Lecture 20:
Interrupt Devices

Today’s Goals

- Use an **edged-triggered** interrupt capable device to cause interrupts.
- Use a **time-based** interrupt capable device to cause periodic interrupts.

Edge-triggered Device

- Edge-triggered
  - When using ports as general-purpose I/O, we continuously sample the input value *until detecting a transition from one value to another*.
- Edge-triggered devices
  - Devices that can be configured to watch for that type of transition in the background.
  - And can generate an interrupt when the transition occurs.

Port H as Edge-triggered

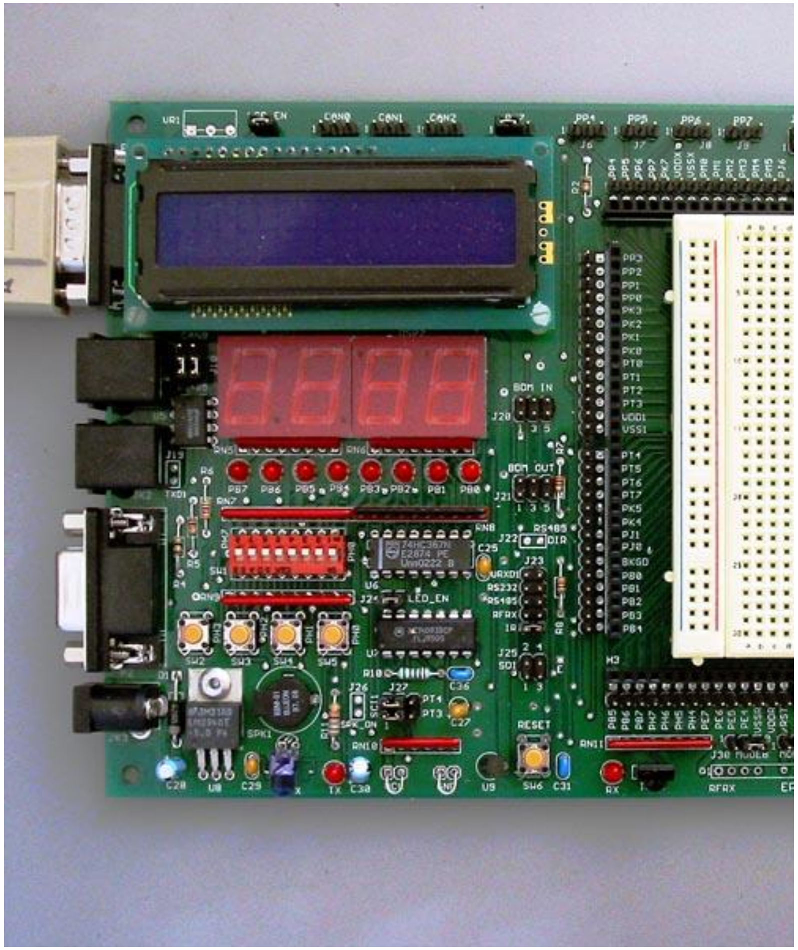
- Port H interrupt
  - Each pin of Port H can be individually configured to generate an interrupt on an edge.
  - However, all pins generate a generic Port H interrupt.
  - Therefore, the ISR needs to determine which of the 8 pins triggered the interrupt.
- Enable/Flag for Port H
  - **PIEH**
  - **PIFH**
- Interrupt Vector
  - The address of the ISR for Port H: $FFCC ($3E4C for Dragon12+)

Example of an Interrupt-driven Program

- Write an interrupt-driven program that displays a digit on the 7 segment display.
  - Pressing BUTT3 should **increment** the value by 1.
  - Pressing BUTT2 should **decrement** the value by 1.
  - *Assume the 7-segment subroutines are available.*
    - *INIT7SEG*
    - *DISP7SEG*

Review: Port H in Dragon 12+

- The port H is used to read the 8-DIP switch and the 4 push buttons
  - 2. Pressing a button/flipping a switch pulls the input voltage low.



Example - constants

A Part of "d12plus.inc"

```
;--------------------------------------------------------------
; Interrupts
PIEH      equ  $0266     ; Port Interrupt Enable for H
PIFH      equ  $0267     ; Port Interrupt Flag for H

;--------------------------------------------------------------
; Interrupt vector table
IVEC_PORTH     equ  $3E4C     ; Actual address is $FFCC
...
```

```
#INCLUDE d12plus.inc

;==============================================================
; Addresses of the subroutines for 7 segment LED digit display
;--------------------------------------------------------------
; Initialize 7 segment LED digits
INIT7SEG     EQU    PROGSTART+$800

;--------------------------------------------------------------
; Display a 7 segment LED digit
; Input:
;      A: which of 4 digits
;      B: ASCII number
DISP7SEG     EQU    PROGSTART+$880

;==============================================================
; Set an ISR to the interrupt vector table
             ORG    IVEC_PORTH
             DC.W   ISR_PSHBUTTN
```

Example – Main

```
          ORG  DATASTART
ASCIINUM  DS.B 4

          ORG  PROGSTART
          LDS  #$3600         ; Init Stack

          ;-------------------------------------------
          ; Init ports
          MOVB #DINP, DDRH
          ; Enable interrupts on BUTT3 and BUTT2
          ; Disable interrupts on all other pins
          MOVB #(BUTT3+BUTT2), PIEH

          ;-------------------------------------------
          ; Initialize a number to display
          MOVB #'0', ASCIINUM
          ;-------------------------------------------
          JSR  INIT7SEG
          ;-------------------------------------------
          ; Enable interrupts
          CLI
          ;-------------------------------------------
          ; and wait for things to happen
LOOP:     LDAA #DIGIT3
          LDAB ASCIINUM
          JSR  DISP7SEG
          BRA  LOOP
```

Example – ISR

```
;------------------------------------------------------------
; Interrupt service routine
;------------------------------------------------------------
             ORG    PROGSTART+$300
ISR_PSHBUTTN BRCLR  PIFH, BUTT3, CHKBUTT2
             BSET   PIFH, BUTT3
             LDAB   ASCIINUM
             JSR    INCASCII
             STAB   ASCIINUM

CHKBUTT2:    BRCLR  PIFH, BUTT2, PSHBUTTNEND
             BSET   PIFH, BUTT2
             LDAB   ASCIINUM
             JSR    DECASCII
             STAB   ASCIINUM
PSHBUTTNEND: RTI
```

Example - Subroutines

```
;-------------------------------------------------------------
; Subroutines
;-------------------------------------------------------------

;-------------------------------------------------------------
; INCASCII
; Input:
;      B: ASCII number
; Output:
;      B: (B)+1 if (B) < '9'
INCASCII      CMPB   #'9'
              BHS    ENDINC
              INCB
ENDINC:       RTS

;-------------------------------------------------------------
; DECASCII
; Input:
;      B: ASCII number
; Output:
;      B: (B)-1 if (B) > '0'
DECASCII      CMPB   #'0'
              BLS    ENDDEC
              DECB
ENDDEC:       RTS
```

Periodic Interrupt Device

- Assuming that we display clock on 7-segment LEDs.
  - We have used loops to generate pauses.
  - However, executing these loops also consume the processor.
  - A periodic interrupt allows the processor to work on other tasks.

Real-time Interrupt

- The Real-time Interrupt (RTI) generates hardware interrupt periodically.
- The rate is selected by the RTICTL register.

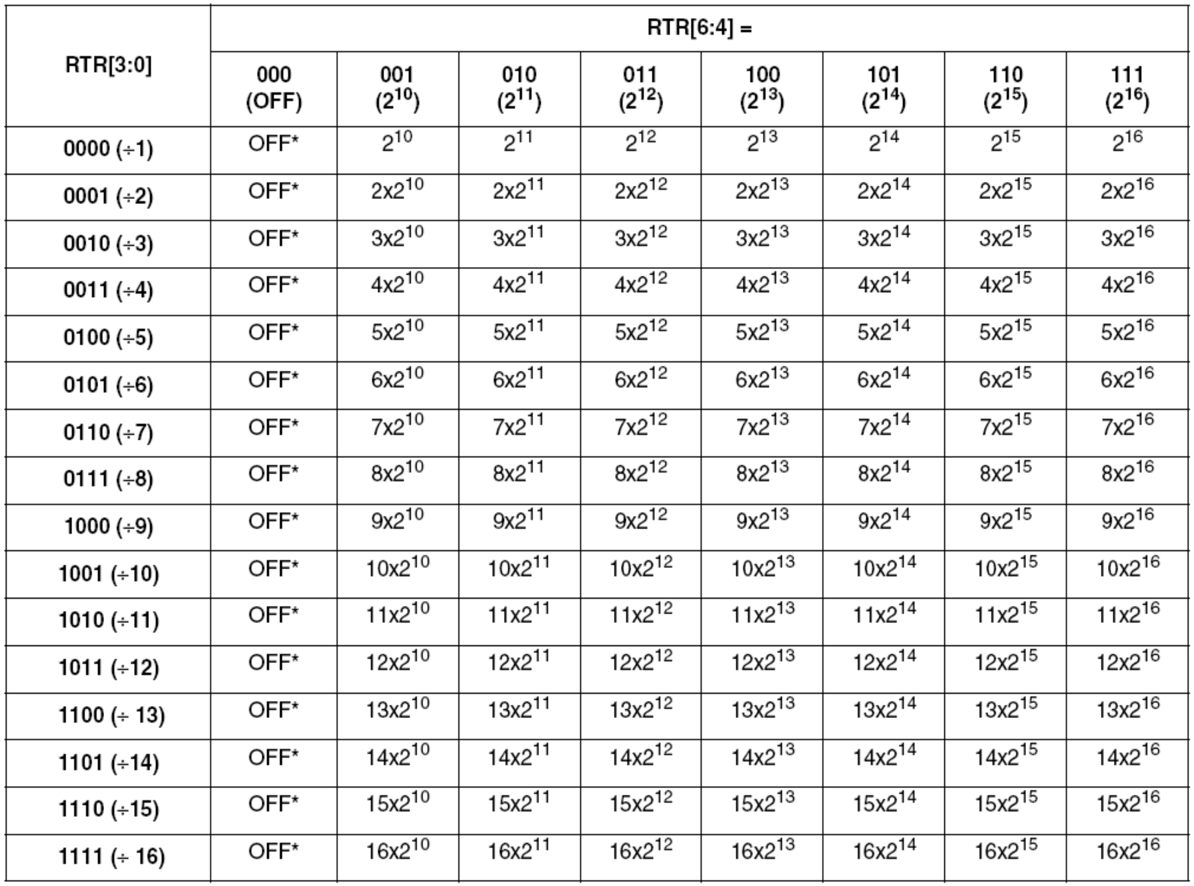
Real-time Interrupt (cont'd)

- The registers that are used to control the RTI interrupt.
    - CRGINT
    - CRGFLG

Real-time Interrupt (cont’d)

- Interrupt vector : $FFF0 ($3E70 in Dragon 12+)
- RTICTL register
  - Real Time Interrupt Control register: $003B
  - The value in RTICTL determines how often the periodic interrupt occurs in terms of external clock periods.
  - *See the table in the following slide*.
- Note that if the RTICTL register is not configured, the interrupt will not occur even if the RTIE bit is set

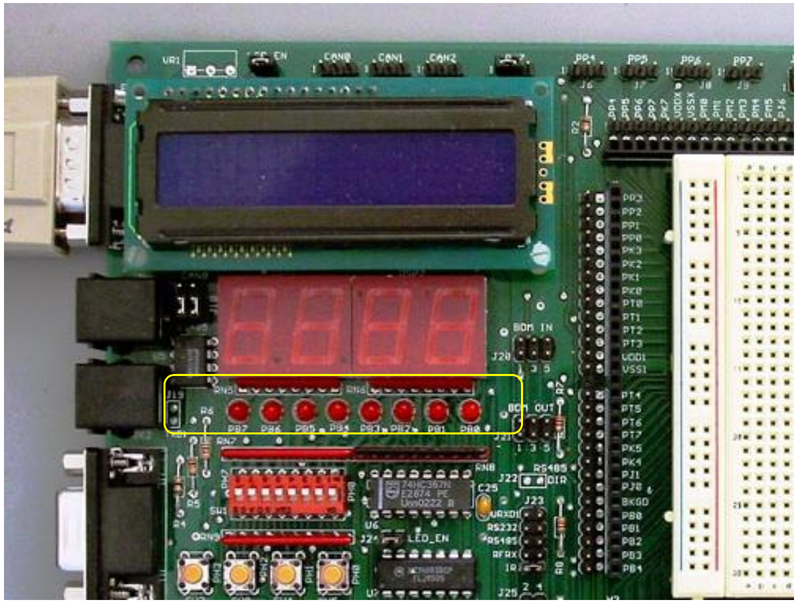
Real-time Interrupt (cont'd)



RTR[3:0]	RTR[6:4] =							
	000 (OFF)	001 (2^{10})	010 (2^{11})	011 (2^{12})	100 (2^{13})	101 (2^{14})	110 (2^{15})	111 (2^{16})
0000 (÷1)	OFF*	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
0001 (÷2)	OFF*	$2x2^{10}$	$2x2^{11}$	$2x2^{12}$	$2x2^{13}$	$2x2^{14}$	$2x2^{15}$	$2x2^{16}$
0010 (÷3)	OFF*	$3x2^{10}$	$3x2^{11}$	$3x2^{12}$	$3x2^{13}$	$3x2^{14}$	$3x2^{15}$	$3x2^{16}$
0011 (÷4)	OFF*	$4x2^{10}$	$4x2^{11}$	$4x2^{12}$	$4x2^{13}$	$4x2^{14}$	$4x2^{15}$	$4x2^{16}$
0100 (÷5)	OFF*	$5x2^{10}$	$5x2^{11}$	$5x2^{12}$	$5x2^{13}$	$5x2^{14}$	$5x2^{15}$	$5x2^{16}$
0101 (÷6)	OFF*	$6x2^{10}$	$6x2^{11}$	$6x2^{12}$	$6x2^{13}$	$6x2^{14}$	$6x2^{15}$	$6x2^{16}$
0110 (÷7)	OFF*	$7x2^{10}$	$7x2^{11}$	$7x2^{12}$	$7x2^{13}$	$7x2^{14}$	$7x2^{15}$	$7x2^{16}$
0111 (÷8)	OFF*	$8x2^{10}$	$8x2^{11}$	$8x2^{12}$	$8x2^{13}$	$8x2^{14}$	$8x2^{15}$	$8x2^{16}$
1000 (÷9)	OFF*	$9x2^{10}$	$9x2^{11}$	$9x2^{12}$	$9x2^{13}$	$9x2^{14}$	$9x2^{15}$	$9x2^{16}$
1001 (÷10)	OFF*	$10x2^{10}$	$10x2^{11}$	$10x2^{12}$	$10x2^{13}$	$10x2^{14}$	$10x2^{15}$	$10x2^{16}$
1010 (÷11)	OFF*	$11x2^{10}$	$11x2^{11}$	$11x2^{12}$	$11x2^{13}$	$11x2^{14}$	$11x2^{15}$	$11x2^{16}$
1011 (÷12)	OFF*	$12x2^{10}$	$12x2^{11}$	$12x2^{12}$	$12x2^{13}$	$12x2^{14}$	$12x2^{15}$	$12x2^{16}$
1100 (÷ 13)	OFF*	$13x2^{10}$	$13x2^{11}$	$13x2^{12}$	$13x2^{13}$	$13x2^{14}$	$13x2^{15}$	$13x2^{16}$
1101 (÷14)	OFF*	$14x2^{10}$	$14x2^{11}$	$14x2^{12}$	$14x2^{13}$	$14x2^{14}$	$14x2^{15}$	$14x2^{16}$
1110 (÷15)	OFF*	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
1111 (÷ 16)	OFF*	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

Example

- Write an interrupt driven program that increments the value in Port B by 1 every second and displays the current value to LEDs.
  - There are 8 simple red LEDs on the Dragon12+ board.
  - Note that when the Port J pin 1 is low, the LEDs are enabled to light.
  - The LEDs are showing the value in the Port B, if the LEDs are enabled.



Example – constants

A Part of "d12plus.inc"

```
; -------------------
; Real time interrupt
CRGFLG         equ  $0037
RTIF      equ  $80
CRGINT         equ  $0038
RTIE      equ  $80
; -------------------
; Real time interval setting
RTICTL         equ  $003B

;  number of interrupts per second
ONESEC         equ  122

;-------------------------------------------------------------
; Interrupt vector table
IVEC_PORTH     equ  $3E4C     ; Actual address is $FFCC
IVEC_RTI  equ  $3E70
...
```

```
#INCLUDE d12plus.inc

;=============================================================
; Set an ISR to the interrupt vector table
          ORG  IVEC_RTI
          DC.W ISR_RTI
```

Example – main

```
; Data section
          ORG  DATASTART
RTICOUNT  DS.W 1

; Main program
          ORG  PROGSTART
          LDS  #STACKSTART
          ;--------------------------------------------
          ; Init ports
          ; Prevent 7-segments from lighting
          MOVB #DOUT, DDRP
          MOVB #TRUE, PORTP   ; disable digits
          ; Init PortB to all outputs
          MOVB #DOUT, DDRB
          ; Enable the 8 single LEDs
          MOVB #%00000010, DDRJ ; configure PJ1 pin for output
          BCLR PORTJ, %00000010 ; enable the LEDs to light
          ; initialize the RTI
          MOVW #ONESEC, RTICOUNT
          MOVB #$70,RTICTL ; 1x2^16     ; $70 = 0100:0000
          BSET CRGINT, RTIE
          ; initialize the display to 0's
          CLR  EIGHTLED
          ;-------------------------------------------
          ; Enable interrupts
          CLI
          ;-------------------------------------------
          ; and wait for things to happen
LOOP:          BRA  LOOP
```

Example – ISR

```
;------------------------------------------------------------
; Interrupt service routine
;------------------------------------------------------------
ISR_RTI   BRCLR     CRGFLG, RTIF, RTIEND
          BSET CRGFLG, RTIF

          LDD  RTICOUNT
          SUBD #1
          STD  RTICOUNT
          BNE  RTIEND

          ; when RTICOUNT reaches to 0, reset RTICOUNT value
          MOVW #ONESEC, RTICOUNT
          INC  EIGHTLED
RTIEND:   RTI
```

Questions?

Wrap-up

What we’ve learned

- Interrupt devices
- Real time interrupt

What to Come