EE 215 Homework #3 Solution

Instructions: Write your name, class number (ZM1801,02,03, or 04), student number, HW3, and date on top of page. **Box all answers**. Due at beginning of class. **Staple upper left** if you have more than one page. Show all work to get credit.

1. Condition codes (20 pts total, 5 pts each)
   Describe the following condition code (CC) flags in 1-2 sentences. [Book section 5.1.3 or User's Guide 6.3.3]
   (a) C
   (b) Z
   (c) N
   (d) V

Solution:

(a) C

A carry flag is set when the number is too big to be represented in the size of operation (byte or word). The carry flag also takes part in rotation and shifts.

(b) Z

The zero flag is set when the result of the operation is zero.
Used to check when two values are equal.

(c) N

Negative flag is equal to the msb of the result which indicates a negative number if the values are signed.

(d) V

Overflow flag is set when the result of a signed operation has overflowed. This happens when you add 2 positive numbers and get a negative number (operation is invalid!), or add 2 negative numbers and get a positive number.

2. Addressing modes (30 pts total, 10 pts each)
There are many ways to get the address for the operands in an instruction. Explain the difference between these pairs of instructions by showing the contents of R5 after the commands:

(a) mov.w R4, R5
mov.w @R4, R5

(b) mov.w X, R5

mov.w #X, R5

(c) mov.w 4(R6),R5

mov.w Start(R7), R5

Label	value	Address of the label value
X	24FC	2400
Start	24FE	2402

Registers	
R4	0x2504
R6	0x2500
R7	6

Memory	
address	data
0x2408	34
0x2409	12
⋮	⋮
0x24FC	89
0x24FD	3A
0x24FE	C7
0x24FF	CB
0x2500	79
0x2501	46
0x2502	11
0x2503	E5
0x2504	92
0x2505	30

Solution:

(a) mov.w R4, R5

Register mode.

Copy the contents of R4 to R5.

R5 = 2504

mov.w @R4, R5

Indirect register mode.

@R4 uses the R4 contents as an address. The word at memory address 0x2504 is copied to R5.

R5 = 3092

(b) mov.w X, R5

X is a label with value 24FC, stored at memory address 0x2400.

The label X value is copied to R5.

R5 = 24FC

mov.w #X, R5

The memory address where label X value is stored, is copied to R5.

R5 = 2400

(c) mov.w 4(R6),R5

Indexed mode.

The sum of the two values determines the address of the source operand.

(contents of R6, which is 2500) + 4 = the address (2500+4=2504) of data copied to R5.

R5 = 3092

mov.w Start(R7), R5

Indexed mode.

(contents of R7, which is 6)

+ (label Start's address, where label Start value is stored, which is 2402)

= the address (6+2402=2408) of data copied to R5.

R5 =1234

3. Addressing modes (20 pts total, 10 pts each)
   (a) Comment each line of code AND describe the overall effect of the program.
   (b) Draw the memory map diagram illustrating the contents of all memory affected by the following block of MSP430 assembly code before and after execution, respectively. (Specify the address by byte-size or word-size)

Example of a word-size memory map:

Address	Data
0x3000	AB45
0x3002	3409
0x3004	0001

Example of a byte-size memory map:

Address	Data
0x3000	45
0x3001	AB
0x3002	09
0x3003	34
0x3004	01
0x3005	00

```
      .data
Array:  .word    0x1011,0x2022,0x3033
Mem:    .space   6
Sum:    .space   2

      .text
   mov  #Array, R4
   mov  R4, R5
   mov #Mem, R6
   mov.b  @R4+, 5(R6)
   mov.b  @R4+, 4(R6)
   mov.b  @R4+, 3(R6)
   mov.b  @R4+, 2(R6)
   mov.b  @R4+, 1(R6)
   mov.b  @R4,  Mem

   clr  R7
   add  Array,   R7
   add  2(R5),   R7
   add  4(R5),   R7
   mov  R7,      Sum
```

Solution:

(a) Comment each line of code AND describe the overall effect of the program.

```
      .data         ; data section
                    ; the lines that follow will be assembled into the .data section.
Array: .word  0x1011,0x2022,0x3033 ; define word size data 0x1011,0x2022, and 0x3033.
          ; Array = 0x1011 as word size
          ; Array = 0x11 as byte size
          ; #Array = 0x2400, which is the address of the first byte labeled by Array
          ; #Array points to the first byte of the data defined in this line.
Mem:   .space 6       ; reserve 6 bytes in the current section and fill them with 0
                      ; label Mem points to the first byte reserved, which is 0x2406
Sum:   .space 2       ; reserve 2 bytes in the current section and fill them with 0s.
                      ;label Sum points to the first byte reserved, which is 0x2408

;-------------------------------------------------------------------------------
;     text section here
;     Begin assembling into the .text section.
;-------------------------------------------------------------------------------
      .text  ; The .text sets .text as the current section.
      ; Lines that follow this directive will be assembled into the .text section,
      ; which usually contains executable code.
      ; The section program counter is set to 0 if nothing has yet been assembled
      ; into the .text section.

;-------------------------------------------------------------------------------
;     This section copies the data at address 0x2400-0x2405 to address 0x2406-0x240B
;     in byte reverse order
;-------------------------------------------------------------------------------
   mov  #Array, R4  ; put the initial byte address of Array to R4. Now R4=2400
   mov  R4, R5      ; copy contents of R4 to R5. Now R5=2400
   mov #Mem, R6     ; put the first reversed byte address of Mem to R6. Now R6=2406
   mov.b  @R4+, 5(R6) ; the byte whose address is in R4, is copied to address R6+5,
                      ; then R4=R4+1
            ; i.e., copy the byte 0x11 at address 2400 to address 240B. Now R4=2401)
   mov.b  @R4+, 4(R6) ; the byte whose address is in R4, is copied to address R6+4,
                      ; then R4=R4+1
            ; i.e., copy the byte 0x10 at address 2401 to address 240A. Now R4=2402
   mov.b  @R4+, 3(R6) ; the byte whose address is in R4, is copied to address R6+3,
                      ; then R4=R4+1
            ; i.e., copy the byte 0x22 at address 2402 to address 2409. Now R4=2403
   mov.b  @R4+, 2(R6) ; the byte whose address is in R4, is copied to address R6+2,
                      ; then R4=R4+1
            ; i.e., copy the byte 0x20 at address 2403 to address 2408. Now R4=2404
   mov.b  @R4+, 1(R6) ; the byte whose address is in R4, is copied to address R6+1,
                      ; then R4=R4+1
            ; copy the byte 0x33 at address 2404 to address 2407. Now R4=2405
   mov.b  @R4,  Mem
      ; the byte whose address is in R4, is copied to reserved space labeled by Mem
      ; copy the byte 0x30 at address 2405 to address 2406.
```

```
;-------------------------------------------------------------------------------
;      This section add the data at address 0x2400-0x2404 as word size
;      and store the result in reserved space labeled by Sum
;-------------------------------------------------------------------------------
       clr   R7           ; clear R7
       add   Array,  R7   ; add the first word in Array to R7. R7=1011
       add   2(R5),  R7   ; add the second word in Array to R7. R7 = 1011+2022 = 3033
       add   4(R5),  R7   ; add the third word in Array to R7. R7 = 3033+3033 = 6066
       mov   R7, Sum      ; copy the data in R7 to memory space pointed by Sum
```

(b) Draw the memory map diagram illustrating the contents of all memory affected by the following block of MSP430 assembly code before and after execution, respectively.

(1) Memory map diagram by word-size address

before execution:

	Memory	
	address	data
Array	0x2400	1011
	0x2402	2022
	0x2404	3033
Mem	0x2406	0000
	0x2408	0000
	0x240A	0000
Sum	0x240C	0000

after execution:

	Memory	
	address	data
Array	0x2400	1011
	0x2402	2022
	0x2404	3033
Mem	0x2406	3330
	0x2408	2220
	0x240A	1110
Sum	0x240C	6066

(2) Memory map diagram by byte-size address

before execution:

	Memory	
	address	data
Array	0x2400	11
	0x2401	10
	0x2402	22
	0x2403	20
	0x2404	33
	0x2405	30
Mem	0x2406	00
	0x2407	00
	0x2408	00
	0x2409	00
	0x240A	00
	0x240B	00
Sum	0x240C	00
	0x240D	00

after execution:

	Memory	
	address	data
Array	0x2400	11
	0x2401	10
	0x2402	22
	0x2403	20
	0x2404	33
	0x2405	30
Mem	0x2406	30
	0x2407	33
	0x2408	20
	0x2409	22
	0x240A	10
	0x240B	11
Sum	0x240C	66
	0x240D	60