141 Microcontroller Lab

**TIMER and IRQ Interrupts**

**BACKGROUND:**

An interrupt is the ability of a processor to temporarily stop a program, execute another program, then return to the original program exactly where it left off. Interrupts allow the processor to multi-task.

Interrupts can be generated from inside or outside the processor. The 68HC11 has over 20 different sources of interrupts. These include the SWI software interrupt and TOF (Timer OverFlow Interrupt) which are internal to the processor, and the IRQ (Interrupt ReQuest) which is an external interrupt. The IRQ is a digital input pin on the processor that permit an external device cause an interrupt in the processor.

**OBJECTIVES:**

Test a program that uses the TOF internal Interrupt.

Test a program that uses the IRQ external Interrupt.

**PROCEDURE:**

Program 1.

Sound the bell on the terminal every 2 seconds.

```plaintext
*Program to sound the Bell on the monitor once every second.
* Program checks for TCNT timer overflow which occurs every 16 ms
* Program checks by polling TFLG2 register
* Adapted from Software and Hardware Engineering (Cady)
**
* TCNT -is free running, can't control it.
* -it sets the TOF bit in TFLG2 when it rolls over
* -roll over is from ffff to 0000.
*
*CP Bleau
*4/27/03
*
*
*Subroutine Equates
outa   equ   $ffb8 ;Subroutine to output to terminal
```

Microcontroller 141 Page 1 of 3
* I/O Register equates
  tflag2 equ $1025 ;tflag2 register
tof equ %10000000 ;locate overflow flag in tflag2 register
*
* Data Equates
* Variables
  org $3000 ;Locate data area
  count rmb 1 ;Place to keep track of the 30 counts
*
* Constants
  bell equ $41 ;Ascii for Bell sound on monitor
  onesec equ 30 ;30 counts of 32.767 milliseconds
*
* Program starts Here
  ****************
  start org $2000
     ldaa #tof
     staa tflag2
 repeat ldaa #onesec
     staa count
 loop ldaa tflag2
     bita #tof
     beq loop
     ldaa #tof
     loop
 for next staa tflag2
     dec count
     bne loop
     ldaa #bell
     jsr outa
 bra repeat
  ****************

Program 2.

Output a message to the terminal “Waiting for IRQ” while waiting for the IRQ Interrupt. Output “IRQ Active” When the IRQ is received. After 2 seconds again output “Waiting for IRQ”. The EZ-MICRO Board must be in single chip mode for this program to operate. The IRQ line must be taken low for the Interrupt to activate.

***********************************************************
**** IRQ Interrupt **************************************
**** Mr. Lokken ***************************************
***********************************************************
irqjmp   equ   $00ef

dly10ms  equ   $e2ef
outstrg  equ   $ffc7

org      $0100

main     sei
ldx      #irqisr
stx      irqjmp
cli

start    ldx      #msgwai
jsr       outstrg
wai
bra       start

start    ldx      #msgwai
jsr       outstrg
wai
bra       start

tdly     ldab     #200
up       jsr       dly10ms
decb     bne       up
rts

irqisr   ldx      #msgirq
jsr       outstrg
jsr       tdly
rts

msgwai   fcc       'Waiting for IRQ'
fcb       $04

msgirq   fcc       'ISR IRQ is running'
fcb       $04