

while (TF0==0);

TR0=0;

TF0=0;

}

In the following program, we create a square wave of 50% duty cycle (with equal portions high and low) on the P1.5 bit. Timer 0 is used to generate the time delay. Analyze the program

MOV TMOD,#01 ;Timer 0, mode 1(16-bit mode)

HERE:

MOV TL0,#0F2H ;TL0=F2H, the low byte

MOV TH0,#0FFH ;TH0=FFH, the high byte

CPL P1.5 ; toggle P1.5

ACALL DELAY

SJMP HERE

DELAY:

SETB TR0 ;start the timer 0

AGAIN: JNB TF0,AGAIN ;monitor timer flag 0 until it rolls over

CLR TR0 ;stop timer 0

CLR TF0 ;clear timer 0 flag

RET

Interrrupt Periority

We can alter the sequence of interrupt priority by assigning a higher priority to any one of the interrupts by programming a register called IP

(interrupt priority)

To give a higher priority to any of the interrupts, we make the corresponding bit in the IP register high

When two or more interrupt bits in the IP register are set to high

While these interrupts have a higher priority than others, they are serviced according to the

sequence of Table

Interrupt Priority Register (Bit-addressable)

-- IP.7 Reserved

-- IP.6 Reserved

PT2 IP.5 Timer 2 interrupt priority bit (8052 only)

PS IP.4 Serial port interrupt priority bit

PT1 IP.3 Timer 1 interrupt priority bit

PX1 IP.2 External interrupt 1 priority bit

PT0 IP.1 Timer 0 interrupt priority bit

PX0 IP.0 External interrupt 0 priority bit

Write an 8051 C program to toggle all bits of P2 continuously every 500 ms. Use Timer 1, mode 1 to create the delay.

**Solution:**

//tested for XTAL = 11.0592 MHz

#include <reg51.h>

void T1M1Delay(void);

void main(void){

unsigned char x;

P2=0x55;

while (1) {

P2=~P2;

for (x=0;x<20;x++)

T1M1Delay();

}

}

void T1M1Delay(void){

TMOD=0x10;

TL1=0xFE;

TH1=0xA5;

TR1=1;

while (TF1==0);

TR1=0;

TF1=0;

}

A switch is connected to pin P1.7. Write a program to check the status

of SW and perform the following:

(a) If SW=0, send letter ‘N’ to P2

(b) If SW=1, send letter ‘Y’ to P2

**Solution:**

SETB P1.7 ;make P1.7 an input

AGAIN: JB P1.2,OVER ;jump if P1.7=1

MOV P2,#’N’ ;SW=0, issue ‘N’ to P2

SJMP AGAIN ;keep monitoring

OVER: MOV P2,#’Y’ ;SW=1, issue ‘Y’ to P2

SJMP AGAIN ;keep monitoring

Different type of Buses:

The CPU is connected to memory and I/O through strips of wire called a bus Carries information from place to place

Address bus

Data bus

Control bus

For a device (memory or I/O) to be recognized by the CPU, it must be assigned an address

The address assigned to a given device must be unique

The CPU puts the address on the address bus, and the decoding circuitry finds the device

**Data bus**

The CPU either gets data from the device or sends data to it

**Control bus**

Provides read or write signals to the device to indicate if the CPU is asking for information or sending it information