Lecture 21:
Interrupt Program Example

Today’s Goals

- Use edge-triggered interrupt and real time interrupt

Example

Interrupt-driven program

- Write an interrupt-driven program that implements the following requirements.

- This program will perform roughly the same function as the stopwatch feature on many wristwatches.

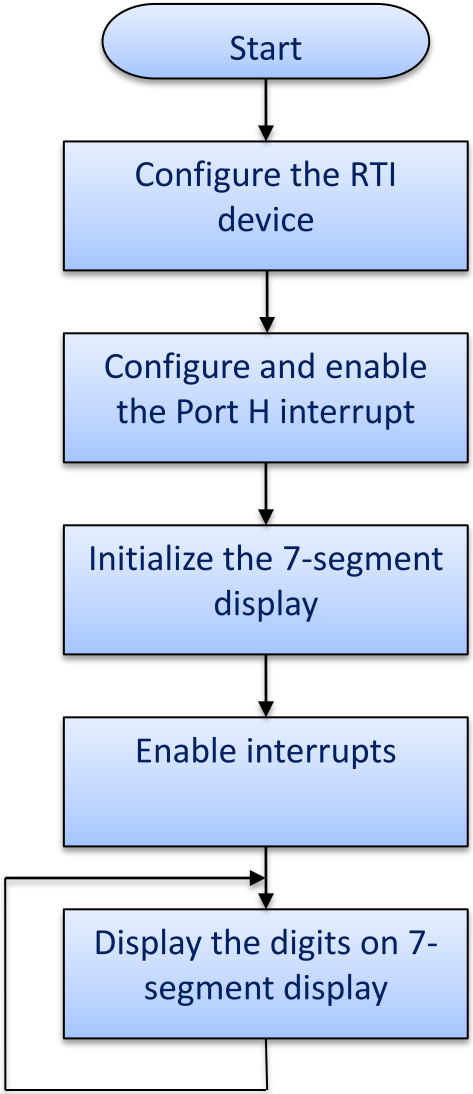
- Requirements
  - When the program begins, the 7-segment display should read “0000” and not incrementing.
  - When the pushbutton BUTTN0 is pressed, the display should increment by 1 at 1-second intervals.
  - When the pushbutton BUTTN0 is pressed a second time, the display should halt at the current value.
  - When the pushbutton BUTTN0 is pressed a third time, the display should reset to “0000” and be ready to start the timing process again

Example – cont’d

Notes

- The main program will be used as a loop to cycle through the 7-segment displays.
- The push button can be pressed at any time so it should be enabled initially.
- The RTI will only need to be enabled during the timing operation.
- There might be much better approaches…

Main Program Flowchart



Start

Configure the RTI device

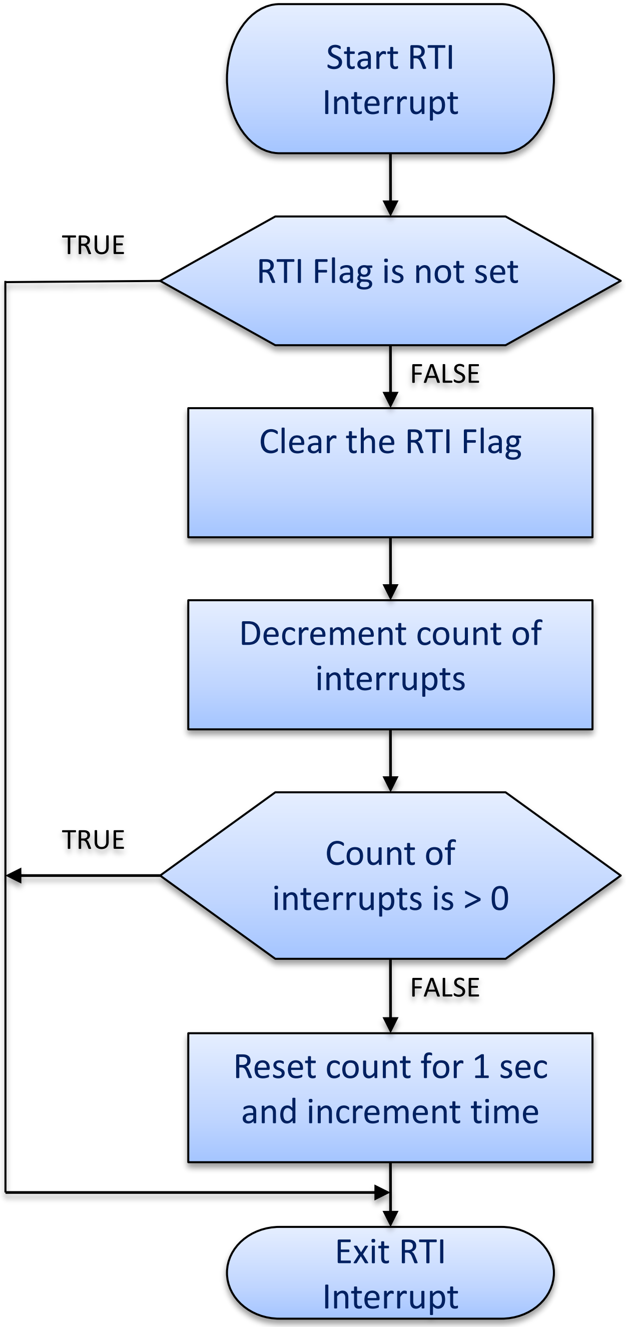
Configure and enable the Port H interrupt

Initialize the 7-segment display

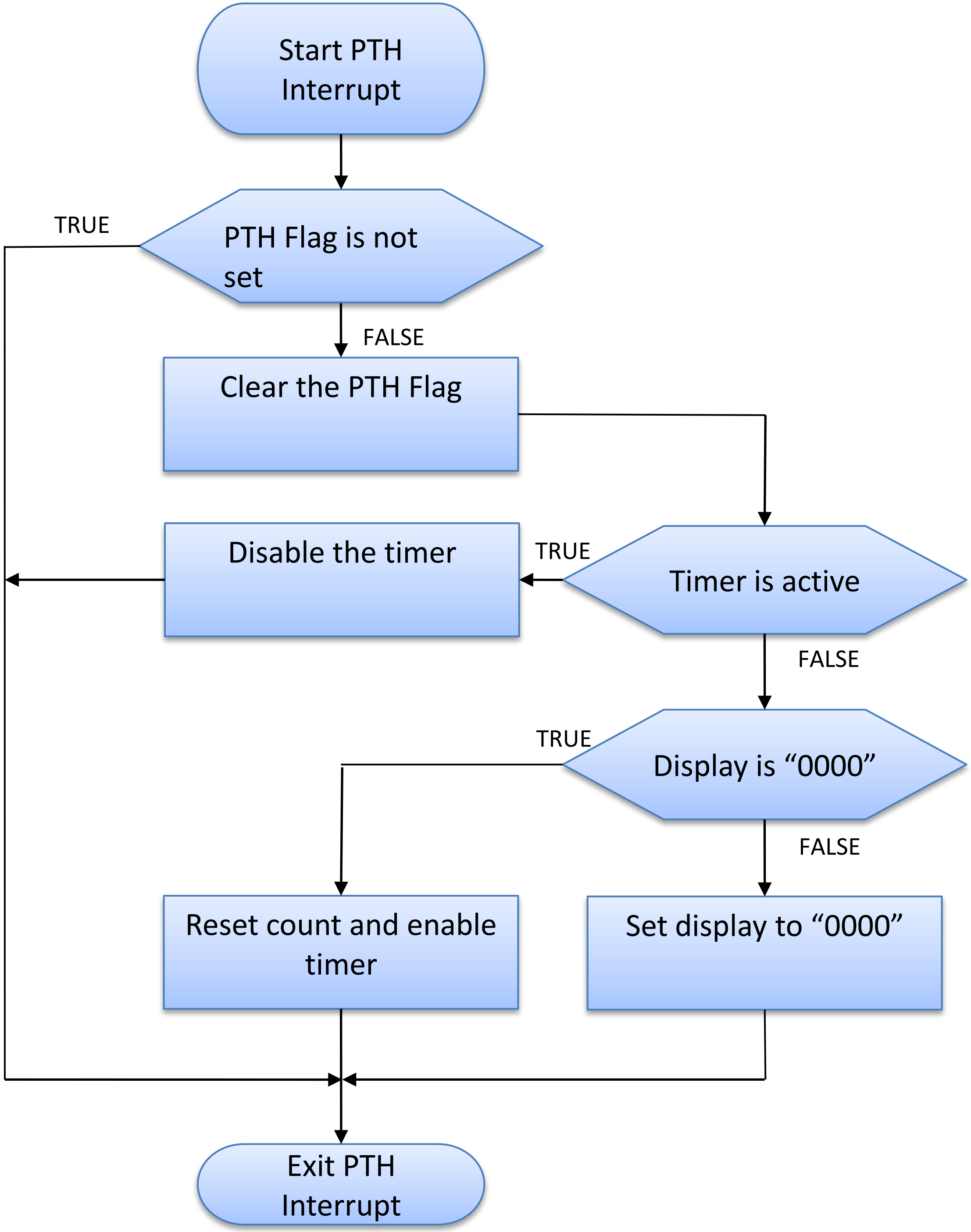
Enable interrupts

Display the digits on 7-segment display

RTI ISR Flowchart


Start RTI Interrupt
RTI Flag is not set
TRUE
FALSE
Clear the RTI Flag
Decrement count of interrupts
Count of interrupts is > 0
TRUE
FALSE
Reset count for 1 sec and increment time
Exit RTI Interrupt

Port H ISR Flowchart



Program Code – Constants

```
#INCLUDE d12plus.inc

;============================================================
; !! This program needs to get 'seven_seg.s19' loaded
;============================================================

;============================================================
; 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
             ORG IVEC_RTI
             DC.W ISR_RTI
```

Program Code – Variables and Macro

```
;============================================================
; Data section
;------------------------------------------------------------
             ORG DATASTART
             ; create storage for the patterns for each digit
ASCIINUM     DS.B 4
             ; real time interrupt counter
RTICOUNT     DS.W 1

;============================================================
; Program section
;------------------------------------------------------------
; Macro
;------------------------------------------------------------
CLEAR_ASCIINUM   MACRO
             ;---------------------------------------------
             ; initialize the display to 0's
             MOVB #'0', ASCIINUM
             MOVB #'0', ASCIINUM+1
             MOVB #'0', ASCIINUM+2
             MOVB #'0', ASCIINUM+3
             ENDM
```

Program Code - Main

```
;------------------------------------------------------------
; Main program
;------------------------------------------------------------
             ORG  PROGSTART
             LDS  #STACKSTART

             JSR  INIT7SEG
             ;---------------------------------------------
             ; configure/enable SW5 interrupt (portH)
             BCLR PORTH, BUTTN0
             BSET PIEH, BUTTN0
             ;---------------------------------------------
             ; configure RTI device
             MOVB #RTICTLVAL, RTICTL
             ;---------------------------------------------
             ; initialize the display to 0's
             CLEAR_ASCIINUM
             ;---------------------------------------------
             ; enable interrupts...
             CLI
```

Program Code – Main (continued)

```
             ;---------------------------------------------
             ; and wait for things to happen
LOOP:        LDAB ASCIINUM
             LDAA #DIGIT3
             JSR  DISP7SEG
             JSR  PAUSE
             LDAB ASCIINUM+1
             LDAA #DIGIT2
             JSR  DISP7SEG
             JSR  PAUSE
             LDAB ASCIINUM+2
             LDAA #DIGIT1
             JSR  DISP7SEG
             JSR  PAUSE
             LDAB ASCIINUM+3
             LDAA #DIGIT0
             JSR  DISP7SEG
             JSR  PAUSE
             BRA  LOOP
```

Program Code – Subroutines

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

;------------------------------------------------------------
; INCASCIIWRAP
; Input:
;   B: ASCII number
; Output:
;   B: (B)+1 if (B) < '9'
INCASCIIWRAP INCB
             CMPB#'9`
             BNE ENDINCWRAP
             LDAB#'0'
ENDINCWRAP:  RTS

;------------------------------------------------------------
; PAUSE
; Pause for about 0.1 ms
PAUSE        LDX #600
PSLOOP:      DEX
             BNE PSLOOP
             RTS
```

Program Code – Real Time ISR

```
ISR_RTI      BRCLR   CRGFLG, RTIF, RTIEND
             LDAA #RTIF
             STAA CRGFLG ; store 1 to reset the flag

             ; count interrupt for incrementing the display
             LDD  RTICOUNT
             SUBD #1
             STD  RTICOUNT
             BNE  RTIEND
             MOVW #ONESEC, RTICOUNT
             ; increment display as a 4-digit value
             LDAB ASCIINUM+3
             JSR  INCASCIIWRAP
             STAB ASCIINUM+3
             CMPB #'0'
             BNE  RTIEND

             LDAB ASCIINUM+2
             JSR  INCASCIIWRAP
             STAB ASCIINUM+2
             CMPB #'0'
             BNE  RTIEND
```

Program Code – Real Time ISR (continued)

```
                LDAB ASCIINUM+1
                JSR  INCASCIIWRAP
                STAB ASCIINUM+1
                CMPB #'0'
                BNE  RTIEND

                LDAB ASCIINUM
                JSR  INCASCIIWRAP
                STAB ASCIINUM
RTIEND:         RTI
```

Program Code – Real Time ISR (continued)

```
ISR_PSHBUTTN BRCLR    PIFH, BUTTN0, PSHBUTTNEND
             ; set to 1 to reset PIFH flag
             BSET PIFH, BUTTN0
             ; if RTI is already enabled, then
             ; go to disable (stop RTI)
             BRSET    CRGINT, RTIE, DISABLE

             LDD  ASCIINUM
             CPD  #$3030
             BNE  CLEARTIME
             LDD  ASCIINUM+2
             CPD  #$3030
             BNE  CLEARTIME

             ;start timer
             MOVW #ONESEC, RTICOUNT
             BSET CRGINT, RTIE
             BRA  PSHBUTTNEND
             ;clear timer
CLEARTIME:   CLEAR_ASCIINUM
             BRA  PSHBUTTNEND

             ;stop timer
DISABLE:     BCLR CRGINT,RTIE
PSHBUTTNEND: RTI
```

Questions?

Wrap-up

What we’ve learned

- Example of interrupt-driven program
  - Trigger-edged interrupt
  - Real time interrupt
- Study this example code
  - Help to do your lab program assignments

What to Come

- Now, we will discuss C programming for embedded systems.