NMK322 - Microcontroller

Lecture 07 – 8051 Interrupt

Interrupt

- Event that triggers specific code execution
  - suspends current execution (override)
  - similar to subroutine flow (only event-triggered)
  - microprocessor hardware feature
- Code to handle interrupt: interrupt handler
  - @ interrupt service routine (ISR)
  - @ event handler
- Event-driven systems appear to multi-task
  - code execution still 1-at-a-time
  - basis for software threads
- Events can be software (internal feature) and/or hardware (external signal)

Case study: Microwave Oven

- running embedded system
  - main code handles power element
  - user presses a key (e.g. cancel or modify duration)
  - interrupt handler for key event is triggered
    - interrupts main code
    - fulfill (@service) the request
    - return control to main code
  - main code continues…
- note: handler is not triggered in main code
  - random event that may occur at any time
  - what if no interrupt feature? →poll!

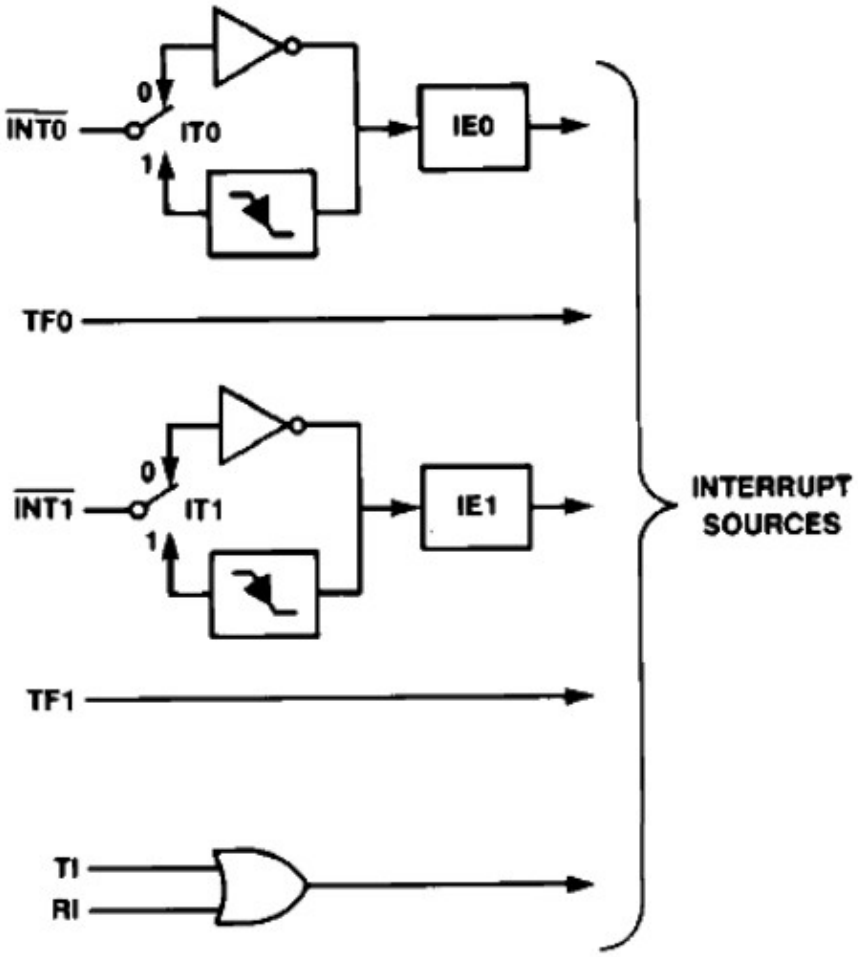
Polling vs. Interrupt

- Polling
  - CPU continuously check service request (@flag)
  - Handle the request when found
  - Spend most of processing time waiting for event!
    - Not so bad if event is periodic @ expected
- Interrupt
  - CPU can idle or do other task(s)
  - When event(s) occur, handler code triggered
  - Free to spend processing time on other task(s)

8051 Interrupt

- 5 interrupt sources:
  - 2 external interrupts (INT0 & INT1)
  - 2 timer interrupts (T0 & T1)
  - 1 serial interrupts (OR'ed logic TI & RI)
- Interrupt trigger bits can be set / cleared by software
  - IE0,IE1 for INT0,INT1 (TCON)
  - TF0,TF1 for T0 T1 (TCON)
  - TI,RI for Serial (SCON) [not cleared by HW!]
- Each source can be individually {en,dis}abled
  - using IE register (@0xA8)

8051 Interrupt: Trigger Bits

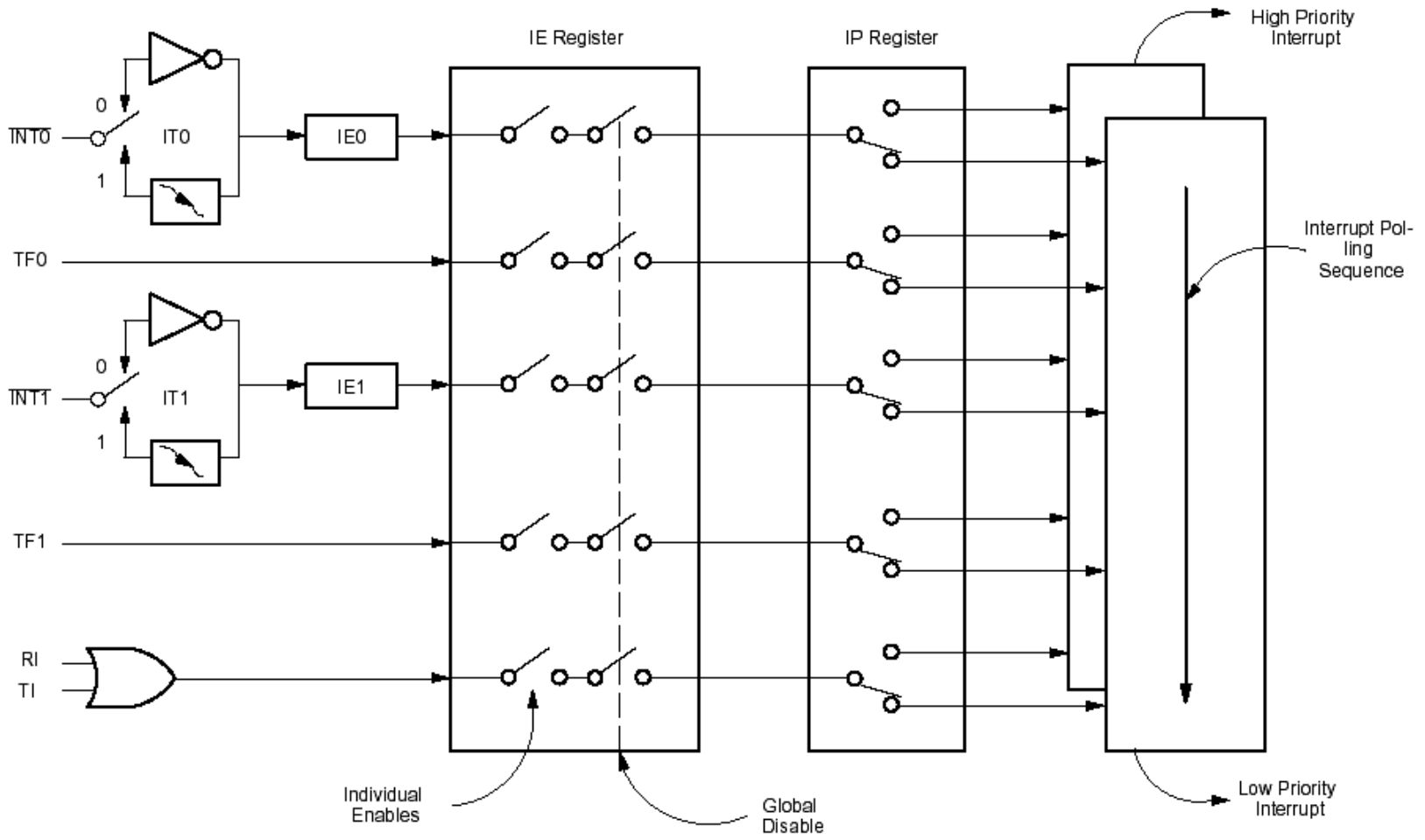


Interrupt Enable Register (IE)

MSB							LSB
EA	-	ET2	ES	ET1	EX1	ET0	EX0

Bit	Name	Description
IE.7	EA	Enable/Disable all interrupts If 0 all interrupts are disabled. If 1, interrupts are enabled based on their individual bits
IE.6	-	Reserved
IE.5	ET2	Enable/Disable Timer 2 interrupt (8052)
IE.4	ES	Enable/Disable Serial Input Interrupt
IE.3	ET1	Enable/Disable Timer 1 Interrupt (TF1)
IE.2	EX1	Enable/Disable External Interrupt 1 (INT1)
IE.1	ET0	Enable/Disable Timer 0 Interrupt (TF0)
IE.0	EX0	Enable/Disable External Interrupt 0 (INT0)

8051 Interrupt Structure



8051 Interrupt Priority

- 2 priority levels
  - High priority: cannot be interrupted by other interrupt source
    - set in IP register (@0xB8)
  - Low priority: (default) can be interrupted by high-priority interrupt source
- Simultaneous request?
  - internal polling sequence: IE0,TF0,IE1,TF1,(TI|RI)
  - only applies on same priority levels

Interrupt Priority Register (IP)

MSB							LSB
-	-	PT2	PS	PT1	PX1	PT0	PX0

Bit	Name	Description
IP.7	-	Reserved
IP.6	-	Reserved
IP.5	PT2	Timer 2 interrupt priority (8052)
IP.4	PS	Serial Port Interrupt priority
IP.3	PT1	Timer 1 Interrupt priority (TF1)
IP.2	PX1	External Interrupt 1 priority (INT1)
IP.1	PT0	Timer 0 Interrupt priority (TF0)
IP.0	PX0	External Interrupt 0 priority (INT0)

Response Time

- Time taken to start handler code execution
  - from actual interrupt event
- For 8051: min. 3 machine cycles, max. 9 machine cycles
  - hint: 1 machine cycle is 12 clock cycles
- Blocking conditions:
  - interrupt already in progress (higher or equal)
  - not final cycle of current instruction in progress
  - RETI or any writes to IE/IP registers

Pending Interrupt

- Blocked or disabled interrupts
  - trigger bits set, but not handled
- Will be handled on next polling or enabled
  - as long as the trigger bits stay set
- Can be ‘canceled’ by software
  - clear the trigger bits

Activation Levels: INT0 / INT1

- INT0 / INT1 trigger type can be configured
  - use IT0 / IT1 bits in TCON
- Level-triggered (default , ITx=0)
  - Logic ‘0’ on INTx triggers interrupt directly
- Edge-triggered (ITx=1)
  - ‘1’ → ‘0’ transition on INTx triggers interrupt (IEx)
  - Can be ‘emulated’ by software (IEx = 1)
  - IEx cleared by hardware on RETI

Activation Levels: T0 / T1

- Basically edge-triggered
  - Counter overflow signal (0xFFFF → 0x0000)
- Trigger bits (TF0 / TF1)
  - cleared by hardware when handler started

Interrupt Vector Table (IVT)

Symbol	Address	Interrupt Source
RESET	00H	Power Up or Reset
EXTI0	03H	External Interrupt 0
TIMER0	0BH	Timer 0 Interrupt
EXTI1	13H	External Interrupt 1
TIMER1	1BH	Timer 1 Interrupt
SINT	23H	Serial Port Interrupt

8051 Code: Event Counting

- Event (negative edge) counting using INT0
  - set count (8-bit) value to P1

```
#include <reg51.h>
unsigned char ecnt;
void count(void) interrupt 0 {
  ecnt++;
}
void main(void) {
  ecnt = 0; IT0 = 1;
  EX0 = 1; EA = 1;
  while (1) {
    P1 = ecnt;
  }
}
```

8051 Code: Pulse Generator

- Generate 500Hz squarewave on P1.0
  - assume all other bits on P1 are not used

```
#include <reg51.h>

void wave(void) interrupt 1 {
  TH0 = 0xfc; TL0 = 0x66;
  P1 ^= 0x01;
}
void main(void) {
  P1  = 0x00; TMOD = 0x11;
  TH0 = 0xfc; TL0 = 0x66;
  ET0 = 1; EA = 1;
  while (1) {
  }
}
```

Code51 Task: Generate alternating blinking LEDs (1Hz) connected to P2

End of Lecture07