Interrupts

Overview

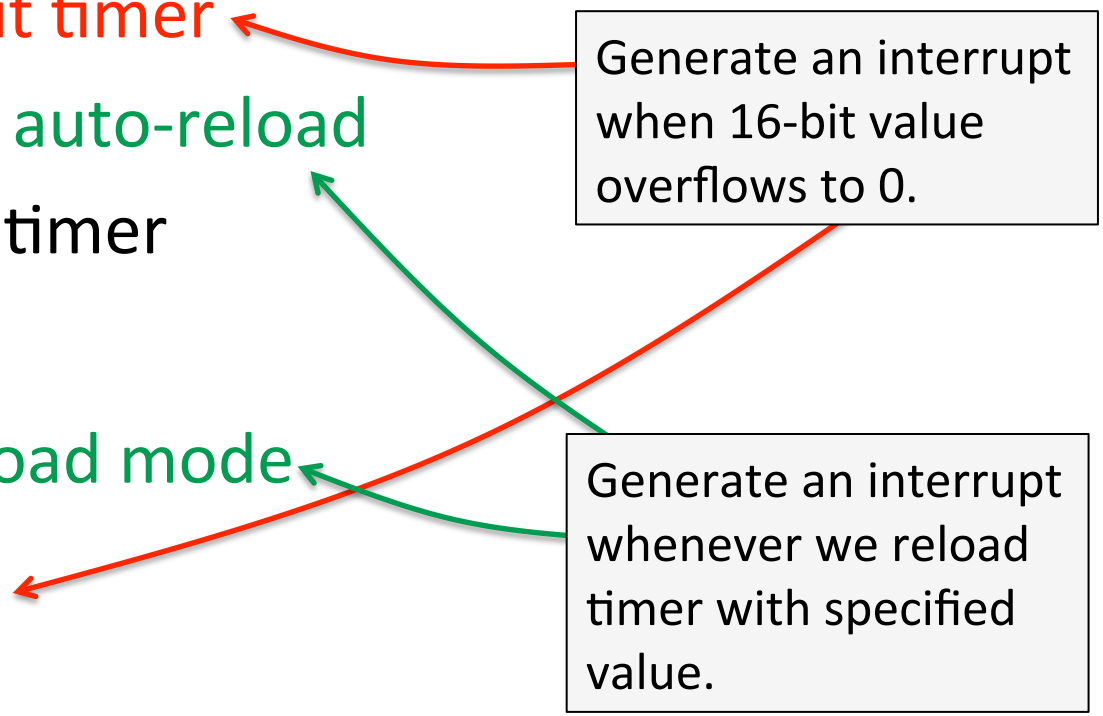
- Interrupts
  - Allow concurrent processing
    - Main program runs
    - Periodically event triggers an interrupt
    - Interrupt Service Routine (ISR) handles event
    - Returns control to main program
  - Advantages
    - Cleaner code
    - Less wasteful than continually polling for event
    - Allow prioritization of events

Interrupt events

- Events that can trigger an 8052 interrupt:
  - Timer 0 overflow/reload
  - Timer 1 overflow/reload
  - Timer 2 overflow/reload
  - Reception/transmission of serial data
  - External event 0
  - External event 1

Timer interrupts

- Timer0, Timer1
  - Mode 0 = 13 bit timer (legacy)
  - Mode 1 = 16-bit timer
  - Mode 2 = 8-bit auto-reload
  - Mode 3 = Split timer
- Timer 2
  - 16-bit auto-reload mode
  - Capture mode

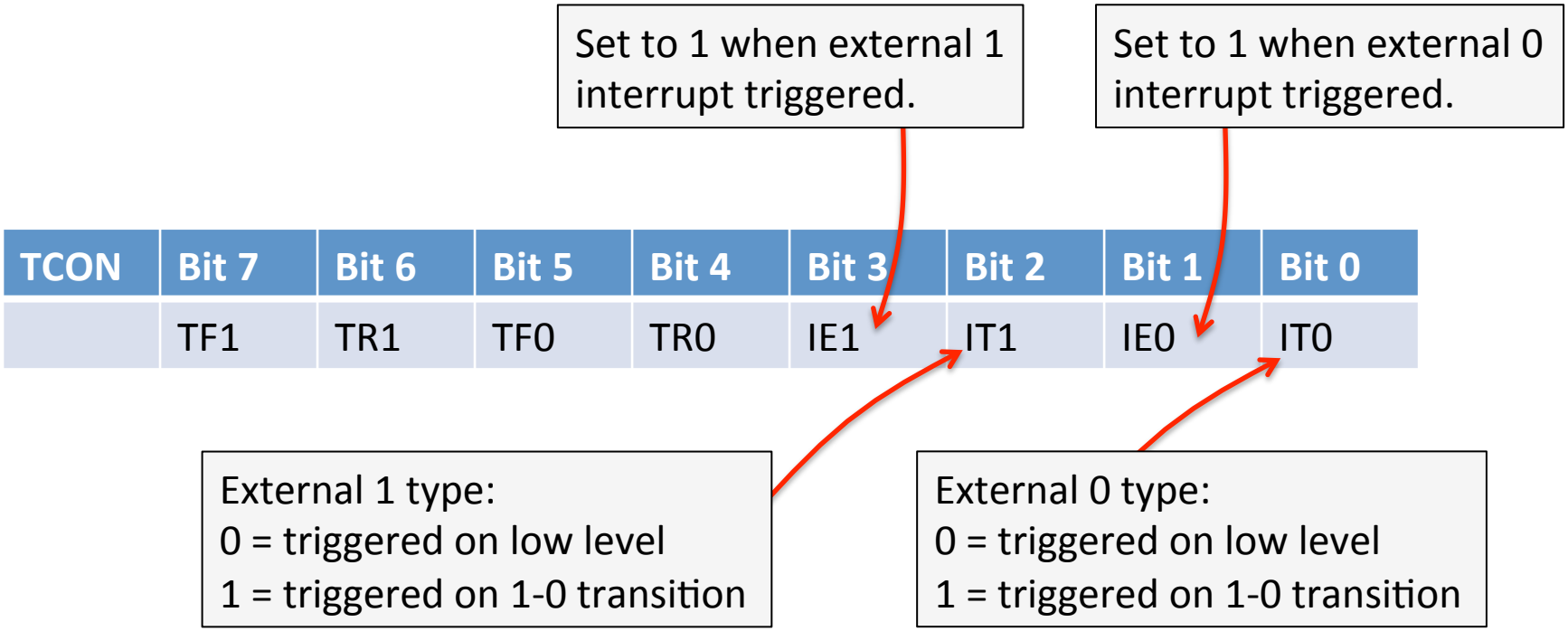


Serial interrupts

- Serial port
  - Receives or transmits data on two wires
  - Sets flag bit RI when byte received
  - Sets flag bit TI when byte transmitted
  - Either flag bit being set can be configured to generate an interrupt

External events

- External 0/1 events
  - Monitor level on pin INT0 (P3.2) and INT1 (P3.3)
  - Event = 1-0 transition or low-level



Interrupt Service Routine

- When an interrupt triggers:
  - Jumps to specific spot in code memory
  - Each is 8 bytes apart
  - ORG directive

Interrupt	Address
External 0	0003h
Timer 0	000Bh
External 1	0013h
Timer 1	001Bh
Serial	0023h
Timer 2	002Bh

```
ORG 0000h
JMP Main
ORG 0003h
JMP External0ISR
ORG 000Bh
JMP Timer0ISR
ORG 0013h
JMP External1ISR
ORG 001Bh
JMP Timer1ISR
ORG 0023h
JMP SerialISR
ORG 002Bh
JMP Timer2ISR
```

Gets your main program safety away from the ISR addresses.

Setting up interrupts

- Must enable interrupts globally
  - IE SFR, set EA bit to 1
- Must enable type(s) you want triggered
  - IE SFR, ES/ET1/EX1/ET0/EX0 bit to 1

IE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	EA	-	-	ES	ET1	EX1	ET0	EX0

Must set to 1 for any interrupts to fire

Enable serial

Enable timer1

Enable external1

Enable timer0

Enable external0

Polling sequence

- 8052 polls events in fixed order:
  - External 0
  - Timer 0
  - External 1
  - Timer 1
  - Serial
  - Timer 2
- In the event of a tie, first one in list goes first

Interrupt priorities

- Interrupt events marked high or low priority
  - High priority ISR run until finished
  - Low priority ISR can be interrupted by high priority interrupt
  - If event at same time, high priority goes first

Setting priorities

- Set bits in IP SFR
  - 0 = low priority, 1 = high priority

IP	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	-	-	-	PS	PT1	PX1	PT0	PX0

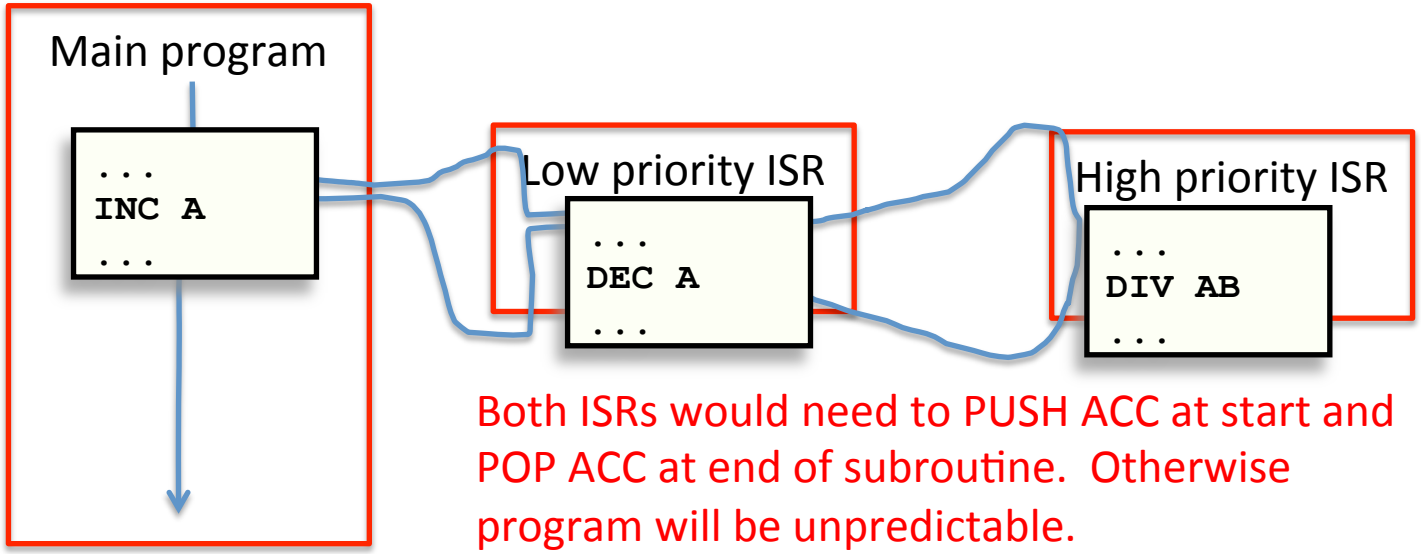
Triggering process

1. Program Counter (PC) saved to stack
2. Interrupts of same/lower priority disabled
3. Timer and external interrupts, clears flag bit
4. PC changed to ISR address
5. ISR executes
6. RETI returns from ISR
   - Pops stack to restore PC
   - Re-enable interrupts

Not RET!

Protecting state

- Don't know when program will be interrupted
  - If main program and ISR use registers, accumulator, PSW, etc.
    - Must PUSH state at start of ISR
    - Must POP state at end of ISR



Both ISRs would need to PUSH ACC at start and POP ACC at end of subroutine. Otherwise program will be unpredictable.

Protection example

```
; Timer 2 overflow ISR
ORG 002Bh                       ; Address of ISR
JMP Timer2ISR                   ; Jump somewhere so ISR > 8 bytes

Timer2ISR:
     PUSH ACC                   ; Save off accumulator
     PUSH PSW                   ; Save off PSW
     MOV  A, R0                 ; Save off R0, step 1
     PUSH ACC                   ; Save off R0, step 2

     ; Do stuff using ACC, PSW, and R0

     POP ACC                    ; Restore R0, step 1
     MOV R0, A                  ; Restore R0, step 2
     POP PSW                    ; Restore PSW
     POP ACC                    ; Restore accumulator
     RETI                       ; Return from ISR
```

Summary

- Interrupts
  - Allow concurrent processing
  - Based on interrupt signal
    - From timer
    - From receiving/transmitting data
    - From external signal