EE 308

Exam 1

February 20, 2006

Name: ____

You may use any of the Motorola data books. No calculators allowed. Show all work. Partial credit will be given. No credit will be given if an answer appears with no supporting work.

1. Fill in the blanks in this table. The numbers are stored in an 8-bit register.

Hex	Binary	Unsigned Decimal	Signed Decimal
B5	10110101	181	-75
B7	10110111	183	-73
52	01010010	82	82
96	10010110	150	-106

2. The following operations are done in accumulator A of an 9S12. Indicate the answer in accumulator A, and the state of the flags after the operations.

	4D + 7E	5A + A6	B9 + 3A	37 - 9B	43 - 7F
Acc. A	CB	00	F3	9C	C4
C	0	1	0	1	1
V	1	0	0	1	0
N	1	0	1	1	1
Z	0	1	1	0	0

3. Below are some data in the 9S12 memory:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1000	F6	20	07	18	16	5C	4B	CF	28	00	B7	D4	2C	E2	6C	2B
1010	3F	F7	34	C6	C8	CD	9C	40	03	26	FD	53	26	F7	3D	3F
1020	07	C2	3A	68	F3	09	C2	67	9A	0F	AA	55	08	40	CD	CF

Reverse assemble the first eight instructions, starting at address 0x1000. Write down the mnemonic and operand for the instructions. Indicate the addressing mode used. Also indicate the effective address – that is, the address in memory which the instruction will use to fetch or store the number it is working on. Assume the registers initially have the following values:

```
A = $55, B = $AA, X = $1234, Y = $5678, SP = $3B13, PC = $1000
```

I have done one instruction for you; you need to do the next seven.

Instruction Address	Mnemonic	Operand	Addressing Mode	Effective Address
0x1000	`ldab`	`$2007`	EXT	`$2007`
0x1003	sba	—	INH	None
0x1005	std	$4b	DIR	$004B
0x1007	lds	`#$2800`	IMM	$1008
0x100A	exg	x,d	INH	None
0x100C	bge	-30 bytes	REL	$0FF0
0x100E	std	5,-x	IDX	(X) - 5
0x1010	swi	—	INH	None

Note: The BGE instruction will branch -30 bytes from address $100e, so it will end up at address $0ff0

Note: The STD instruction will store its data at the address of the X register minus 5. The first instruction loads A with the contents of address $2007 (which we cannot know from the data given), and the fifth instruction puts what was in D (A:B) into X, so we cannot know what is in the X register at the time of execution of the STD instruction.

4. Using the same data in the 9S12's memory as in Problem 3, indicate the values in the registers after the 9S12 executes the following instructions. Also write down the number of cycles needed to execute each instruction. Show what will be in the registers (in hex) after each of the instructions. If the instruction does not change a register, you may leave that entry blank.

Instruction	D: A	D: B	X	Y	SP	N	Z	V	C	Addressing Mode	Cycles
	AA	25	ABCD	1234	1020	1	0	0	1		
`ldx #$1008`			$1008			0				IMM	2
`ldy $1008`				$2800						EXT	3
`ldaa 5,-x`	$18		$1003							IDX	3
`sba`	$F3					1				INH	2
`pulx`			$07C2		$1002					INH	3
`negb`		$DB								INH	1

5. Using the same memory contents as for Problem 3, show what will be in the registers after you execute the following instructions. (The first instruction in the sequence is located at memory address `$2000`.)

Instruction	D: A	D: B	X	Y	SP	PC
	AA	25	ABCD	1234	1020	2000
PSHX					\$1020	\$2000
PULA	\$AB				\$101F	\$2001
PULB		\$CD			\$1020	\$2002
PULX			\$07C2		\$1022	\$2003
RTS					\$1024	\$3A68

6. An 9S12 on the Enterprise has been hooked up such that the 8-bit signed number stored at address `$0072` represents the threat level of a Romulan attack. If the value of this number is zero or negative, intelligence indicates that the Romulans are in a peaceful mood. If the value is positive, intelligence indicates that the Romulans are preparing to attack. If the number is 100 or greater, an attack in imminent.

   A switch which controls power to the shields is connected to Bit 0 of `PORTA`; a switch which controls power to the photon torpedoes is connected to Bit 4 of `PORTA`. (Writing a 0 to Bit 0 turns the shields off; writing a 1 to Bit 0 activates the shields. Writing a 0 to Bit 4 turns the photon torpedoes off; writing a 1 to Bit 4 turns on power to the photon torpedoes.) Write a program for the 9S12 which will do the following:

   - Set up Bits 0 and 4 of `PORTA` as output bits. The other bits of `PORTA` should be set up as inputs.
   - If the threat level zero or negative, turn off power to the shields and to the photon torpedoes.
   - If the threat level is positive, activate the shields.
   - If the threat level is 100 or greater, turn on power to the photon torpedoes.
   - Repeat this set of instructions forever.

```
PORTA:   equ   $0000
DDRA:    equ   $0002
threat:  equ   $0072

prog:    equ   $1000

         org   prog

         movb  #$11,DDRA     ; Make bits 4 and 0 of PORTA outputs
loop:    ldaa  threat        ; Load threat -- sets N and Z bits of CCR
         ble   safe          ; No threat if $0072 less than or equal to zero
         cmpa  #100
         blt   shields_only  ; If threat < 100, turn on shields only
both:    bset  PORTA,$11     ; If we got here, threat >= 100
                             ; turn on shields and power to torpedoes
         bra   loop
safe:    bclr  PORTA,$11     ; turn off shields and power to torpedoes
         bra   loop
shields: bset  PORTA,$01     ; turn on shields
         bclr  PORTA,$10     ; turn off power to torpedoes
         bra   loop
```