

Jump Instructions – Greater/Same and Lower

Lab Goal

Understand the effect and use of jump instructions.

Instructions/Questions:

- 1) Open Workspace and Project Lab 13. Connect the [FET kit](#) to the computer. Edit the asm.s43 file for this project.
- 2) The code is already set up to read an internal temperature sensor connected to an analog to digital converter and place [the result in R14](#).
For the sake of this lab, assume that [the value stored in R14 is always a negative](#) value and the value gets less negative as the temperature goes higher.
- 3) Write code in the “Student Code Section” that [turns off the LED](#) when the value in R14 [is greater than or equal](#) to the [stable temperature](#) value [plus 8](#) and [turns on the LED](#) when it is [less than this value](#).

Test your code using the debugger by [warming the MCU](#) on the development board with your breath or other heat source.

Paste your code below.

Solution :

1)- Download Attached Files [JGE JL.zip](#) (59.533 KB)

From : [Lab 12 \(JGE JL\)](#) in Blackboard .

Open file JGE JL type IAR IDE

Edit file asm.s43



2)- Since the code is set up to read the internal temperature sensor, we use the Debug -> Auto step menu. Enter 500 for the delay and leave the level at Step Into and Click Start.

Select R5 to get the stable temperature threshold value plus 8 (use the ADD #8, R5 instruction)

After performing the auto debug. We have the stable temperature threshold result as **R5 = 0xfee6**.

3) Write code in the “Student Code Section”

CODE :

```

; Load the stable temperature into R5
MOV # 0xfee6, R5          ; Load stable temperature into R5
ADD # 8, R5               ; Add 8 to get threshold
CMP R5, R14               ; Compare R14 (current temperature) with threshold in R5
JGE TurnOffLED            ; If R14 >= R4, jump to TurnOffLED
JL TurnOnLED              ; If R14 < R4, jump to TurnOnLED

TurnOffLED:
BIC.B #0x01, &P1OUT        ; Set P1.0 high to turn off LED
JMP continue              ; Continue with the code was set up

TurnOnLED:
BIS.B #0x01, &P1OUT        ; Clear P1.0 to turn on LED
JMP continue              ; Continue with the code was set up

```

Test your code using the debugger by warming the MCU on the development board with your breath or other heat source.

Use a thumb to place and hold it for a while to heat up on the MCU. We see the LED next to the MCU light up and flash, and the Code Test with the Debugger is complete.



- ✚ IAR EW screen for MSP 430 in project asm.s43 file,, with Code set up to read internal temperature sensor along with result after performing auto debug for stable temperature threshold $R5 = 0xfee6$. and code of “Student Code Section”

The screenshot displays the IAR Embedded Workbench IDE for the MSP430. The main window shows the assembly code for the 'asm.s43' file. The code is organized into sections: initialization (init), setup of the ADC (SetupADC10), and a main loop (main). The main loop reads the temperature sensor (R4) and compares it to a threshold (R5 = 0xfee6). If the temperature is above the threshold, the LED is turned off; otherwise, it is turned on. The registers window shows the current state of the CPU registers, with R5 containing the value 0xfee6. The disassembly window shows the machine code instructions corresponding to the assembly code.

```

4      PUBLIC main                ; make the main label visible
5
6      ORG 0F800h                 ; outside this module
7
8  init: MOV.W #0200h, SP          ; set up stack
9      MOV.W #WDTPW+WDTHOLD,&WDTCTL ; Stop watchdog timer
10     BIS.B #DIVS_3,&SCSCTL2      ; SMCLK/8
11     MOV.W #0000h, R14
12     MOV.W #0000h, R4
13
14  setupP1: BIS.B #001h, &P1DIR    ; P1.0 output
15
16  SetupADC10: MOV.W #INCH_10+ADC10DIV_3,&ADC10CTL1 ; Temp Sensor ADC10CLK/4
17             MOV.W #SREF_1+ADC10SHT_3+REFON+ADC10ON+ADC10IE,&ADC10CTL0 ;
18
19  main: BIS.W #GIE,SR             ; enable interrupts
20       NOP                       ; main program
21
22  ;-----
23  ; Student Code Section
24  ; Load the stable temperature into R4
25
26  MOV.W #0xfee6, R5              ; Add 8 to get threshold
27
28  ADD #8, R5                     ; Add 8 to get threshold
29
30  CMP R5, R14                    ; Compare R14 (current temp) with R4
31  JGE TurnOffLED                 ; If R14 >= R4, jump to TurnOffLED
32  JLT TurnOnLED                  ; If R14 < R4, jump to TurnOnLED
33
34  TurnOffLED:                    ; Set P1.0 high to turn off LED
35  BIC.B #0x01, &P1OUT
36  JMP continue                   ; Loop back to check temperature again
37
38  TurnOnLED:                     ; Clear P1.0 to turn on LED
39  BIS.B #0x01, &P1OUT
40  JMP continue                   ; Loop back to check temperature again
41
42  ;-----

```

The registers window shows the following values:

Name	Value	Access
PC	0xf82c	ReadWrite
SP	0x0280	ReadWrite
SR	0x0008	ReadWrite
Reserved	0x0	ReadWrite
V	0	ReadWrite
SCG1	0	ReadWrite
SCG0	0	ReadWrite
OscOff	0	ReadWrite
CPWOff	0	ReadWrite
GIE	1	ReadWrite
N	0	ReadWrite
Z	0	ReadWrite
C	0	ReadWrite
R4	0x0000	ReadWrite
R5	0xfee6	ReadWrite
R6	0x0089	ReadWrite
R7	0x00ff	ReadWrite
R8	0x369c	ReadWrite
R9	0x4dc3	ReadWrite
R10	0x0000	ReadWrite
R11	0xf195	ReadWrite
R12	0xf7f1	ReadWrite
R13	0xfefc	ReadWrite
R14	0x0000	ReadWrite
R15	0x0121	ReadWrite
CYCLECOUNTER	61	ReadOnly
CTIMER1	61	ReadWrite
CTIMER2	61	ReadWrite
CCSTEP	2	ReadOnly

The disassembly window shows the following instructions:

```

init:
00F900 4031 0280      mov.w #0x280,SP
00F904 40B2 5A80 0120  mov.w #0x5A80,&WDTCTL
00F90A D0F2 0006 0058  bis.b #0x6,&SCSCTL2
00F910 MOV.W #0000h,R14
00F912 430E          clr.w R14
00F914 MOV.W #0000h,R4
00F916 4304          clr.w R4
00F918 BIS.B #001h,&P1DIR
00F91A D3D2 0022      bis.b #0x1,&P1DIR
00F91C SetupADC10: mov.w #INCH_10+ADC10DIV_3,&ADC10CTL1
00F91E SetupADC10: mov.w #SREF_1+ADC10SHT_3+REFON+ADC10ON+ADC10IE,&ADC10CTL0
00F920 MOV.W #GIE,SR
00F922 NOP
00F924 D232          eint
00F926 4303          nop
00F928 MOV.W #0xfee6,R5
00F92A ADD #8,R5
00F92C 5235          add.w #0x8,R5
00F92E CMP.R5,R14
00F930 JGE TurnOffLED
00F932 JLT TurnOnLED
00F934 BIC.B #0x01,&P1OUT
00F936 JMP continue
00F938 TurnOffLED:
00F93A BIS.B #0x01,&P1OUT
00F93C JMP continue
00F93E TurnOnLED:
00F940 CLR.B #0x01,&P1OUT
00F942 JMP continue

```

