

MICROCONTROLLER

UNIT-III

Lecture-10

INTERRUPTS

Example

- ▶ **Example:** Show the instructions to (a) enable the serial interrupt, timer 0 interrupt, and external hardware interrupt 1 (EX1), and (b) disable (mask) the timer 0 interrupt, then (c) show how to disable all the interrupts with a single instruction.
- ▶ **Solution:** (a) `MOV IE, #10010110B` ;enable serial, ;timer 0, EX1 Another way to perform the same manipulation is

Contd.

SETB IE.7 ;EA=1, global enable

SETB IE.4 ;enable serial interrupt

SETB IE.1 ;enable Timer 0 interrupt

SETB IE.2 ;enable EX1

(b) CLR IE.1 ;mask (disable) timer 0;interrupt only

(c) CLR IE.7 ;disable all interrupts

- ▶ **Example:** Write a program that continuously get 8-bit data from P0 and sends it to P1 while simultaneously creating a square wave of 200 μ s period on pin P2.1.

Contd.

- ▶ Use timer 0 to create the square wave. Assume that XTAL = 11.0592 MHz.
- ▶ **Solution:** We will use timer 0 in mode 2 (auto reload). $TH0 = 100 / 1.085 \text{ us} = 92$
;--upon wake-up go to main, avoid using
;memory allocated to Interrupt Vector Table

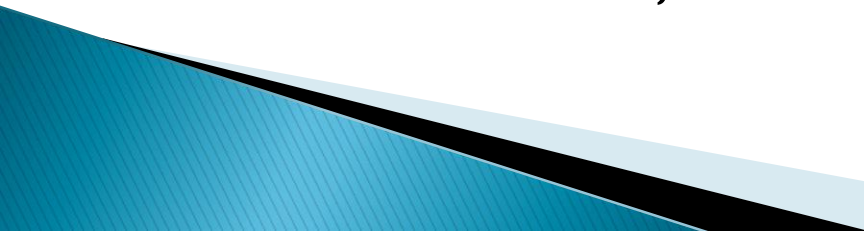
```
ORG 0000H
```

```
LJMP MAIN ;by-pass interrupt  
vector table
```

```
;
```

Contd.

```
;-ISR for timer 0 to generate square wave
    ORG 000BH ;Timer 0 interrupt
vector table
    CPL P2.1 ;toggle P2.1 pin
    RETI ;return from ISR
;-The main program for initialization
    ORG 0030H ;after vector table space
MAIN:    MOV TMOD,#02H ;Timer 0,
mode 2
    MOV P0,#0FFH ;make P0 an input port
    MOV TH0,#-92 ;TH0=A4H for -92
```

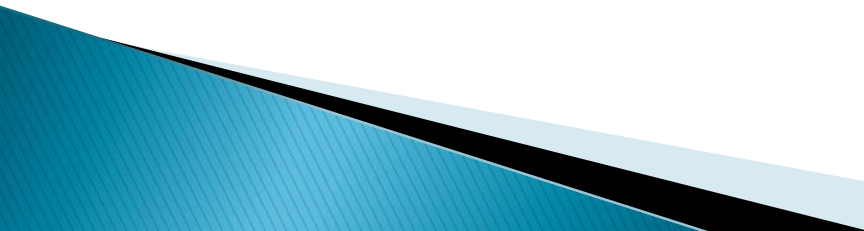


Contd.

```
MOV IE,#82H    ;IE=10000010 (bin) enable  
               ;Timer 0
```

```
SETB TR0 ;Start Timer 0
```

```
BACK:    MOV A,P0 ;get data from P0  
          MOV P1,A ;issue it to P1  
          SJMP BACK ;keep doing it loop  
                   ;unless interrupted by TF0  
END
```

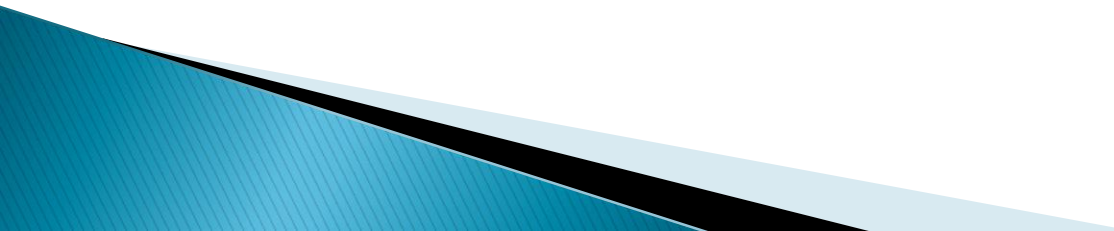


Contd.

- ▶ **Example:** Rewrite previous Example to create a square wave that has a high portion of 1085 us and a low portion of 15 us. Assume XTAL=11.0592MHz. Use timer 1.
 - ▶ **Solution:** Since 1085 us is 1000×1.085 we need to use mode 1 of timer 1.
- ;--upon wake-up go to main, avoid using
;memory allocated to Interrupt Vector Table
ORG 0000H

Contd.

```
LJMP MAIN      ;by-pass int. vector table  
;--ISR for timer 1 to generate square wave  
ORG 001BH     ;Timer 1 int. vector table  
LJMP ISR_T1    ;jump to ISR
```



Contd.

```
;--The main program for initialization
      ORG 0030H      ;after vector table space
MAIN:  MOV  TMOD,#10H ;Timer 1, mode 1
      MOV  P0,#0FFH  ;make P0 an input port
      MOV  TL1,#018H ;TL1=18 low byte of -1000
      MOV  TH1,#0FCH ;TH1=FC high byte of -1000
      MOV  IE,#88H   ;10001000 enable Timer 1 int
      SETB TR1       ;Start Timer 1
BACK:  MOV  A,P0      ;get data
      MOV  P1,A       ;issue data
      SJMP BACK       ;keep doing it
;Timer 1 ISR. Must be reloaded, not auto-reload
ISR_T1: CLR TR1       ;stop Timer 1
      MOV  R2,#4      ;
      CLR P2.1        ;P2.1=0, start of low portion
HERE:  DJNZ R2,HERE   ;4x2 machine cycle
      MOV  TL1,#18H   ;load T1 low byte value
      MOV  TH1,#0FCH  ;load T1 high byte value
      SETB TR1        ;starts timer1
      SETB P2.1       ;P2.1=1, back to high
      RETI            ;return to main
      END
```

Low portion of the pulse is created by 14 MC
 $14 \times 1.085 \text{ us} = 15.19 \text{ us}$

2MC
8MC
2MC
2MC
1MC
1MC