

# Chapter 09

## Programming in Assembly

# Lesson 06

## Programming Examples for Blinking of LEDs

# Successive sequences of ON 4000 ms and OFF 4000 ms of the LEDs

- Write program for successive overflow and interrupts for delayed actions every 4000 ms (=4s)
- Blink a red LED at P2 pin 1 using the interrupts
- 4 s ON- 4s OFF

# Successive sequences of ON 4000 ms and OFF 4000 ms of the LEDs

- Blink a green LED at P2 pin 0 using the interrupts
- 4 s OFF- 4 s ON
- Output is opposite to that at P2 pin 1 LED
- Assume that Xtal frequency is 11.0592 MHz

# Delay of 4000 ms

- When Xtal frequency is 11.0592 MHz then 922 internal clock inputs are required in 1 ms
- Let initial value in R4–R3 is hbyte–lbyte and is defined at address DFNDLY
- Delay of 4000 ms means R4–R3 = 0E-A0H.
- Mode 1 T0 is programmed for overflow and interrupt after 1 ms (922 clock inputs).

# ISR

- Counts the number of interrupts
- Stops the timer
- Calls a routine for executing the delay routine
- A and B registers save the values of R4-R3 so that these remain unaffected at the end of the program
- The delayed task used to blink the red LED and blink green LED by complement of the bit at port bit of red LED

# Set the counter in R4-R3

- DFN4s:MOV R3, #0E; Define R4-R3 = 0EA0 for 4000 ms delay
- MOV R4, #A0
- Start: ANL TMOD, #F1 ; Define T0 mode 1, internal clock inputs and internal start/stop

# Load the timer

- `MOV TH0, #0FCH ; TH0-TL0 overflows after 922 clock inputs`  
`MOV TL0, #66H`



# Enable Interrupts

- SETB EA; Enable interrupts
- MOV A, R3
- MOV R6, A; Save R3 in R6
- MOV B, R4
- MOV R7, B; Save R4 in R7
- SETB ET0; Enable T0 interrupts

# Run Timer

- `SETB TR0; Run timer T0`
- `SJMP -2; Wait forever`
- `ENDblink: ; End of the blinking`

# Interrupt Service Routine for T0

- ISRT0:MOV TH0, #0FCH ; Load FC66H in TH0-TL0 next overflow after 922 clock inputs
- MOV TL0, #66H
- DEC R6; Decrease 1byte
- CJNE R6, #FFH, ENDPT0; 1byte if not = 0 before decrement, jump to check if more delay time left

# ISR for T0

- `DEC R7; Decrease hbyte CJNE R7, #FFH, ENDPT0; hbyte if not = 0, jump to end, more delay left`

# ISR for T0

- `ACALL DlyTask ; Call delayed task`
- `MOV A, R3 ;`
- `MOV R6, A; Restore R3 in R6`
- `MOV B, R4`
- `MOV R7, B; Restore R4 in R7`
- `ENDPT0: RETI; Return from interrupt`

# Delayed task 1

- DlyTask:CPL P2^1; Complement the port bit 1 to which the LED connects
- MOV C, P2^1; Read in C the bit P2.1
- CPL C; Complement carry
- MOV P2^0, C; P2^0 green LED is complement of red LED at P2^1
- RET; Return from the routine

# Summary

# We learnt

- Red and Green LEDs Blinking Program
- Red and green LED blinks for 4000 ms, each
- Mode 1 for 0 used for 1 ms delay
- 4000<sup>th</sup> interrupt on 4000<sup>th</sup> overflow of timer T0 toggles Red LED port P2<sup>^</sup>1
- Green LED P2<sup>^</sup>0 is complement of P2<sup>^</sup>1



End of Lesson 06 on

Programming Examples for  
Blinking of LEDs