Coding Standards for EELE465

1) Program Headers should follow the following format:

```
;*****************************************************************
;*      Program Name: Lab# - Title of Lab
;*      Author Names:  Both your's and your partner's
;*      Date:
;*      Description:   This should be a succinct and informative description
;*                     of the way in which this program is meant to
;*                     operate. Be brief, but let the reader know what is
;*                     supposed to happen.
;*
;*
;*
;*****************************************************************
```

2) Subroutine headers should be as follows:

```
;*****************************************************************
;*      Subroutine Name: Use the same name as entry label for subroutine
;*      Description:   This should be a succinct and informative description
;*                     of the way in which this subroutine is meant to
;*                     operate. Be brief, but let the reader know what is
;*                     supposed to happen.
;*      Registers Modified: If this subroutine modifies any of the accumulators
;*                          or registers let the reader know which ones
;*      Entry Variables:    List any variables referenced in this subroutine
;*                          that are defined in other files
;*      Exit Variables:     List the variable that will carry values back to
;*                          other portions of the program.
;*****************************************************************
```

3) Code commenting:

Code should be commented in a pseudo code fashion that makes the assembly more readable and makes writing the flow charts easier. For Example:

```
;IF count == 30
        LDA   count
        CMP   #30
        BNE   not_equal_to_30
;THEN do what ever needed for equal condition
        { write code to do then condition}
;END_IF count == 30
not_equal_to_30:
```

Comments at the end of lines should help further understanding of things which are not immediately apparent. This is also a good place to explain “magic numbers” and any “clever” coding done so that it makes sense a week or more from when you were writing it, to someone reading your code for the first time.

Following is a full example from Freescale.

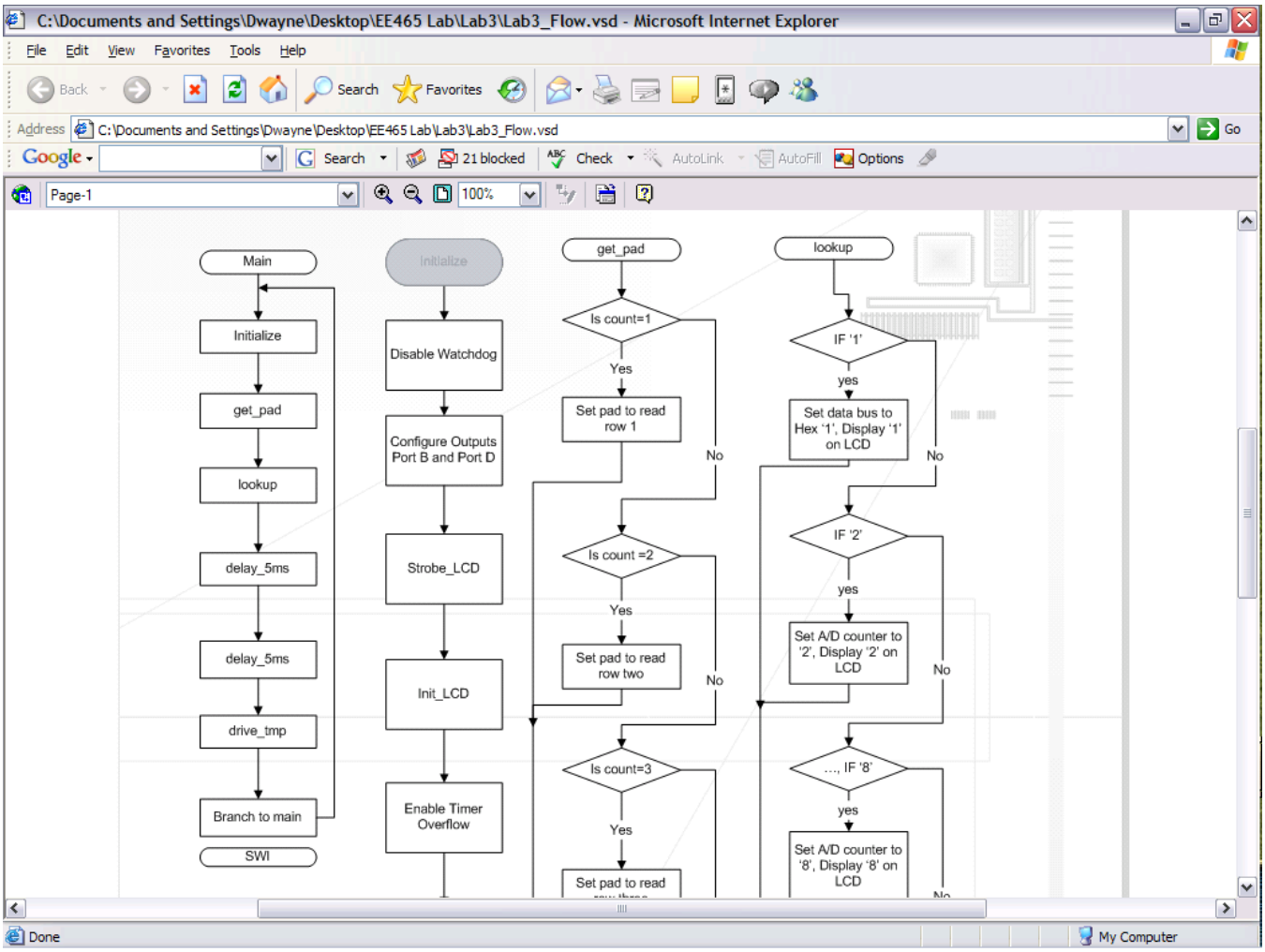
```
;*****************************************************************************

;*****************************************************************************
;                                                                             *
;              Subroutine DelaymS3 - Delay for whole number of milliseconds   *
;                                                                             *
;*****************************************************************************
;                                                                             *
; File Name: delayms3.ASM                    Copyright (c) Motorola 1997     *
;                                                                             *
; Curent Revision: 1.00                                                       *
; Current Release Level: PA                                                   *
; Current Revision Release Date: 02/11/97                                     *
;                                                                             *
; Current Release Written By: David Yoder                                     *
;                             Motorola CMCU Applications - Austin, Texas      *
;                                                                             *
; Assembled Under: IASM08   Ver. 3.03                                         *
;                                                                             *
; Documentation File Name: n/a                    Revision: n/a               *
;                                                                             *
; Brief Description of Module Purpose:                                        *
;             The routine delays for a whole number of milliseconds.          *
;             Minimum = 1mS, Maximum = 256mS.                                 *
;             Assumes an HC08 CPU with 1.2288MHz bus speed.                   *
;             Assumes no bus cycles are stolen by a coprocessor.              *
;             Will Reset a cop_a module during the delay                      *
;                                                                             *
; Part Family Software Module Works With:  HC08                               *
; Part Module Software Module Works With:  cpu8_a, cop_a                      *
; Part Module Software Module Tested With: cpu8_a21, cop_a01                  *
;                                                                             *
; Calling Sequence: LDA #desired number of milliseconds                       *
;                   JSR DelaymS3_Body                                         *
;                                                                             *
; Entry Label:                    DelaymS3_Body                               *
;  Module Size (Bytes):           29 bytes                                    *
;  Stack Space Used (Bytes):      4 bytes                                     *
;  RAM Used (Bytes):              0                                           *
;  Worst Case Execution (Cycles): 314398 cycles (with $00 in Acc)             *
;  Entry Conditions:              # of mS to delay in Acc                     *
;  Part Resources Needed:         CPU08 running at 1.2288MHz bus frequency   *
;  External Variables Used:       none                                        *
;  Subroutines Used:              none                                        *
;  Number of Exit Points:         1                                           *
;    Exit Label:                  DelaymS3015                                 *
;      Exit Conditions:           Registers left as they were when the        *
;                                 routine was called                          *
```

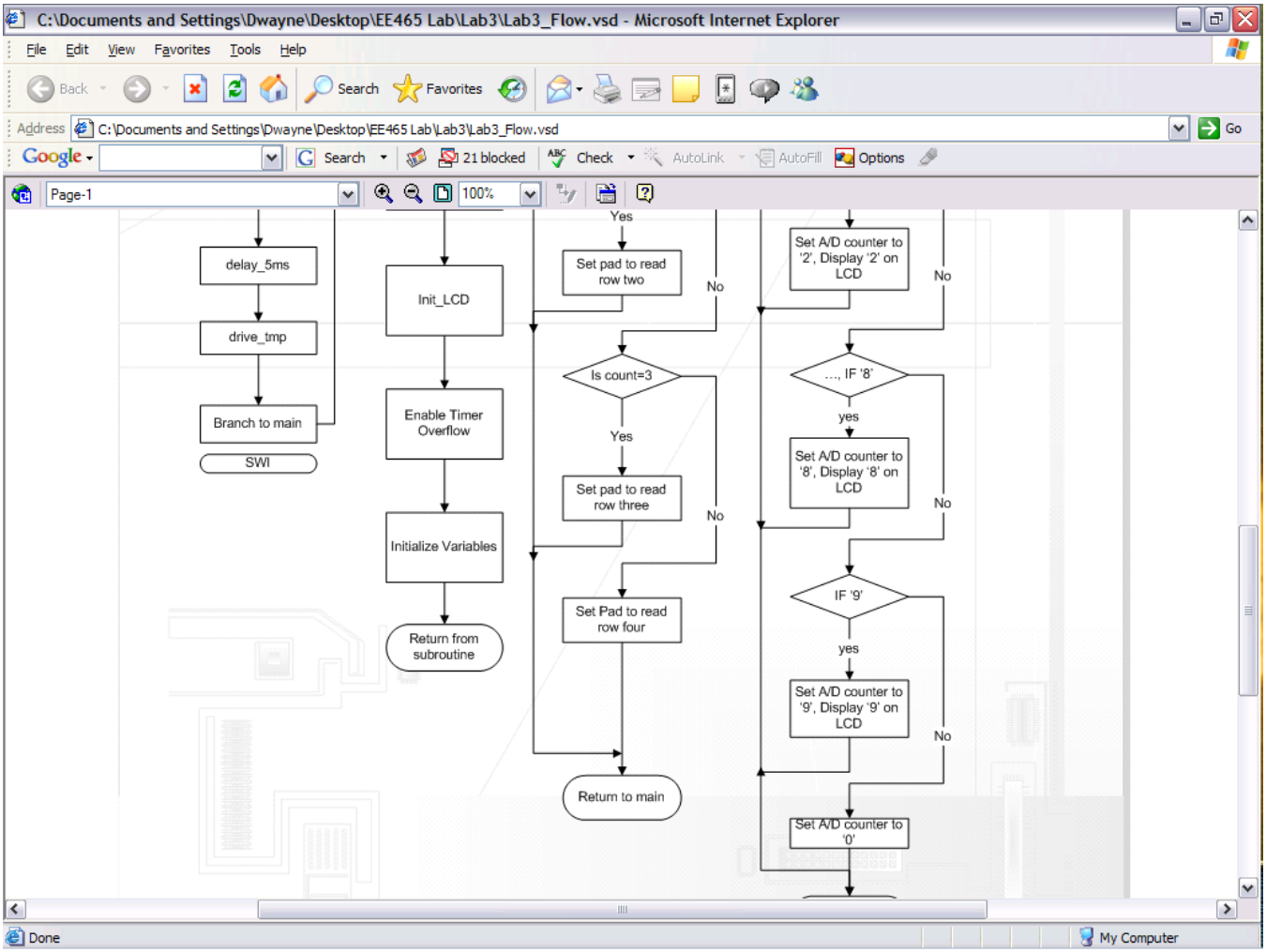
```
;                                                                             *
; Full Functional Description of Module Design:                               *
;                                                                             *
; This routine delays operation for a whole number of milliseconds.           *
; The routine does not alter any registers.                                   *
; Accuracy is +.025%(256mS delay) - +2.1%(1mS delay) of the desired time.     *
; Accuracy is +.064mS(256mS delay) - +.022mS(1mS delay)                       *
; The number of milliseconds to delay is passed in the accumulator.           *
; An 1.2288MHz bus is assumed.                                                *
; The smallest delay is 1256 cycles which occurs when Acc = 1 (1mS).          *
; The largest delay is 314651 cycles which occurs when Acc = 0 (256mS).       *
;                                                                             *
; Please note that passing 0 will NOT result in zero delay, but 256mS delay.  *
;                                                                             *
;                                                                             *
; The number of milliseconds to delay is passed in the accumulator. The       *
; routine is formed by two loops. The inner loop (DelaymS020) executes in     *
; 1224 cycles. The outer loop executes once for each millisecond and adds     *
; 5 bus cyces each time through the loop. This creates 1228 cycles for        *
; each millisecond of delay. The JSR, RTS, and stacking instructions add 30   *
; bus cycles to the total time.                                               *
;                                                                             *
; The COP Watchdog is reset each time through the inner loop.                 *
;                                                                             *
; The exact number of cycles for this routine to execute may be calculated    *
; from:                                                                       *
;                cycles = 27 + ( Acc * 1229 )                                 *
;                                                                             *
; Upon exit, all registers will be returned to their previous values          *
;                                                                             *
;                                                                             *
;*****************************************************************************
;                                                                             *
; Update History:                                                             *
; Rev:      Author:     Date:        Description of Change:                   *
; ----      -------     -----        ----------------------                   *
; ESS 1.0   Yoder       09/13/94     Original Release                         *
; ESS 1.2   Yoder       07/20/95     Ported to MASM 5.0                       *
; ESS 1.3   Yoder       10/16/96     Fixed incorrect delay time               *
;                                                                             *
;*****************************************************************************
;*****************************************************************************
;                                                                             *
; Motorola reserves the right to make changes without further notice to any   *
; product herein to improve reliability, function, or design. Motorola does   *
; not assume any liability arising out of the application or use of any       *
; product, circuit, or software described herein; neither does it convey any  *
; license under its patent rights nor the rights of others. Motorola          *
; products are not designed, intended, or authorized for use as components    *
; in systems intended for surgical implant into the body, or other            *
; applications intended to support life, or for any other application in      *
; which the failure of the Motorola product could create a situation where    *
; personal injury or death may occur. Should Buyer purchase or use Motorola   *
; products for any such intended or unauthorized application, Buyer shall     *
; indemnify and hold Motorola and its officers, employees, subsidiaries,      *
; affiliates, and distributors harmless against all claims, costs, damages,   *
; and expenses, and reasonable attorney fees arising out of, directly or      *
```

```
; indirectly, any claim of personal injury or death associated with such     *
; unintended or unauthorized use, even if such claim alleges that Motorola   *
; was negligent regarding the design or manufacture of the part. Motorola    *
; and the Motorola Logo are registered trademarks of Motorola Inc.           *
;                                                                           *
;***************************************************************************
;***************************************************************************
      PAGE
;***************************************************************************
;                                                                           *
; Delay for XmS                                                             *
;                                                                           *
; Number of mS is passed through the accumulator.                           *
;                                                                           *
;***************************************************************************
DelaymS3_Body:                          ;5 cycles for JSR ext
                psha                    ;2 Stack Accumulator
                pshx                    ;2 Stack Index Register X
                pshh                    ;2 Stack Index Register H
                tax                     ;1 Save Accumulator in Index X
                tpa                     ;1 Get Condition Code reg into Acc
                psha                    ;2 Stack Condition Code reg
                txa                     ;1 Restore Accumulator

DelaymS3010:

DelaymS3020:    ldhx    #$005E          ;  3 Load delay into H:X
DelaymS3030:    aix     #-1             ;  2 Decrement delay
                sta     $FFFF       ;$FFFF=COPCTL ,  4 Kick the WDOG so the part
doesn't reset
                nop                     ;  1 Burn 1 cycle
                cphx    #0              ;  3 Done yet?
                bne     DelaymS3030     ;  3 Branch if not done
                                        ;  3+(13*94)=1225
                nop                     ; 1 burn 1 bus cycle
                dbnza   DelaymS3010     ; 3 decrement # of mS to delay
                                        ;   branch if not done
                                        ; Acc*(1225+4) = Acc * 1229

                pula                    ;2 Unstack Condition Code register data
                tap                     ;1 Restore Condition Code register
                pulh                    ;2 Unstack Index Register H
                pulx                    ;2 Unstack Index Register X
                pula                    ;2 Unstack Accumulator

DelaymS3015:    rts                     ;4 all done - return to calling routine
                                        ;27 + Acc*1229
```

Flow Charting

Flow charts should be setup so that all flow is vertical, and that the execution steps follow the pseudo code written into your program.




C:\Documents and Settings\Dwayne\Desktop\EE465 Lab\Lab3\Lab3_Flow.vsd - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\Dwayne\Desktop\EE465 Lab\Lab3\Lab3_Flow.vsd
Go
Google Search 21 blocked Check AutoLink AutoFill Options
Page-1
100%
delay_5ms
drive_tmp
Branch to main
SWI
Init_LCD
Enable Timer Overflow
Initialize Variables
Return from subroutine
Yes
Set pad to read row two
No
Is count=3
Yes
Set pad to read row three
No
Set Pad to read row four
Return to main
Set A/D counter to '2', Display '2' on LCD
No
..., IF '8'
yes
Set A/D counter to '8', Display '8' on LCD
No
IF '9'
yes
Set A/D counter to '9', Display '9' on LCD
No
Set A/D counter to '0'
Done
My Computer