

EE 371 LAB 2- F05 ADDRESSING AND ARITHMETIC

Schedule: 9/14, 15: Pre-lab due, do the lab
9/21, 22: Last time to demonstrate the working program

Name	_____
Partner	_____
MEETING DAY	_____ HR
Test data	_____
Demo (2)	_____

1. Pre-Lab:

Come to lab with a matrix showing exactly what you expect your program to place in the four buffers BUF1, BUF2, BUF3, and BUF4. You should do this before you write your program. You may work on this with your lab partner and hand one in for both of you. During the lab you must demonstrate that your program is performing as required. **Note: If your program is not working 100% correctly when it is demonstrated, a 2 point deduction from the score will be taken. This applies each time you demonstrate so you could, theoretically, end up with a negative score if you are being really obtuse.**

Also . . . Come to the lab with a source file program prepared for the program specified in 2.

You may debug your program and ensure that it is working using the CodeWarrior simulator. You do not have to use the P&E debugger for this lab.

2. Create the following program

a. Your program is to include the constant data in buffers DATA1 and DATA2.

```
MyConst: SECTION...
; Constant data used by the program
DATA1: DC.B    $45,$45,$33,$37,$31,$20,$69,$73,$20,$66,$75,$6e,$21,$21,$20,$20
DATA2: DC.B    $02,$2d,$32,$2a,$43,$3b,$01,$fc,$42,$f7,$cc,$f5,$44,$5d,$5e,$5e
```

b. You are to set up 4 variable data buffers of 16 bytes each. Use the DS.B assembler directives to allocate storage for BUF1, 2, 3 and 4 as shown. Use the labels BUF1 - BUF4 and refer to them in your program.

```
MyData: SECTION
; Variable data storage
BUF1: DS.B 16          ; Allocate 16 bytes of storage starting at memory loc BUF1
BUF2: DS.B 16          ; Allocate storage for BUF2
BUF3: DS.B $10         ; BUF3
BUF4: DS.B $10         ; BUF4
```

The program is to:

c. Initialize BUF1 with the 16 bytes of data shown in the DATA1 constant data and BUF2 with the 16 bytes of constant DATA2.

d. 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)

e. Exclusive-OR each byte in BUF3 with \$20 and store the results in BUF4.

A partial solution using high level design comments for this problem is shown below. To avoid having to type it in you may download L4_partial.asm from the lab's website <http://www.coe.montana.edu/ee/courses/ee/ee371/ee371labs.htm> or from N:\EE371\lab02_f05\

3. Grading: pre-lab test data (5), demo (2) 10 points (-2 points for each demo that is not 100% correct),

```
; EE371 Lab 4 F 04 Partial Solution
; F. M. Cady L4_partial.asm
; Stationery for use in EE371 using the the
; Freescale Student Learning Kit and the Axiom
; CSM12C32 CPU module
; EE371, Fall 2005
; F. M. Cady
;*****
```

```

; Define the entry point for the main program
    XDEF  Entry, main
    XREF  __SEG_END_SSTACK ; Note double underbar
;*****
; Include files
;*****
; Register definitions
;*****
; Constants definitions
;*****
; Code Section
MyCode: SECTION
Entry:
main:
;*****
; Initialize stack pointer register
    lds  #__SEG_END_SSTACK
;*****
; Your program code goes here
; Initialize I/O
main_loop:
; DO_1
;   Init B register to count number of bytes
;   Init X pointing to DATA1
;   Init Y pointing to BUF1
; ENDO_1
; DO initialize BUF1 and BUF2 with data from
; DATA1 and DATA2
DO_WHILE_1:
;   Move byte from DATA1 to BUF1
;   Move byte from DATA2 to BUF2
;   Increment pointers
;   Decrement counter
; WHILE counter != 0
; Branch not equal to zero to DO_WHILE_1:
; ENDOWHILE_1
; Reinitialize the counter and buffer pointer
;   Init memory counter to number of bytes
;   Init X pointing to BUF1
; DO Add and XOR the bytes
DO_WHILE_2:
;   Get value from BUF1
;   Add value from BUF2
;   Store sum in BUF3
;   XOR sum
;   Store in BUF4
;   Increment pointers
;   Decrement the counter
; WHILE counter != 0
; Branch not zero to DO_WHILE_2
; ENDOWHILE_2
; Stay here forever
forever:
    bra  forever
; FOREVER

MyConst:SECTION
; Place constant data here
DATA1:    DC.B  $45,$45,$33,$37,$31,$20,$69,$73,$20,$66,$75,$6e,$21,$21,$20,$20
DATA2:    DC.B  $02,$2d,$32,$2a,$43,$3b,$01,$fc,$42,$f7,$cc,$f5,$44,$5D,$5E,$5E

```

```
;*****
```

```
MyData: SECTION
```

```
; Place variable data here
```