EE 308 – Homework 2

1. Consider Program 1 from Lab 2:

```
prog:   equ     $2000           ; Starting address from program
data:   equ      $1000          ; Starting address for data

        org     prog            ; Set initial program counter value
        ldx     #1234           ;
        ldab    #235
        abx                     ;
        stx     result          ;
        swi

        org     data            ; Put data starting at this location
resut:  ds.w    1               ;
```

(a) Hand-assemble the program. That is, figure out what the op codes of the instructions Are and where they will be located in memory, the addressing mode, number of cycles, and the status of the NVCZ bits.

Address	Obj. Code Program Mode Cycles N V C Z
Address	**Obj. Code Data Mode Cycles N V C Z**

(b) How many cycles will it take the MC9S12 to execute this program. (Do not include the swi instruction.)

(c) How long will it take an MC9S12 with a 24 MHz E clock to execute this program?

(d) Determine the state of the N, Z, V and C bits after each instruction has been executed. (Assume that, when the program starts, all these bits are zero.)

(e) What will be the contents of addresses $1000 and $1001 after the program executes?

2. Consider Program 2 from Lab 2:

```
; MC9S12 program to copy a table of data from one location to another
; The copied data is the negative of the original data

prog:   equu    $2000       ; Starting address from program
data:   equ     $1000       ; Starting address for data
count:  equ     8           ; 8 elements in the table

        org     prog        ; Set initial program counter value
        ldab    #count      ; ACCB keeps count of number to transfer
        ldx     #table_1    ; X points at table_1
        ldy     #table_2    ; Y points at table_2
repeat: ldaa    1,X+        ; get data from table_1, X points to next element
        nega
        staa    1,Y+        ; save into table_2, Y points to next element
        decb                ; Decrement counter
        bne     repeat      ; If not done, continue with next element
        swi

        org     data        ; Put data starting at this location

; Initialize data in table

table_1: dc.b   $44,$61,$74,$61,$20,$54,$61,$62
table_2: ds.b   count       ; Reserve count bytes of memory for results
```

(a) Hand-assemble the program. That is, figure out what the op codes of the instructions Are, where they will be located in memory, and the addressing mode.

Address	Obj. code	Program	Mode
Address	**Obj. code**	**Data**	**Mode**

3. Write an instruction sequence to set the upper four bits of the number at address $0050 to 0, and leave the lower four bits unchanged.

4. Consider the following program fragment:

```
        ldy     #5000
loop1:  ldx     #5000
loop2:  dbne    x,loop2
        dbne    y,loop1
        swi
```

(a) Hand assemble the program. (Add an org assembler directive to put the program in memory starting at address 0x2000.)

Address	Obj. code	Program	#cycles

(b) How many instruction cycles will it take the MC9S12 to execute the program? (Do not consider the swi instruction.)

(c) How many seconds will this take the MC9S12 with an 24 Mhz E-clock? (You should give the answer to the nearest millisecond.)

5. An MC9S12 has the following data in its memory:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
10D0	10	A4	BF	43	31	F1	05	91	DD	AB	32	78	43	43	65	22
10E0	10	D8	01	45	87	54	C2	67	09	D2	A1	53	CF	E9	41	00
10F0	78	92	88	4F	3C	22	23	65	9C	4B	C3	B4	12	CB	98	65

Determine the contents of the B and Y register after executing the following code fragments. (Before the first instruction, the Y register has $0000.) List the values in hexadecimal. Also, indicate what addressing mode is used, and what the effective address of the instruction is. (Assume that the first instruction is at address $2000, and that the instructions that follow are in subsequent locations - i.e., the instruction of (a) takes two bytes, so the first instruction of (b) is at address $2002.)

	Mode	Instruction Addr.	Val. of Y and B	Eff. Addr.
a) Ldab #21				
b) ldy $10E7				
c) ldy $10E0 ldab -2,Y				
d) ldy #$10E0 ldab -2,Y				
e) ldy #$10E0 ldab 2,+Y				
f) ldy #$10E0 ldab 2,Y+				