

CpE390 Lab 2

Write Programs to Do Arithmetic Operations

Bring a copy of your homework to the lab if they haven't been graded yet prior to your scheduled lab session. **This lab expects you to demonstrate your working programs assigned in the "Pre-Lab" by running the programs on the EVB.** You can discuss your solutions with your lab partners during the lab session and **pick one** of the best to demonstrate the programs.

PROJECT :

Create the following program:

1. Locate the program at \$8000 using the following memory map equates:

```
PROG EQU $8000
DATA EQU $9000
```

2. Set up 4 variable data buffers of 14 bytes each. Use ORG and DS assembler directives to allocate for BUF1, 2, 3, and 4 as shown. Use the labels BUF1 - BUF4 and refer to them in your program.

```
        ; variable data storage
ORG DATA
BUF1 DS 15      ; allocate 15 bytes of storage starting at DATA
ORG DATA+$10  ; locate the data on a 16 byte boundary
BUF2 DS 15      ; allocate 15 bytes of storage starting at DATA+$10
ORG DATA+$20
BUF3 DS 15
ORG DATA+$30
BUF4 DS 15
```

3. Locate your program appropriately and write the code to make it work as described below. Your program is to include the constant data in buffers DATA1 and DATA2. You suppose initialize a "loop" to move your "DATA" one by one to "BUF", also use Post Increment Addressing Mode to adjust address. Sample code as follow:

```
        ldaa #15 ; register A holds counter value
        ldx  #data1
        ldy  #buf1
loop    movb 1,x+, 1,y+
        dbne a, loop
```

This is your program structure:

```
PROG EQU $8000
DATA EQU $9000

ORG DATA
```

```

BUF1 DS 15
ORG DATA+$10
...
    ORG PROG
    ...
    (write your program here)

```

SWI

; constant data used by the program

```
DATA1 DB $48,$41,$56,$45, $20,$41, $20,$47, $4F, $4F,$44, $20, $44,$41,$59
```

```
DATA2 DB $0F,$04,$F6,$FE,$2F,$0C,$25,$D9,$05, $00,$DC,$2C,$1D,$21,$1A
```

(Note: you can copy these two lines from a text file to avoid having to type them in)

The program is to:

4. Initialize BUF1 with the 14 bytes shown in the DATA1 constant data and BUF2 with the 14 bytes shown in the DATA2.

(Note: BUF1 is a string: HAVE A GOOD DAY)

5. Add each byte in BUF1 with the corresponding byte in BUF2 and store the result in the corresponding byte in BUF3, i.e. [BUF3] = [BUF1] + [BUF2]

6. Exclusive-OR each byte in BUF2 with \$20 and store the results in BUF4. use “eor” or “eorb” to do exclusive-or logic operation.

Sample Code as follow

```

ldab #$20
eorb 1,x+

```

7. Each of the four buffers should be printed exactly as follows:

```
DATA1 DB $48,$41,$56,$45, $20,$41, $20,$47, $4F, $4F,$44, $20, $44,$41,$59
```

```
DATA2 DB $0F,$04,$F6,$FE,$2F,$0C,$25,$D9,$05, $00,$DC,$2C,$1D,$21,$1A
```

```
BUF3 ... (content of BUF3)
```

```
BUF4 ... (content of BUF4)
```