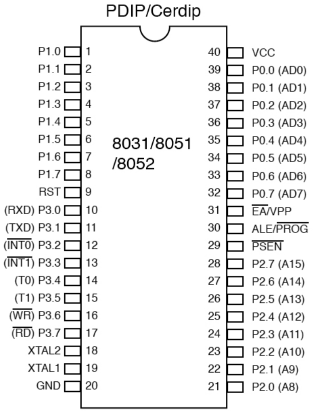
8051 Microcontroller: Interrupts



Virendra Singh

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

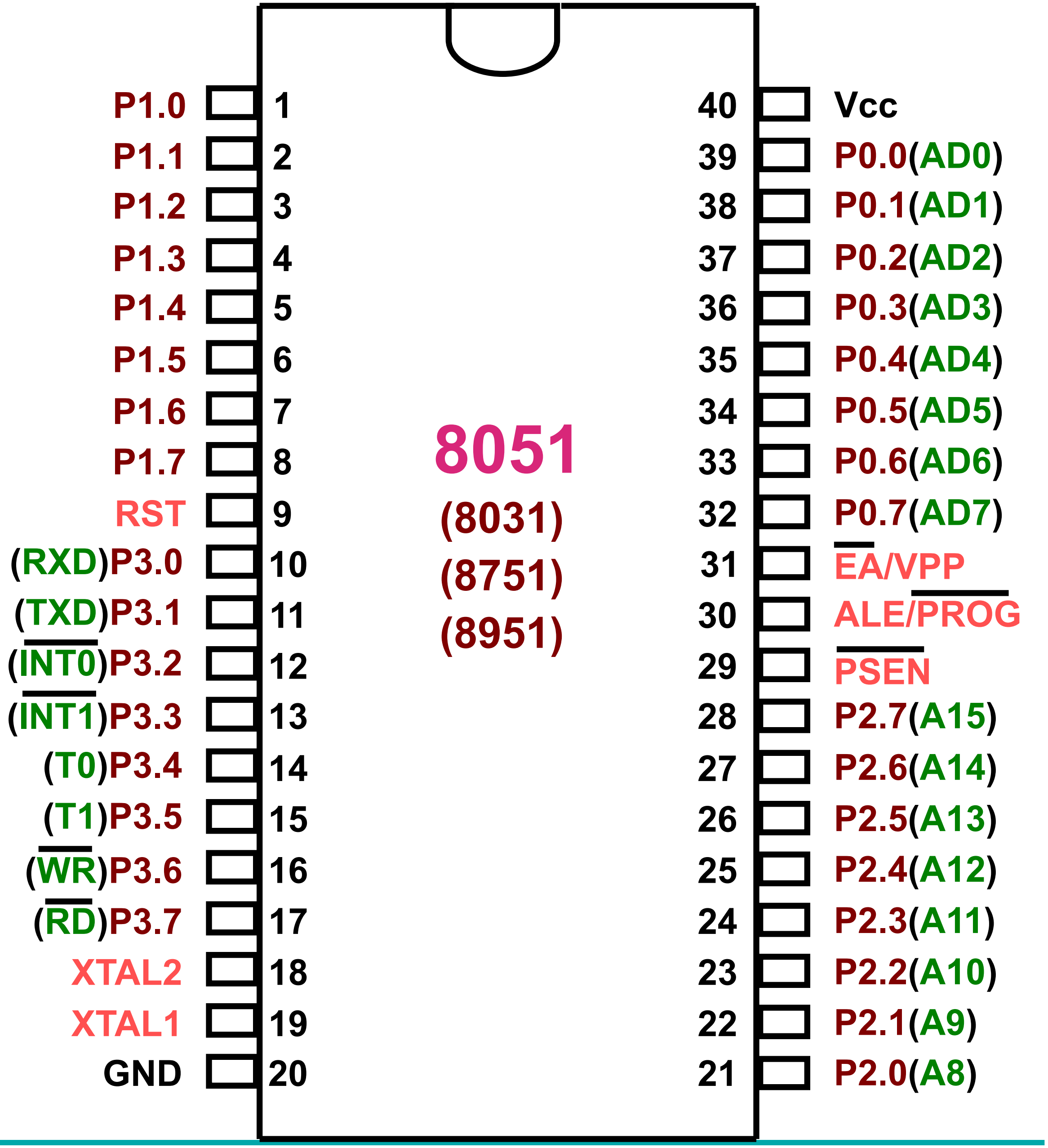
http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

EE-309: Microprocessors

Lecture 10 (10 Aug 2015)

8051 Pin Diagram



8051

(8031)

(8751)

(8951)

P1.0	1	40	Vcc
P1.1	2	39	P0.0(AD0)
P1.2	3	38	P0.1(AD1)
P1.3	4	37	P0.2(AD2)
P1.4	5	36	P0.3(AD3)
P1.5	6	35	P0.4(AD4)
P1.6	7	34	P0.5(AD5)
P1.7	8	33	P0.6(AD6)
RST	9	32	P0.7(AD7)
(RXD)P3.0	10	31	$\overline{\text{EA}}$/VPP
(TXD)P3.1	11	30	ALE/$\overline{\text{PROG}}$
($\overline{\text{INT0}}$)P3.2	12	29	$\overline{\text{PSEN}}$
($\overline{\text{INT1}}$)P3.3	13	28	P2.7(A15)
(T0)P3.4	14	27	P2.6(A14)
(T1)P3.5	15	26	P2.5(A13)
($\overline{\text{WR}}$)P3.6	16	25	P2.4(A12)
($\overline{\text{RD}}$)P3.7	17	24	P2.3(A11)
XTAL2	18	23	P2.2(A10)
XTAL1	19	22	P2.1(A9)
GND	20	21	P2.0(A8)

Interrupt

- An interrupt is the occurrence of a condition--an event -- that cause a temporary suspension of a program while the event is serviced by another program (Interrupt Service Routine ISR or Interrupt Handler).
- Interrupt is an asynchronous event
- Interrupt-Driven System-- gives the illusion of doing many things simultaneously, quick response to events, suitable for real-time control application.

Interrupt Process

If interrupt event occurs AND interrupt flag for that event is enabled, AND interrupts are enabled, then:

1. Current PC is pushed on stack.
2. Program execution continues at the interrupt vector address for that interrupt.
3. When a RETI instruction is encountered, the PC is popped from the stack and program execution resumes where it left off.

Interrupt Sources

- Original 8051 has 5 sources of interrupts
  - Timer 0 overflow
  - Timer 1 overflow
  - External Interrupt 0
  - External Interrupt 1
  - Serial Port events (buffer full, buffer empty, etc)
- Enhanced version has 22 sources
  - More timers, programmable counter array, ADC, more external interrupts, another serial port (UART)

Interrupt Vectors

Each interrupt has a specific place in code memory where program execution (interrupt service routine) begins.

External Interrupt 0:	0003h
Timer 0 overflow:	000Bh
External Interrupt 1:	0013h
Timer 1 overflow:	001Bh
Serial :	0023h

Note: that there are only 8 memory locations between vectors.

Interrupt Execution

- Each Interrupt can be individually enabled or disabled

EA	X	X	ES	ET1	EX1	ET0	EX0

- Each source of interrupt can be programmed to one of two priority levels

X	X	X	PS	PT1	PX1	PT0	PX0

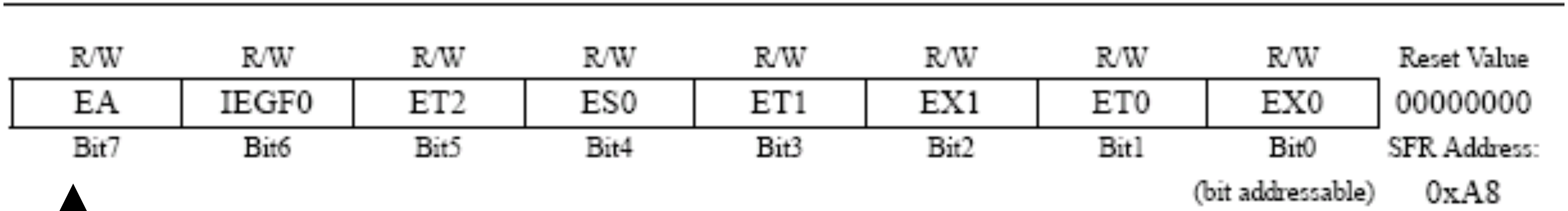
- Priority of an interrupt under execution should be stored

Interrupt Priorities

- What if two interrupt sources interrupt at the same time?
- The interrupt with the highest PRIORITY gets serviced first.
- All interrupts have a default priority order.
- Priority can also be set to “high” or “low”.

Interrupt SFRs

Figure 12.9. IE: Interrupt Enable



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	IEGF0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xA8

Interrupt enables for the 5 original 8051 interrupts:
Timer 2
Serial (UART0)
Timer 1
External 1
Timer 0
External 0

Global Interrupt Enable – must be set to 1 for any interrupt to be enabled

1 = Enable
0 = Disable

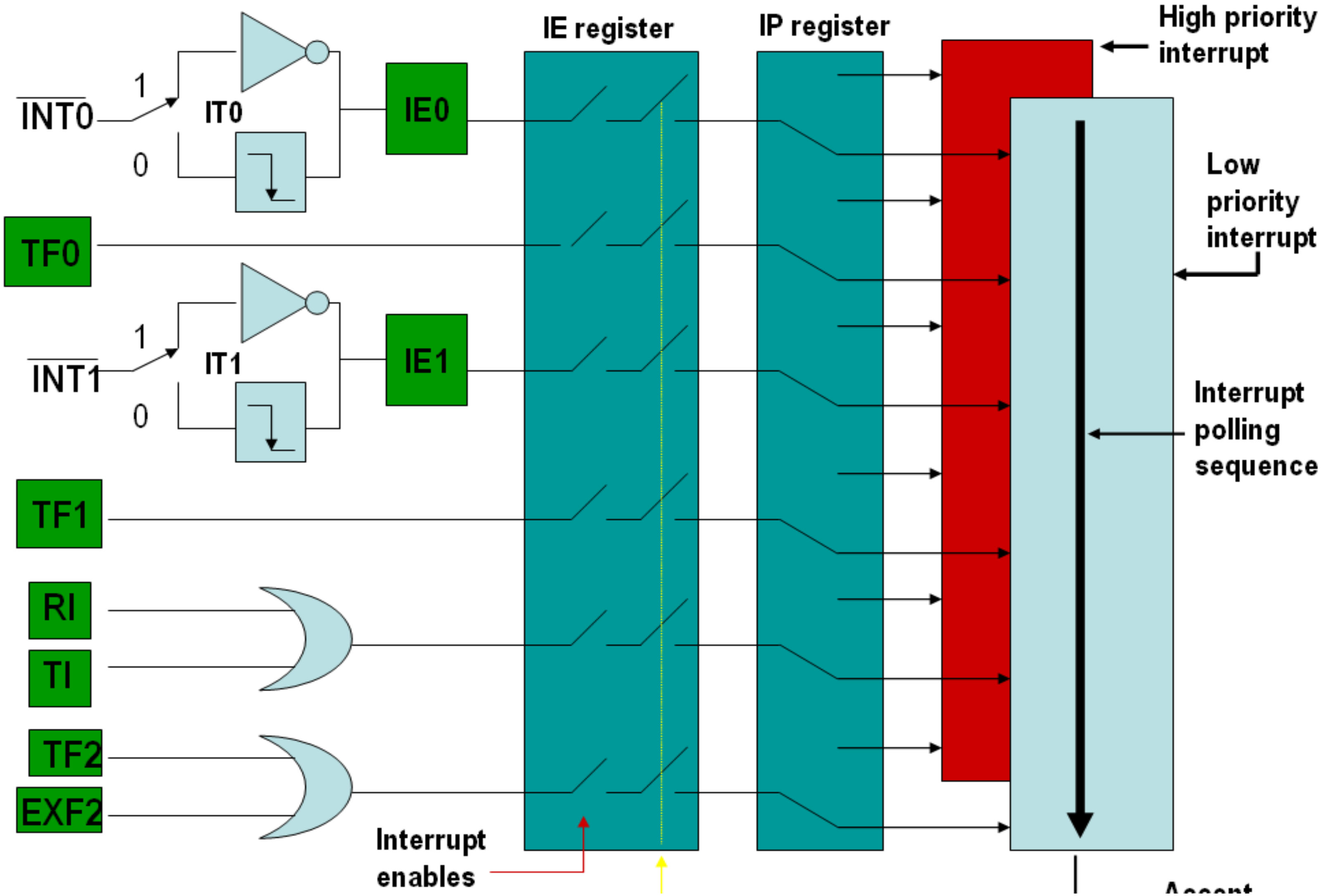
Interrupt Flag Bits

Interrupt	Flag	SFR Register & Bit Position
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial port	TI	SCON.1
Serial Port	RI	SCON.0

The state of all interrupt sources is available through the respective flag bits in the SFRs.

If any interrupt is disabled, an interrupt does not occur, but software can still test the interrupt flag.

Interrupt Control



Application: LED

```
            ORG 0
            LJMP    MAIN
            ORG 0003H
LED1:       MOV P0, #0FFH
            MOV R0, #255
            DJNZ R0, LED1
            RETI
            ORG 0013H
LED2:       MOV P2, #0FFH
            MOV R0, #255
            DJNZ R0, LED1
            RETI
            ORG 0030HH
MAIN:       MOV IE, #85H     ;enable INT0 & INT1
AGAIN       SJMP AGAIN
```

Application: Square wave Generator

```
            MOV TMOD, #01
LOOP:       MOV TL0, #0EEH
            MOV TH0, #0FFH
            CPL P1.0
            ACALL DELAY
            SJMP LOOP
DELAY:      SETB TR0
AGAIN:      JNB TF0, AGAIN
            CLR TR0
            CLR TF0
            RET
```

Application: Square wave Generator

```
            ORG 0
            LJMP    MAIN
            ORG 000BH
            CPL P2.0
            RETI
            ORG 0040H
MAIN:       MOV TMOD, #02H
            MOV TH0, #-92
            MOV IE, #82H
            SETB TR0
AGAIN:      MOV A, P1
            MOV P0, A
            SJMP AGAIN
```

Application: External Events

- Counting external events on P3-5

```
        MOV  TMOD, #01100000B
        MOV TH1, #0
        SETB P3.5
AGAIN:  SETB TR1
BACK:   MOV A, TL1
        MOV P1, A
        JNB TF1, BACK
        CLR TR1
        CLR TF1
        SJMP AGAIN
```

Hybrid Approach

- Two 16 bit timer/counter section

TH1	TL1

  - MOV TL0, #FEH
- TCON Register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- TMOD Register

GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0

Thank You