EE 215 Homework #3

(3 problems, 80 pts total)

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

Answers:

(a) C (Carry): The C flag is set when an arithmetic operation results in a carry-out from the most significant bit or a borrow into it, indicating an overflow in unsigned arithmetic.

(b) Z (Zero): The Z flag is set when the result of an operation is zero, indicating that the data involved in the operation is equal.

(c) N (Negative): The N flag is set when the result of an operation is negative, typically in signed arithmetic, if the most significant bit (sign bit) is set.

(d) V (Overflow): The V flag is set when an arithmetic operation results in an overflow, meaning the result is too large to be represented in the number of bits available, often seen in signed arithmetic.

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

Answers:

(a) mov.w R4, R5

registers mode

Copy the values stored in R4 to R5

R5=2504

mov.w @R4, R5

Indirect register mode.

@R4 means to take the value of R4 as the address and transfer the value of 0x2504 memory into R5

R5=3092

(b)

mov.w X, R5

Transfer the value of X to R5

R5=24FC

mov.w #X, R5

Transfer the address of X to R5

R5=2400

(c)

mov.w 4(R6),R5

Indexed mode.

Add the value of R6 2500 plus 4 as the address, and transfer the value from the memory address to R5

R5=3092

mov.w Start(R7), R5

Indexed mode.

Add the address of Start with the value of R7 to get a new address of 2402+6=2408, and transfer the value corresponding to 2408 in memory to R5

R5=1234

3. Addressing modes (30 pts total, 15 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

Answers:

(a)

```
      .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
```

```
   .data  ; We start with the data section.
Array: .word 0x1011, 0x2022, 0x3033 ; An array with three values.
Mem: .space 6 ; Reserve 6 bytes for Mem.
Sum: .space 2 ; Reserve 2 bytes for Sum.

;----------------------------------------------------------------------
----------
; Now, let's move to the text section where the action happens.
;----------------------------------------------------------------------
----------
    .text  ; We're now in the text section.

;----------------------------------------------------------------------
----------
; This section copies data from Array to Mem in reverse order.
;----------------------------------------------------------------------
----------
mov  #Array, R4  ; R4 now holds the starting address of Array (R4 =
0x2400).
mov  R4, R5      ; Copy the address to R5 (R5 = 0x2400).
mov #Mem, R6     ; Set R6 to point to the first byte of Mem (R6 =
0x2406).

; We're copying bytes from Array to Mem in reverse order.
mov.b  @R4+, 5(R6) ; Copy a byte from Array to Mem at R6+5, R4
increments (R4 = 0x2401).
```

```
mov.b  @R4+, 4(R6) ; Copy a byte from Array to Mem at R6+4, R4
increments (R4 = 0x2402).
mov.b  @R4+, 3(R6) ; Copy a byte from Array to Mem at R6+3, R4
increments (R4 = 0x2403).
mov.b  @R4+, 2(R6) ; Copy a byte from Array to Mem at R6+2, R4
increments (R4 = 0x2404).
mov.b  @R4+, 1(R6) ; Copy a byte from Array to Mem at R6+1, R4
increments (R4 = 0x2405).
mov.b  @R4, Mem    ; Copy the last byte from Array to Mem (R4 remains
at 0x2405).

;---------------------------------------------------------------------
----------
; In this section, we add data from Array and store the result in Sum.
;---------------------------------------------------------------------
----------
clr  R7            ; Clear R7 (R7 = 0).
add  Array, R7     ; Add the first word in Array to R7 (R7 = 0x1011).
add  2(R5), R7     ; Add the second word in Array to R7 (R7 = 0x1011 +
0x2022 = 0x3033).
add  4(R5), R7     ; Add the third word in Array to R7 (R7 = 0x3033 +
0x3033 = 0x6066).
mov  R7, Sum       ; Store the result in Sum.
```

(b)

(1) by word-size address

	Before		After	
	Memory		Memory	
	Address	Data	Address	Data
Array	0x2400	1011	0x2400	1011
	0x2402	2022	0x2402	2022
	0x2404	3033	0x2404	3033
Mem	0x2406	0000	0x2406	3330

	0x2408	0000	0x2408	2220
	0x240A	0000	0x240A	1110
Sum	0x240C	0000	0x240C	6066

(2) by byte-size address

	Before		After	
	Memory		Memory	
	Address	Data	Address	Data
Array	0x2400	11	0x2400	11
	0x2401	10	0x2401	10
	0x2402	22	0x2402	22
	0x2403	20	0x2403	20
	0x2404	33	0x2404	33
	0x2405	30	0x2405	30
Mem	0x2406	00	0x2406	30
	0x2407	00	0x2407	33
	0x2408	00	0x2408	20
	0x2409	00	0x2409	22
	0x240A	00	0x240A	10
	0x240B	00	0x240B	11
Sum	0x240C	00	0x240C	66
	0x240D	00	0x240D	60