

ECE 271 Homework 5b answers

1) Write a program module that will branch to memory location \$120 if the **unsigned** number in memory location \$2000 is less than 50 (decimal).

```
LDAA  $2000
CMPA  #50
BLO   $120
```

2) Rewrite the above code assuming the number in memory is a **signed** number.

```
LDAA  $2000
CMPA  #50
BLT   $120
```

3) Give an example of a value in memory which would cause one of the above versions of code to branch, but not the other.

Any negative number (128 to 255₁₀)

4) Write a program module that will branch to memory location \$120 if the unsigned number pointed to by the X register is less than the unsigned number which is 5 bytes after the location pointed to by the X register.

```
LDAA  0,X
CMPA  5,X
BLO   $120
```

5) Write a program module that will branch to memory location \$120 if the signed number at memory location \$2000 is negative. For full credit, do not use any of the registers.

```
TST   $2000
BMI   $120
```

6) Write a program module that will subtract the number 50 (decimal) from the **unsigned** number in the A register and then branch to memory location \$120 if the result is incorrect.

```
SUBA  #50
BCS   $120
```

7) Write a program module that will subtract the number 50 (decimal) from the **signed** number in the A register and then branch to memory location \$120 if the result is incorrect.

```
SUBA  #50
BVS   $120
```

8) Write a program module to exchange the numbers in the A and B accumulators.

```
STAA  $2000
TBA
LDAB  $2000
```

9) Explain how the COM instruction is different from the NEG instruction. Give an example of how each might be used.

The COM instruction complements all the bits of the value in memory location or register, changing 0's to 1's and 1's to 0's. The NEG instruction changes the sign of the number in the memory location or register.

Example: To 'flip' the bits as we did in lab 2 we were able to have the LEDs light when the respective switch was activated, we used the COM instruction here.

```
LDAA $1003
COMA
STAA $1004
```

If you are asked us to negate only the positive numbers in a data block you could go through the block using X as a pointer, TST each value and BMI around a NEG statement.

```
BACK    TST    0,X
        BMI    SKIP
        NEG    0,X
SKIP    INX
        BRA    BACK ; actually a conditional branch is needed
```

10) Rewrite the "table copy" example of Figure 2-48 (same in both editions) as actual assembly code (e.g., the first instruction is LDX \$30 rather than LDX DIR). Give the assemble directive that will make the code begin at the correct address.

```
ORG    $C010
LDX    $30
LDAA   $32
STAA   $33
BEQ    $C022
LDAA   0,X
STAA   $20,X
INX
DEC    $0033
BRA    $C016
STOP
```

11) Modify the "table copy" example of Figure 2-48, so it becomes a "block clear" program; i.e., instead of copying the table to somewhere else, set all of its values to \$00.

```
ORG    $C010
LDX    $30
LDAA   $32
STAA   $33
BEQ    $C022
CLR    0,X
INX
DEC    $0033
BRA    $C016
STOP
```