EE 308 – Homework 2

Due Feb. 1, 2009

1. Consider the following 8-bit hexadecimal numbers as unsigned. Find their decimal equivalents:

   (a) 0xA2

   (b) 0x6C

   (c) 0xF9

   (d) 0x2C

2. Repeat Problem 1, considering the numbers as signed.

3. Do the operations indicated below. The operations are performed in an 8-bit accumulator. Find the 8-bit results for the operations. Indicate the state of the N, Z, C and V bits after each operation.

   (a) 0x4D + 0x6F

   (b) 0xD2 + 0x9C

   (c) 0xD2 + 0x6A

   (d) 0xAA + 0x56

   (e) 0xC2 - 0x73

   (f) 0x73 - 0xC2

   (g) 0x3C - 0xC2

4. Reverse assemble the following HC12 op codes:

   CF 20 00 75 10 01 52 DB 2C 27 F8

   Indicate what instructions these bytes correspond to. For each instruction indicate the addressing mode which is used.

5. Repeat Problem 1 for the following op codes:

   FE 27 F8 E8 6E 04 87 B8 B7 21

6. Which of the conditional branch instructions in the following list will cause a branch to be taken if the condition code flags are: N=0, Z=1, V=0, C=1:

   (a) BCC label

   (b) BNE label

   (c) BGE label

   (d) BGT label

   (e) BHI label

   (f) BMI label

   (g) BLS label

7. Consider the following instructions. Indicate what addressing mode is used, what the effective address is, and what the value of the Y register is after the instruction. Assume for each part the the Y register has a value of 0x1100, and that the first byte of the instruction is at address 0x2000.

    (a) `LDS #$2000`

    (b) `STAA $01`

    (c) `LDD $1234`

    (d) `ADDA -1024,Y`

    (e) `SUBA 2,-Y`

    (f) `DEC 2,Y-`

    (g) `MOVW #$ABCD,8,-Y`

8. Below shows a sequence of instructions to be executed by a 68HCS12. Fill in the table, showing the value in accumulator `A` and the state of the condition flags `N, Z, V` and `C` after each instruction. The table shows the initial value of the condition flags and `A`

Instruction	Accumulator A	N	Z	V	C
	$00	1	0	1	0
LDAA #$80					
DECA					
SUBA #$E2					
CMPA #$3A					
ROLA					
COMA					
CLRA					

9. Write a program to count the number of odd bytes in memory from address 0xff00 to 0xffff. Store this number in address 0x0001.

10. Write a program which puts the exclusive OR of the eight-bit numbers from memory locations 0x8000 through 0x8FFF and store the answer in address 0x0001.