CSCI 255 – Intro to Embedded Systems
Homework #9
Fall 2013

Work must be done individually and show work!!

Due: **11/6/2013** by **beginning of class**

Fill in the tables with the correct register & carry-bit values in accordance with the code that gets interrupted. All interrupts occur in the middle of indicated instructions. Evaluate each register after the execution of the indicated instruction of the columns. All values must be in HEX format. **Fill-in all blocks**.

1)

```
MAIN CODE                                        ISR
--------------------------------          ------------------------------------

; assume initializations have been made       Random_ISR:       mov.b   #0x41, R5
mov.b    #0xE3, R4                                             add.b   R4, R5
bis.b    R4, R5                                                reti
sub.b    R5, R10 <----  interrupt occurs here
mov.b    R10, R6
and.b    R4, R5

...
```

Location	**Initial Values**	After **bis.b R4,R5**	After **sub.b R5, R10**	After **add.b R4, R5**	After **and.b R4, R5**
SP	0x0300				
R4	0x0021				
R5	0x0055				
R6	0x003F				
R10	0x0087				

2)

```
MAIN CODE                                        ISR
--------------------------------          ------------------------------------

; assume initializations have been made       Random_ISR:       mov.b  #0x41, R5
mov.b    #0x03, R7                                             swpb   R5
inc.b    R7                                                    reti
rla.b    R7  <----  interrupt occurs here
mov.b    R7, R6
xor.b    R6, R5 <----  interrupt occurs here

...
```

Location	**Initial Values**	After **rla.b R7**	After **swpb R5 (1st time)**	After **xor.b R4, R5**	After **swpb R5 (2nd time)**
SP	0x05F3				
R5	0x0053				
R6	0x00AB				
R7	0x0012				

3)

```
MAIN CODE                                   ISR
--------------------------------            ------------------------------------

; assume initializations have been made     Random_ISR:       mov.b  #0xF3, R6
mov.b    #0x03, R10                                          xor.b  R5, R4
and.b    R7, R10                                             add.b  R4, R6
rla.b    R10 <----interrupt occurs here                      reti
call     #One
dec.b    R6                                       Subroutine
...                                         ------------------------------------
                                            One:              mov.b #0xFF,P1OUT
                                                              push.b #0xAA
                                                              setc
                                  interrupt occurs here ----> sub.b  R10, R5
                                                              pop.b  R5
                                                              ret
```

Location	**Initial Values**	After **and.b R7,R10**	After **rla.b R10**	After **xor.b R5, R6**	After **add.b R4,R6**	After **pop.b R5**	After **dec.b R6**
SP	0x05F3						
R4	0x0043						
R5	0x0053						
R6	0x00AB						
R7	0x0012						
R10	0x00BD						

4)

```
MAIN CODE                                        ISR
--------------------------------                 ------------------------------------

; assume initializations have been made          Random_ISR:       mov.b  #0x01, R7
clrc                                                               bis.b  R5, R4
call     #Two                                                      inc.b  R4
and.b    R4, R5                                                    add.b  R4, R7
addc.b   R5, R10 <----interrupt occurs here                        reti
call     #Two <---- interrupt occurs here
swpb     R6                                           Subroutine
...                                              ------------------------------------
-
                                                 Two:              push.b #0x30
                                                                   setc
                                                                   rla.b  R5
                                                                   pop.b  R10
                                                                   ret
```

Location	**Initial Values**	After **rla.b R5 (1st time)**	After **and.b R4,R5**	After **addc.b R5, R10**	After **add.b R4,R7**	After **rla.b R5 (2nd time)**	After **swpb R6**
SP	0x05F3						
R4	0x0043						
R5	0x0053						
R6	0x00AB						
R7	0x0012						
R10	0x00BD						