Quiz 5 Solution
ECE 2560
Spring 2015

Do exactly what you did in the Project with the following modifications: Modify the main program and the subroutine so that the input and output to the subroutine are handled on the stack and not through the core registers.

Start with the solution of the Project that I will post on our website (on Tuesday) and modify it to accommodate the problem statement of this quiz. This way all the students will start from a known correct solution of the Project posted on our web site. This will also make the grading of the quiz easier for the graders. Just take my solution of the Project, make a few modifications and you are done!
Pseudo Code

for(i = 0, i<16, ++i)
{
    If( ith bit of R10 is set)
    {
        reset ith bit of R10
    }
    else
    {
        set ith bit of R10
    }
}

R11 = R10 + 1
Quiz5 Sol

Flow Chart

```cpp
reserve m stack 0(sp) ← i
reserve on stack 2(sp) ← moving bit
6(sp) ← space for output
8(sp) ← input

2(sp) = 1

0(sp) = 0

for_con_label:

for_break

if : 

0(sp) ≥ 16

if_label:

bit.w 2(sp), 8(sp)

N

bis.w 2(sp), 9(sp)

if_break

if_label:

bic.w 2(sp), 8(sp)

if_break:

r.la.w 2(sp)

++0(sp)

for_con_label

for_break:

mov.w 2(sp), 6(sp) inc.w 6(sp)

...
mov.w $2(sp), $4(sp)  
inc.w $4(sp)  
add.w $8, $4, $5  
ret
Code

;--------------------------------------------------------------------------------
; MSP430 Assembler Code Template for use with TI Code Composer Studio
;
;--------------------------------------------------------------------------------

.cdecls C,LIST,"msp430.h" ; Include device header file

.data
Result: .space 2

.text ; Assemble into program memory
.retain ; Override ELF conditional linking
        ; and retain current section
.retainrefs ; Additionally retain any sections
        ; that have references to current
        ; section

; RESET mov.w #__STACK_END,SP ; Initialize stackpointer
StopWDT mov.w #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer

; Main loop here
; call subroutine TwosComp

    push.w #28 ; prepare input for TwosComp
    sub.w #2, SP ; space for output
    call #TwosComp ; call subroutine TwosComp
    pop.w &Result ; move result from TwosComp to
    variable Result
    add.w #2, SP ; reclaim space

Loop:    jmp  Loop

; Subroutine: TwosComp

; Input: on stack
; Output: on stack : 2's compliment of input
;
;--------------------------------------------------------------------------------

... Continued ...
--- Continued ...

TwosComp:

    sub.w #4, SP ; reserve space on the stack for 0(SP) and 2(SP)
    ; 0(SP) -> index for the for loop
    ; 2(SP) -> variable with the moving bit to be used with bit.w, bis.w and bic.w
    ; 4(SP) -> PC Value
    ; 6(SP) -> Space for output
    ; 8(SP) -> input

    mov.w #1, 2(SP)
    ;------- start for loop ------------------------
    mov.w #0, 0(SP)

    for_cond_label:
        cmp.w #16, 0(SP)
        jge    for_break
    ;------- start if-else ------------------------
    bit.w 2(SP), 8(SP)
    jc     if_label
        bis.w 2(SP), 8(SP)
    jmp    if_break
    if_label:
        bic.w 2(SP), 8(SP)
    if_break:
    ;------- end if-else ------------------------
    rla.w 2(SP)
    inc.w 0(SP)
    jmp    for_cond_label

    for_break:
    ;------- end for loop ------------------------
    mov.w 8(SP), 6(SP)
    inc.w 6(SP)

    add.w #4, SP ; release space on the stack

    ret

; Stack Pointer definition
;---------------------------------------------------------------
.global __STACK_END
.sect .stack

; Interrupt Vectors
;---------------------------------------------------------------
.sect ".reset" ; MSP430 RESET Vector
.short RESET