Chapter 6
Interrupts

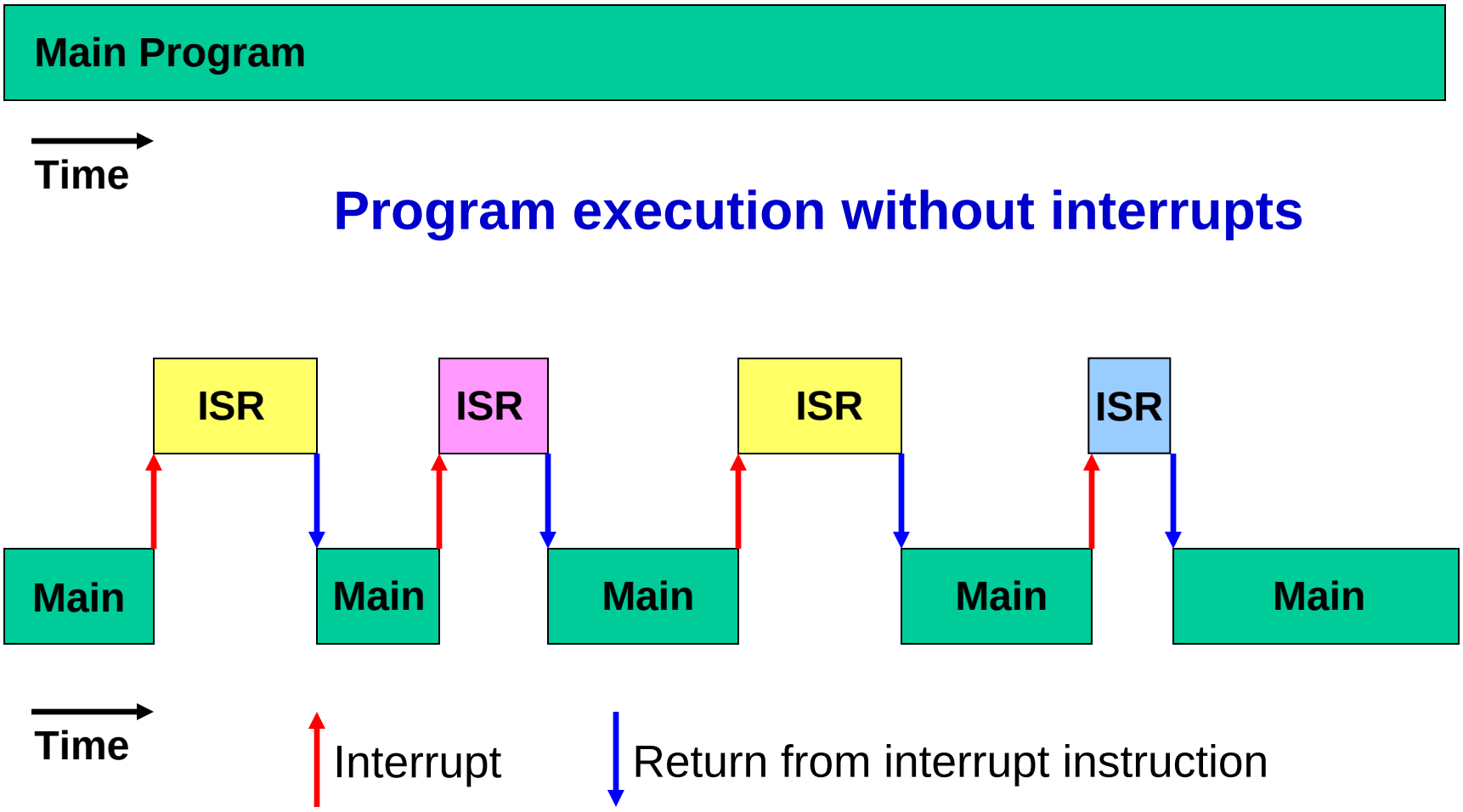
(I. Scott Mackenzie)

By: **Masud-ul-Hasan**

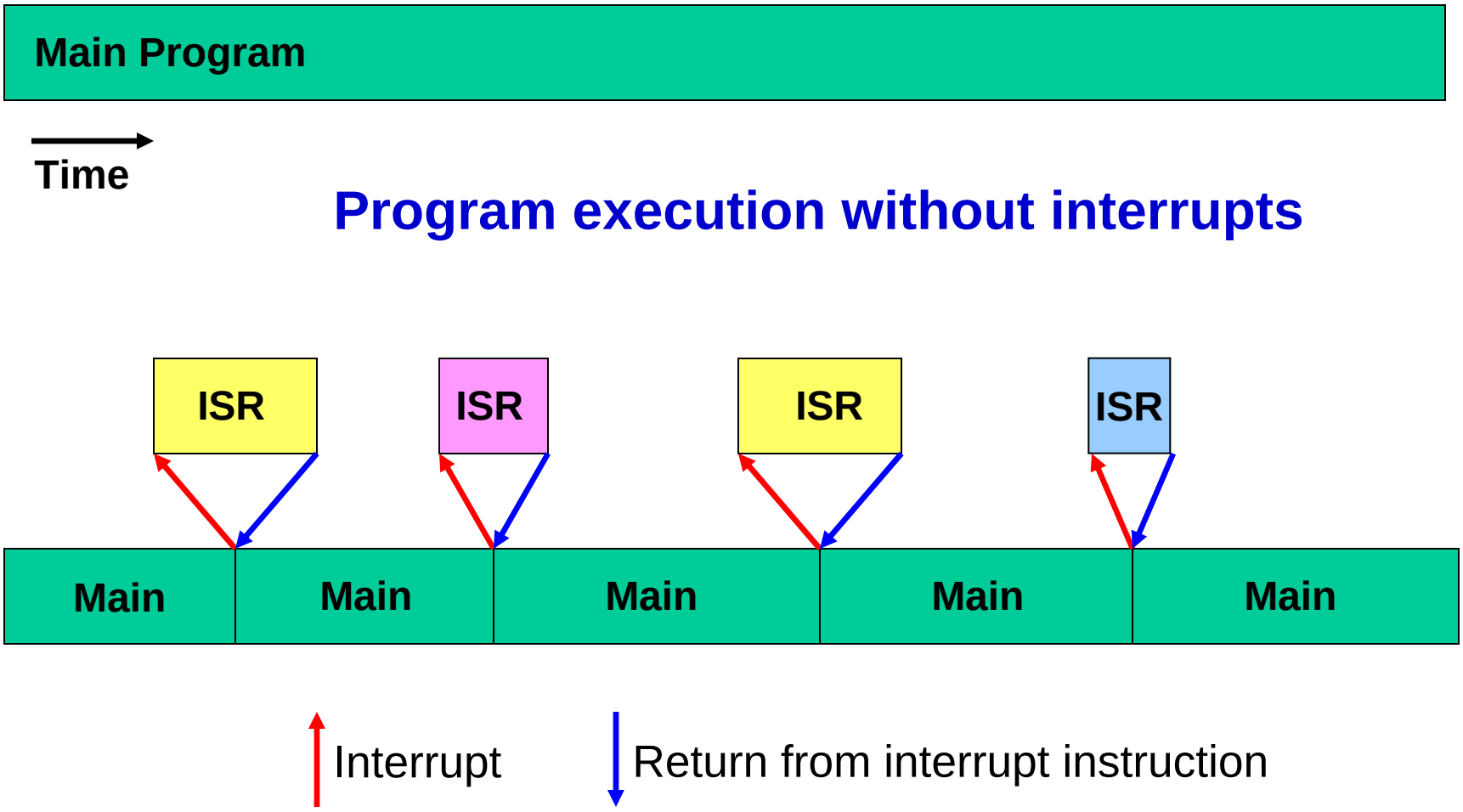
Interrupts

- An interrupt is the occurrence of an event that causes a temporary suspension of a program while the condition is serviced by another program.
- It is like a sub-routine. CPU cannot execute more than one instruction at a time; but it can temporarily suspend execution of one program, execute another, then return to the first program.
- Difference in interrupt and subroutine is that in an interrupt-driven system, the interruption occur asynchronously with the main program, and it is not known when the main program will be interrupted.
- Program that deals with the interrupt is called as **ISR (Interrupt Service Routine).**

Program Execution



Program Execution



Interrupt System

- Five interrupt sources in order of polling (priority) sequence are:

External Interrupt 0

Timer 0

External Interrupt 1

Timer 1

Serial Port

- The polling sequence is fixed but each interrupt type can be programmed to one of two priority levels.
- If two interrupts of same priority occur simultaneously then polling sequence will determine which is serviced first.
- External interrupts can be programmed for edge or level sensitivity.
- Each interrupt type has a separate vector address.
- All interrupts are disabled after a system reset and enabled individually by software.

Processing Interrupts

When an interrupt occurs and is accepted by CPU, the following actions occur:

- –Current instruction’s complete execution
- –PC is saved on the stack
- –PC is loaded with the vector address of the ISR
- –ISR executes and takes action in response to interrupt
- –ISR finishes with a RETI instruction
- –PC is loaded with its old value from the stack
- –Execution of main program continues where it left off

Interrupt Vector

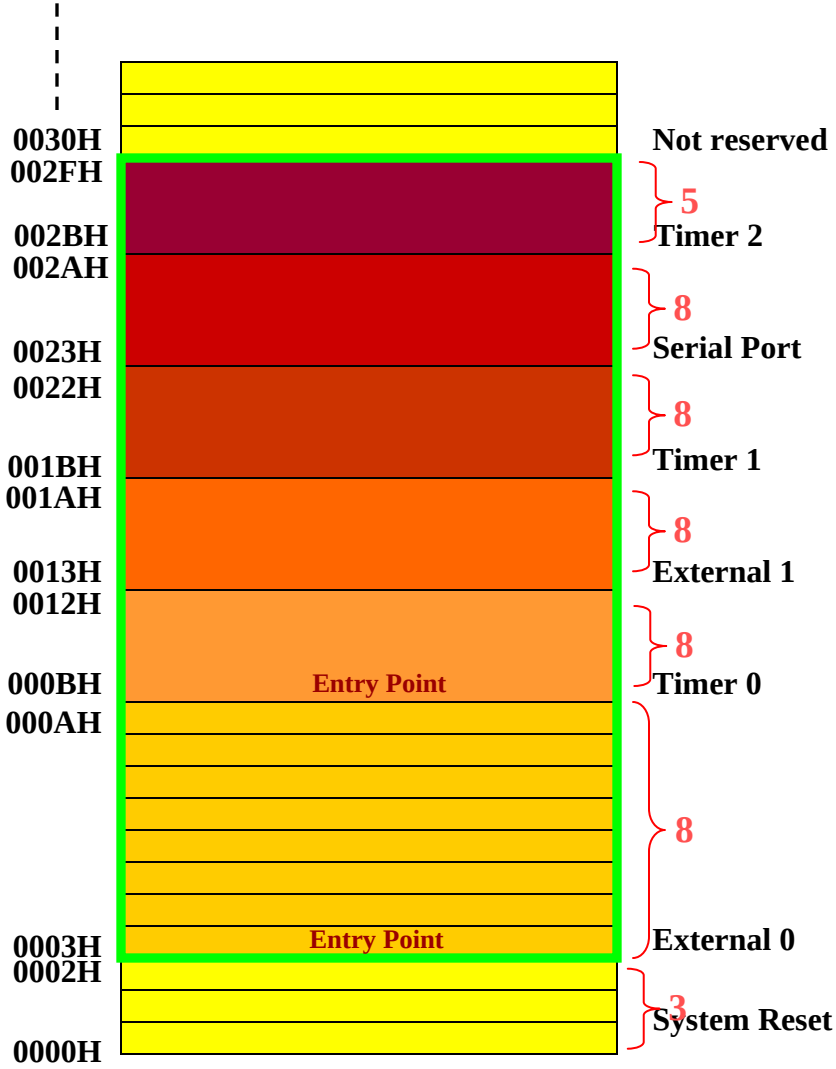
- When an interrupt is accepted, the value loaded into the PC is called the interrupt vector. It is the address of the start of the ISR for the interrupting source.
- When an interrupt is vectored, the flag that caused the interrupt is automatically cleared by hardware.
- Timer interrupts occur when the timer registers (TLx/THx) overflow and set the overflow flag (TFx).

Interrupt Vector Addresses

Interrupt	Flag	Vector Address	SFR & Bit Position
System Reset	RST	0000H	
External 0	IE0	0003H	TCON.1
Timer 0	TF0	000BH	TCON.5
External 1	IE1	0013H	TCON.3
Timer 1	TF1	001BH	TCON.7
Serial Port	RI or TI	0023H	SCON.0 or SCON.1

- Each one is 8 byte in size.

Interrupt Vector Addresses



By: **Masud-ul-Hasan**

IE : Interrupt Enable Register (0A8H)

EA	----	----	ES	ET1	EX1	ET0	EX0

(1 = Enabled, 0 = Disabled)

- EA : Global interrupt enable/ disable.
- EX0: External interrupt 0 enable/ disable.
- ET0: Timer 0 interrupt enable/ disable.
- EX1: External interrupt 1 enable/ disable.
- ET1: Timer 1 interrupt enable/ disable.
- ES : Serial port interrupt enable/ disable.

e.g., Timer 1 interrupt can be enabled as follows:

```
SETB EA          ; Enable global interrupt bit
SETB ET1         ; Enable Timer 1 interrupt
```

Or `MOV IE, #10001000B`

IP: Interrupt Priority Register (0B8H)

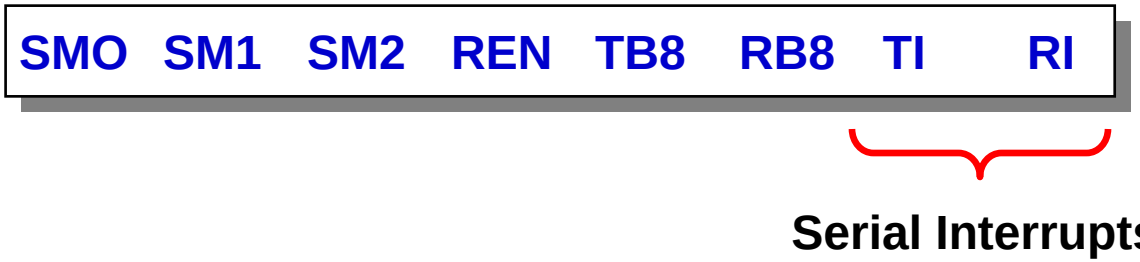
-----	-----	-----	PS	PT1	PX1	PT0	PX0

(1 = High priority, 0 = Low priority)

- PX0 : Priority for External interrupt 0.
- PT0 : Priority for Timer 0 interrupt.
- PX1 : Priority for External interrupt 1.
- PT1: Priority for Timer 1 interrupt.
- PS : Priority for Serial port interrupt.

IP is cleared after a system reset to place all interrupts at the lower priority level by default.

SCON : Serial Port CONtrol Register (098H)

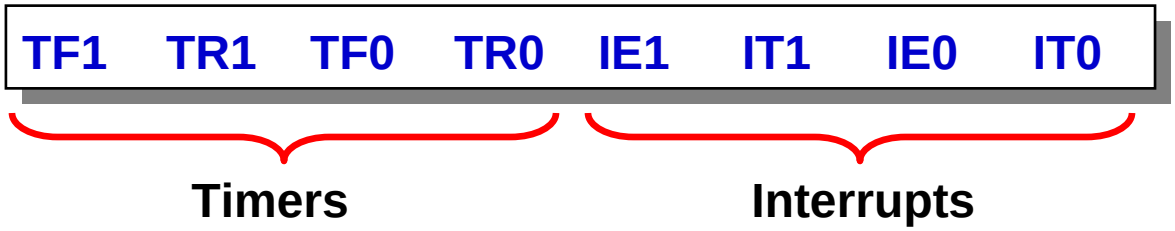


- **TI:** Transmit interrupt flag. Set at the end of character transmission; cleared by software.
- **RI:** Receive interrupt flag. Set at the end of character reception; cleared by software.

External Interrupts

- External interrupt occurs as a result of a low-level or negative-edge on the $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ pin of 8051.
- Flags that generate these interrupts are bits IE0 and IE1 in TCON. These are automatically cleared when the CPU vectors to the interrupt.
- Low-level or negative-edge activated interrupts can be programmed through IT0 and IT1 bits in TCON, i.e., ITx = 0 means low-level and ITx = 1 means negative-edge triggered.

TCON : Counter/Timer CONtrol Register (088H)



- TF1, TF0 : Overflow flags for Timer 1 and Timer 0. *
- TR1, TR0 : Run control bits for Timer 1 and Timer 0. *
  Set to run, reset to hold.

- IT1, IT0 : Type bit for external interrupts.
  Set for falling edge interrupts, reset for low level interrupts.
- IE1, IE0 : Flag for external interrupts 1 and 0.
  Set by interrupt, cleared when interrupt is processed.

* = related to counter/timer operation

10KHz Square Wave on P1.0 Using Timer Interrupts

```
        ORG   0000H        ;reset entry point
        LJMP  Main
        ORG   000BH        ;T0 interrupt vector
T0ISR:  CPL   P1.0         ;toggle port bit
        RETI
        ORG   0030H
Main:   MOV   TMOD,#02H    ;T0 MODE 2
        MOV   TH0,#-50     ;50 ms DELAY
        SETB  TR0          ;START TIMER 0
        MOV   IE,#82H      ;ENABLE T0 INT
        SJMP  $            ;DO NOTHING
```

Two 7KHz & 500Hz Square Waves Using Interrupts

```
        ORG   0              ;reset entry point
        LJMP  Main
        ORG   000BH          ;T0 interrupt vector
        LJMP  T0ISR
        ORG   001BH          ;T1 vector
        LJMP  T1ISR
        ORG   0030H          ;main starts here
Main:   MOV   TMOD,#12H      ;T0 mode 2, T1 mode 1
        MOV   TH0,#-71       ;7-kHz using T0
        SETB  TR0            ;START TIMER 0
        SETB  TF1            ;force T1 int
        MOV   IE,#8AH        ;ENABLE T0,T1 INT
        SJMP  $              ;DO NOTHING
```

Two 7KHz & 500Hz Square Waves Using Interrupts

(Continued)

```
T0ISR:   CPL   P1.7                ;7-kHz on P1.7
         RETI
T1ISR:   CLR   TR1                 ;stop T1
         MOV   TH1,#HIGH(-1000)
         MOV   TL1,#LOW(-1000)     ;1 ms for T1
         SETB  TR1                 ;START TIMER 1
         CPL   P1.6                ;500-Hz on P1.6
         RETI
```

Furnace Controller

$\overline{\text{HOT}}$=0 if T>21 $\overline{\text{INT0}}$ (P3.2)
$\overline{\text{COLD}}$ =0 if T< 19 $\overline{\text{INT1}}$ (P3.3)

```
                ORG   0000H
                LJMP  MAIN          ; 3-bytes long
EX0ISR:         CLR   P1.7          ; EX0 vector at 0003H
                RETI                ; turn furnace OFF
                ORG   0013H
EX1ISR:         SETB  P1.7          ; turn furnace ON
                RETI
                ORG   0030H
MAIN:           MOV   IE, #85H      ; enable external interrupts
                SETB  IT0           ; negative edge triggered
                SETB  IT1
                SETB  P1.7          ; turn furnace ON initially
                JB    P3.2, SKIP    ; if T>21°, P3.2 is zero
                CLR   P1.7          ; turn furnace OFF
SKIP:           SJMP   $            ; do nothing
                END
```