**ECE 222 Post Lab Report 1**

Prepared by

Candice L. Pun

Tae Young Kim

**Hand Assembled Code**

bloop1

4a18 tst.b (A0)+

66fc bne bloop1

5388 suba.l #1,A0

247c 1022 0400 movea.l #BUFFER+1024,A2

\*\*\*\*\*\*\*\*\*\*\* Added Code \*\*\*\*\*\*\*\*\*\*\*\*

224a movea.l A2,A1

reverse\_loop

14e0 move.b -(A0),(A2)+

4a10 tst.b (A0)

66fa bne reverse\_loop

\*\*\*\*\*\*\*\*\*\*\* Added Code \*\*\*\*\*\*\*\*\*\*\*\*

6100 ff96 bsr.w out\_string

6100 ff78 bsr.w out\_crlf

7000 move.l #$0000,d0

4e4f TRAP #15

end

**Added Code with Commenting**

movea.l A2,A1 ;Points the string register to the beginning of the reversed string.

reverse\_loop

move.b -(A0),(A2)+ ;Reverse loop starts here. The last character of the original string,

;which is pointed A0, will be copied to the location pointed by A2,

;which is the start of the reversed string.

tst.b (A0) ;See if we have reached the end of the original string by checking

;the content stored in the location pointed by A0.

bne reverse\_loop ;Keep looping if the content stored in the location pointed by A0 is

;not equal to null.

**DEC2BIN Subroutine**

CLR.L Dy ; clear Dy

ASCII\_BIN ; loop for ASCII-decimal to binary convertion

CLR.L Dx ; clear Dx

MOVE.B (A1)+,Dx ; take an ascii\_decimal character

SUB.L #$30,Dx ; subtract 0x30 to convert to decimal

MULU.W #10,Dy ; multiply current encoded binary by 10

ADD.L Dx,Dy ; add digit to current encoded binary

TST.B (A1)

BNE ASCII\_BIN ; repeat for every character,

; A1 should point to NULL at the end

RTS ; goes back to main program

**BIN2DEC Subroutine**

ASCII\_DEC ; loop for binary to ASCII-decimal convertion

DIVU.W #$000A,D1 ; divide binary by 10

SWAP D1 ; move remainder to lsw of D1

ADD.L #$0030,D1 ; add 0x30 to change to ASCII

MOVE.B D1,(A1)+ ; move remainder (byte) into A1

SWAP D1 ; move quotient to lower side of D1

AND.L #$0000FFFF,D1 ; clear upper side of D1, leaving only quotient

TST.B D1

BNE ASCII\_DEC ; repeat until quotient is 0

MOVE.B #0,(A1) ; add null sign

; A1 points to NULL after normal string

MOVEA.L (4,A1),A2 ; A2 points to start of second buffer

MOVEA.L A2,A0 ; A0 points to start of reversed buffer

REVERSE\_LOOP

MOVE.B -(A1),(A2)+ ; set last letter as first letter of reversed string

TST.B (A1) ; check if we have moved through all characters

BNE REVERSE\_LOOP ; loop until we go through all characters

MOVEA.L A0,A2 ; A2 points back to start of second buffer

RTS ; return to main program