CodeWarrior Debugger/Simulator Aid

Homework 3 Sample Program

*main.asm - Notepad

File Edit Format View Help

```
*************************************************************************
*
* Title:          LED Light ON/OFF and Switch ON/OFF
*
* Objective:      CMPEN 472 Homework 3 in-class-room demonstration,
*                 Sample Program
*
* Revision:       V3.1  for CodeWarrior 5.2 Debugger Simulation
*
* Date:           Sep. 7, 2020
*
* Programmer:     Kyusun Choi
*
* Company:        The Pennsylvania State University
*                 Department of Computer Science and Engineering
*
* Program:        LED 4 blink every 1 second
*                 ON for 0.2 second, OFF for 0.8 second when switch 1 is not pressed
*                 ON for 0.8 second, OFF for 0.2 second when switch 1 is pressed
*
* Note:
*                 On CSM-12C128 board,
*                 Switch 1 is at PORTB bit 0, and
*                 LED 4 is at PORTB bit 7.
*                 This program is developed and simulated using
*                 CodeWorrior 5.2 only, with switch simulation problem.
*                 So, one MUST set "switch 1" at PORTB bit 0 as an
*                 OUTPUT - not an INPUT.
*                 (If running on CSM-12C128 board, PORTB bit 0 must be set to INPUT).
*
* Algorithm:      Simple Parallel I/O use and time delay-loop demo
*
* Register use:   A: LED Light on/off state and Switch on/off state
*                 X,Y: Delay loop counters
*
* Memory use:     RAM Locations from $3000 for data,
*                 RAM Locations from $3100 for program
*
* Input:          Parameters hard-coded in the program - PORTB
*                 Switch 1 at PORTB bit 0
*                   (set this bit as an output for simulation only - and add Switch)
*                 Switch 2 at PORTB bit 1
*                 Switch 3 at PORTB bit 2
*                 Switch 4 at PORTB bit 3
*
* Output:            LED 1 at PORTB bit 4
*                    LED 2 at PORTB bit 5
*                    LED 3 at PORTB bit 6
*                    LED 4 at PORTB bit 7
*
* Observation:    This is a program that blinks LEDs and blinking period can
*                 be changed with the delay loop counter value.
*
*************************************************************************
* Parameter Declearation Section
*
* Export Symbols
            XDEF        pstart       ; export 'pstart' symbol
            ABSENTRY    pstart       ; for assembly entry point

* Symbols and Macros
PORTA       EQU         $0000        ; i/o port A addresses
DDRA        EQU         $0002
PORTB       EQU         $0001        ; i/o port B addresses
DDRB        EQU         $0003
```

Ln 201, Col 1 | 100% | Windows (CRLF) | UTF-8

*main.asm - Notepad

File Edit Format View Help

```
* Symbols and Macros
PORTA       EQU         $0000       ; i/o port A addresses
DDRA        EQU         $0002
PORTB       EQU         $0001       ; i/o port B addresses
DDRB        EQU         $0003

*****************************************************************
* Data Section: address used [ $3000 to $30FF ] RAM memory
*
            ORG         $3000       ; Reserved RAM memory starting address
                                    ;   for Data for CMPEN 472 class
Counter1    DC.W        $008F       ; X register count number for time delay
                                    ;   inner loop for msec
Counter2    DC.W        $000C       ; Y register count number for time delay
                                    ;   outer loop for sec
* Number $008F and $000C will result 1/10 second delay on my PC

                                    ; Remaining data memory space for stack,
                                    ;   up to program memory start

*
*****************************************************************
* Program Section: address used [ $3100 to $3FFF ] RAM memory
*
            ORG         $3100       ; Program start address, in RAM
pstart      LDS         #$3100      ; initialize the stack pointer

            LDAA        #%11110001  ; LED 1,2,3,4 at PORTB bit 4,5,6,7
            STAA        DDRB        ; set PORTB bit 4,5,6,7 as output
                                    ; plus the bit 0 for switch 1

            LDAA        #%00000000
            STAA        PORTB       ; clear all bits of PORTB

mainLoop
            LDAA        PORTB       ; check bit 0 of PORTB, switch 1
            ANDA        #%00000001  ; if 0, run blinkLED4 20% light level
            BNE         p80LED4     ; if 1, run blinkLED4 80% light level

p20LED4
            JSR         LED4on      ; 20% light level (duty cycle)
            JSR         LED4on
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            BRA         mainLoop    ; check switch, loop forever!

p80LED4
            JSR         LED4on      ; 80% light level (duty cycle)
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4off
            JSR         LED4off
            BRA         mainLoop    ; check switch, loop forever!
```

Ln 201, Col 1 | 100% | Windows (CRLF) | UTF-8

*main.asm - Notepad

File Edit Format View Help

```
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            JSR         LED4off
            BRA         mainLoop     ; check switch, loop forever!

p80LED4
            JSR         LED4on       ; 80% light level (duty cycle)
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4on
            JSR         LED4off
            JSR         LED4off
            BRA         mainLoop     ; check switch, loop forever!


*****************************************************************
* Subroutine Section: address used [ $3100 to $3FFF ] RAM memory
*
;
;*****************************************************************
; LED4 turn-on and turn-off subroutines
;   Students MUST put much comments for the following subroutines
LED4off
            PSHA                     ; save A register
            LDAA        #%01111111   ; Turn off LED 4 at PORTB bit 7
            ANDA        PORTB
            STAA        PORTB
            JSR         delay1sec    ; Wait for 1 second
            PULA                     ; restore A register
            RTS
;
;
LED4on
            PSHA                     ; save A register
            LDAA        #%10000000   ; Turn on LED 4 at PORTB bit 7
            ORAA        PORTB
            STAA        PORTB
            JSR         delay1sec    ; Wait for 1 second
            PULA                     ; restore A register
            RTS
;
;
;
;***************************************************************
; delay1sec subroutine
;
; Please be sure to include your comments here!
;

delay1sec
            PSHY                    ; save Y
            LDY   Counter2          ; long delay by

dly1Loop    JSR   delayMS           ; total time delay = Y * delayMS
            DEY
            BNE   dly1Loop

            PULY                    ; restore Y
            RTS                     ; return

.*************************************************************
```

Ln 119, Col 30 | 100% | Windows (CRLF) | UTF-8

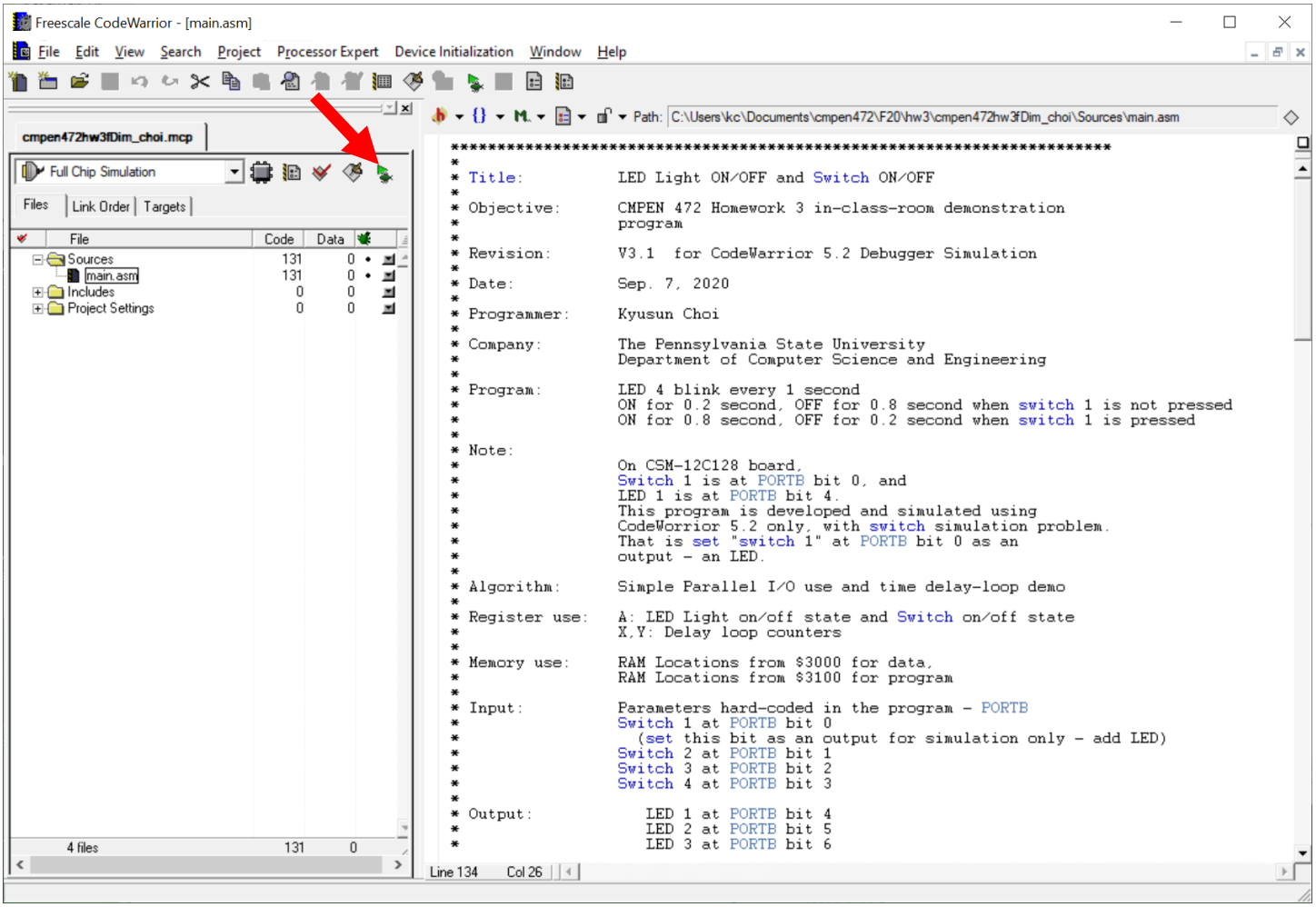
*main.asm - Notepad

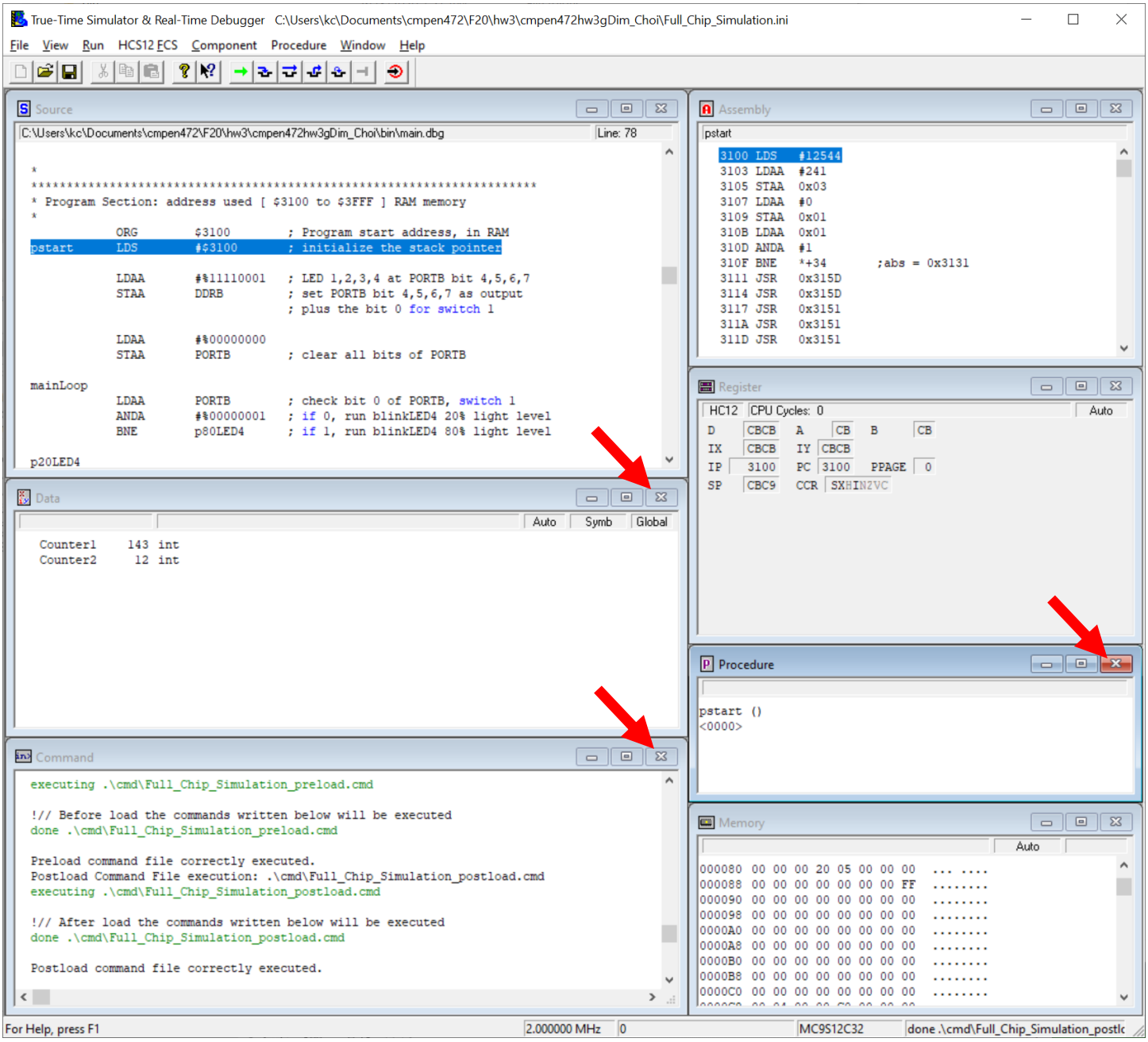
File Edit Format View Help

```
            LDAA        #%01111111   ; Turn off LED 4 at PORTB bit 7
            ANDA        PORTB
            STAA        PORTB
            JSR         delay1sec    ; Wait for 1 second
            PULA                     ; restore A register
            RTS
;
;
LED4on
            PSHA                     ; save A register
            LDAA        #%10000000   ; Turn on LED 4 at PORTB bit 7
            ORAA        PORTB
            STAA        PORTB
            JSR         delay1sec    ; Wait for 1 second
            PULA                     ; restore A register
            RTS
;
;
;
;***************************************************************
; delay1sec subroutine
;
; Please be sure to include your comments here!
;

delay1sec
            PSHY                     ; save Y
            LDY   Counter2           ; long delay by

dly1Loop    JSR   delayMS            ; total time delay = Y * delayMS
            DEY
            BNE   dly1Loop

            PULY                     ; restore Y
            RTS                      ; return

;***************************************************************
; delayMS subroutine
;
; This subroutine cause few msec. delay
;
; Input:  a 16bit count number in 'Counter1'
; Output: time delay, cpu cycle waisted
; Registers in use: X register, as counter
; Memory locations in use: a 16bit input number at 'Counter1'
;
; Comments: one can add more NOP instructions to lengthen
;           the delay time.

delayMS
            PSHX                     ; save X
            LDX   Counter1           ; short delay

dlyMSLoop   NOP                      ; total time delay = X * NOP
            DEX
            BNE   dlyMSLoop

            PULX                     ; restore X
            RTS                      ; return

*
* Add any subroutines here
*

            end                      ;last line of a file
```

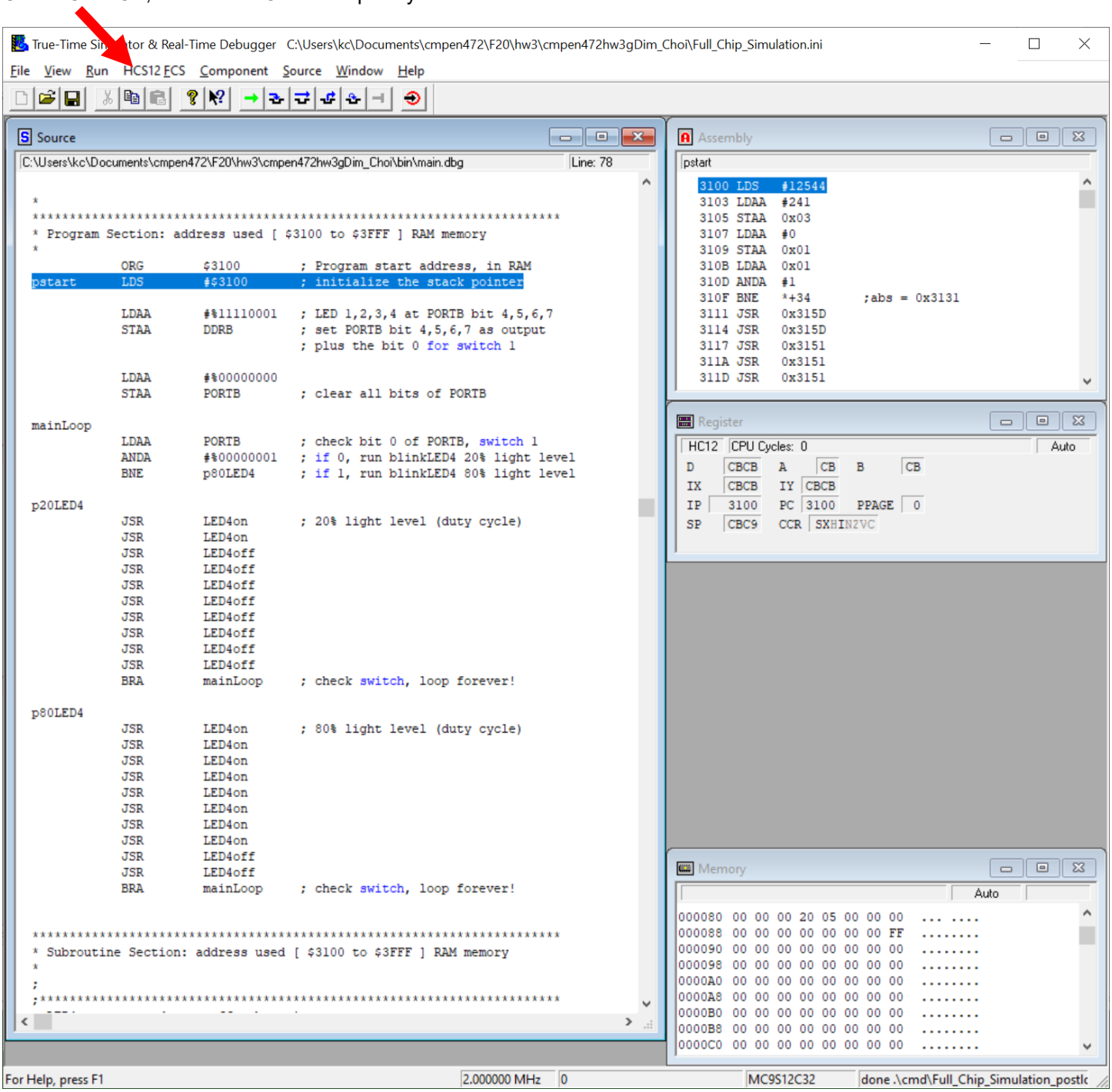
Ln 201, Col 1 | 100% | Windows (CRLF) | UTF-8

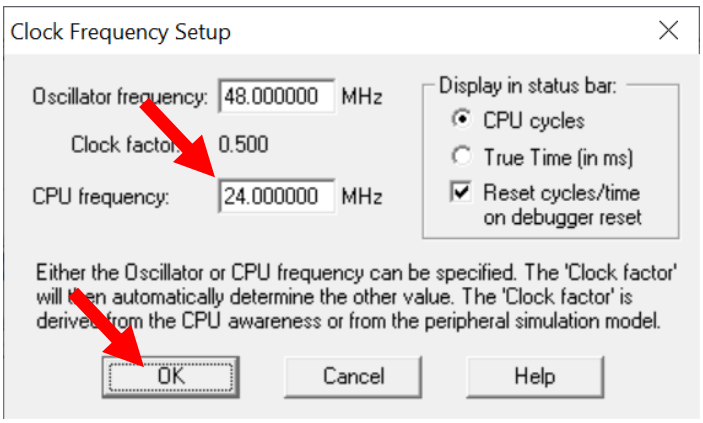
Start CodeWarrior Debugger/Simulator

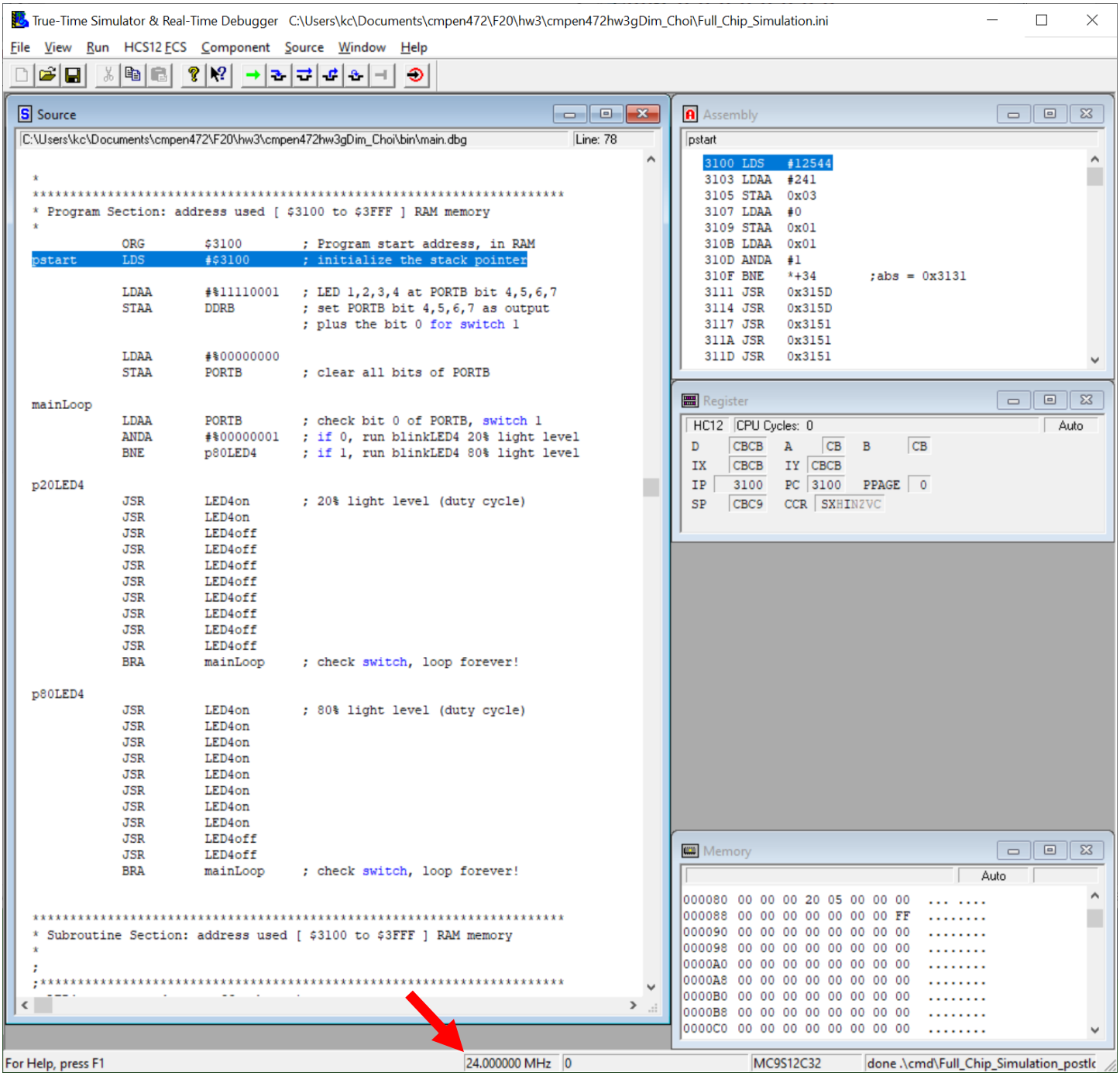



True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw3\cmpen472hw3gDim_Choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Procedure Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw3\cmpen472hw3gDim_Choi\bin\main.dbg
Line: 78
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
LDAA #%11110001 ; LED 1,2,3,4 at PORTB bit 4,5,6,7
STAA DDRB ; set PORTB bit 4,5,6,7 as output
; plus the bit 0 for switch 1
LDAA #%00000000
STAA PORTB ; clear all bits of PORTB
mainLoop
LDAA PORTB ; check bit 0 of PORTB, switch 1
ANDA #%00000001 ; if 0, run blinkLED4 20% light level
BNE p80LED4 ; if 1, run blinkLED4 80% light level
p20LED4
Assembly
pstart
3100 LDS #12544
3103 LDAA #241
3105 STAA 0x03
3107 LDAA #0
3109 STAA 0x01
310B LDAA 0x01
310D ANDA #1
310F BNE *+34 ;abs = 0x3131
3111 JSR 0x315D
3114 JSR 0x315D
3117 JSR 0x3151
311A JSR 0x3151
311D JSR 0x3151
Register
HC12 CPU Cycles: 0 Auto
D CBCB A CB B CB
IX CBCB IY CBCB
IP 3100 PC 3100 PPAGE 0
SP CBC9 CCR SXHIN2VC
Data
Auto Symb Global
Counter1 143 int
Counter2 12 int
Procedure
pstart ()
<0000>
Command
executing .\cmd\Full_Chip_Simulation_preload.cmd
!// Before load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_preload.cmd
Preload command file correctly executed.
Postload Command File execution: .\cmd\Full_Chip_Simulation_postload.cmd
executing .\cmd\Full_Chip_Simulation_postload.cmd
!// After load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_postload.cmd
Postload command file correctly executed.
Memory
Auto
000080 00 00 00 20 05 00 00 00 ... .
000088 00 00 00 00 00 00 00 FF ....
000090 00 00 00 00 00 00 00 00 ....
000098 00 00 00 00 00 00 00 00 ....
0000A0 00 00 00 00 00 00 00 00 ....
0000A8 00 00 00 00 00 00 00 00 ....
0000B0 00 00 00 00 00 00 00 00 ....
0000B8 00 00 00 00 00 00 00 00 ....
0000C0 00 00 00 00 00 00 00 00 ....
For Help, press F1
2.000000 MHz
0
MC9S12C32
done .\cmd\Full_Chip_Simulation_postlc

Click HC12FCS, and select “Clock Frequency . . . . ”

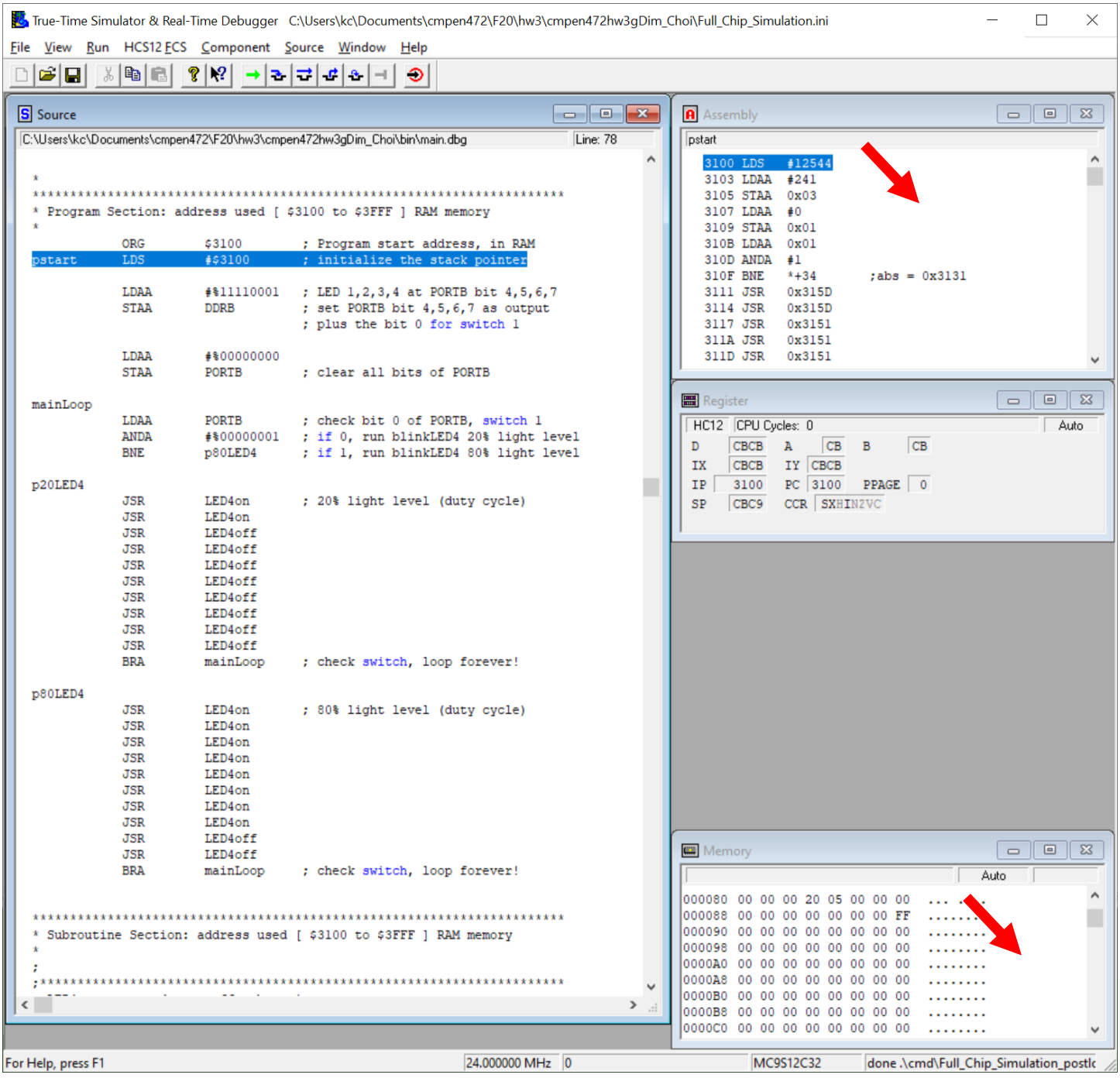


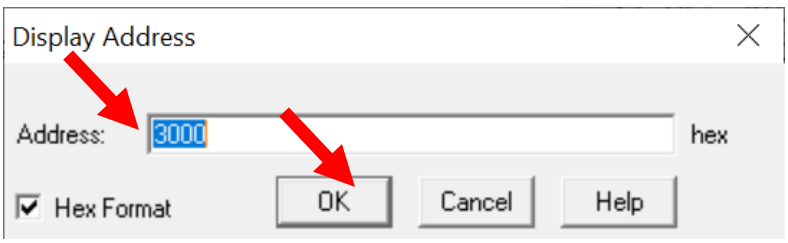


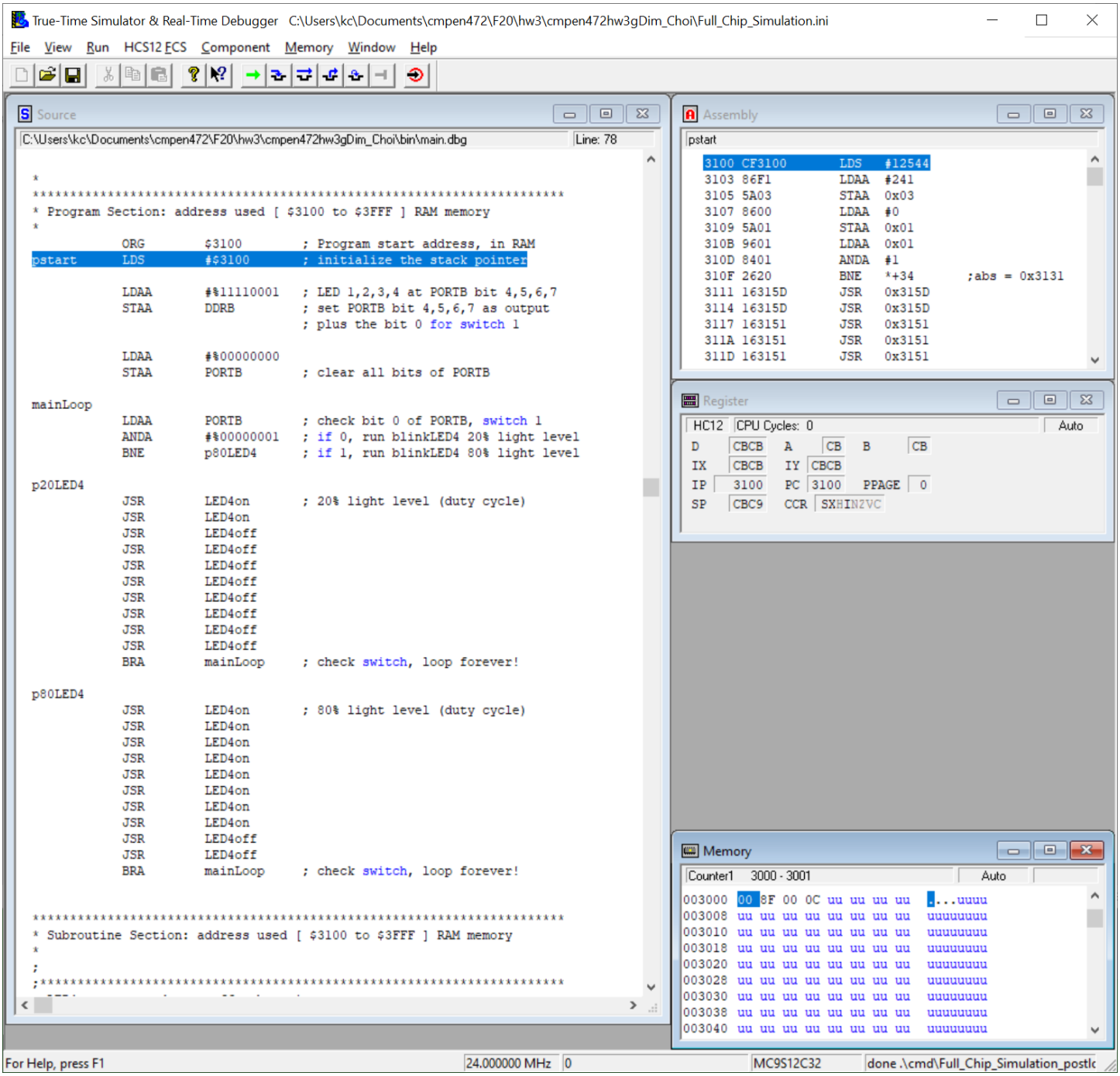

True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw3\cmpen472hw3gDim_Choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Source Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw3\cmpen472hw3gDim_Choi\bin\main.dbg
Line: 78
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
LDAA #%11110001 ; LED 1,2,3,4 at PORTB bit 4,5,6,7
STAA DDRB ; set PORTB bit 4,5,6,7 as output
; plus the bit 0 for switch 1
LDAA #%00000000
STAA PORTB ; clear all bits of PORTB
mainLoop
LDAA PORTB ; check bit 0 of PORTB, switch 1
ANDA #%00000001 ; if 0, run blinkLED4 20% light level
BNE p80LED4 ; if 1, run blinkLED4 80% light level
p20LED4
JSR LED4on ; 20% light level (duty cycle)
JSR LED4on
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
BRA mainLoop ; check switch, loop forever!
p80LED4
JSR LED4on ; 80% light level (duty cycle)
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4off
JSR LED4off
BRA mainLoop ; check switch, loop forever!
* Subroutine Section: address used [$3100 to $3FFF] RAM memory
Assembly
pstart
3100 LDS #12544
3103 LDAA #241
3105 STAA 0x03
3107 LDAA #0
3109 STAA 0x01
310B LDAA 0x01
310D ANDA #1
310F BNE *+34 ;abs = 0x3131
3111 JSR 0x315D
3114 JSR 0x315D
3117 JSR 0x3151
311A JSR 0x3151
311D JSR 0x3151
Register
HC12 CPU Cycles: 0 Auto
D CBCB A CB B CB
IX CBCB IY CBCB
IP 3100 PC 3100 PPAGE 0
SP CBC9 CCR SXHINZVC
Memory
Auto
000080 00 00 00 20 05 00 00 00 ... .
000088 00 00 00 00 00 00 00 FF ....
000090 00 00 00 00 00 00 00 00 ....
000098 00 00 00 00 00 00 00 00 ....
0000A0 00 00 00 00 00 00 00 00 ....
0000A8 00 00 00 00 00 00 00 00 ....
0000B0 00 00 00 00 00 00 00 00 ....
0000B8 00 00 00 00 00 00 00 00 ....
0000C0 00 00 00 00 00 00 00 00 ....
For Help, press F1
24.000000 MHz
0
MC9S12C32
done .\cmd\Full_Chip_Simulation_postlc

Display machine code on Assembly window: right click Assembly window and select “Display”, and then select “Code”.

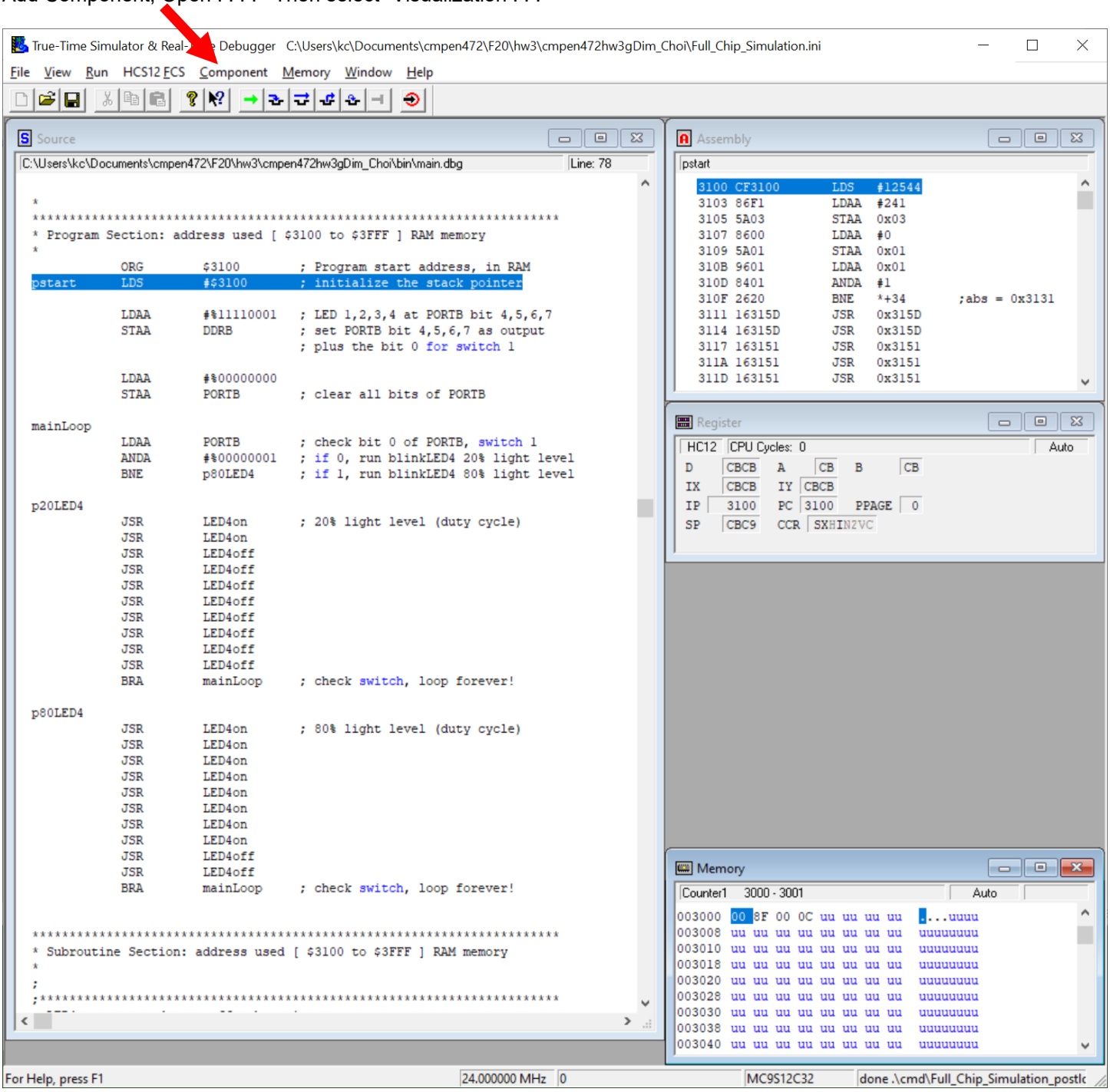
Also display memory location $3000 on the Memory window: right click Memory window and select “Address”. Then type “3000” and click “OK”.

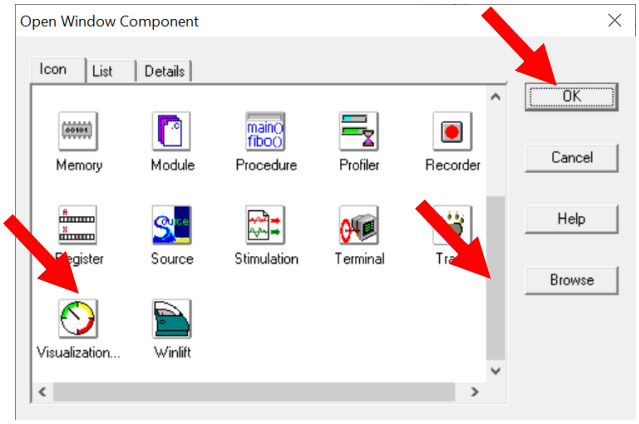




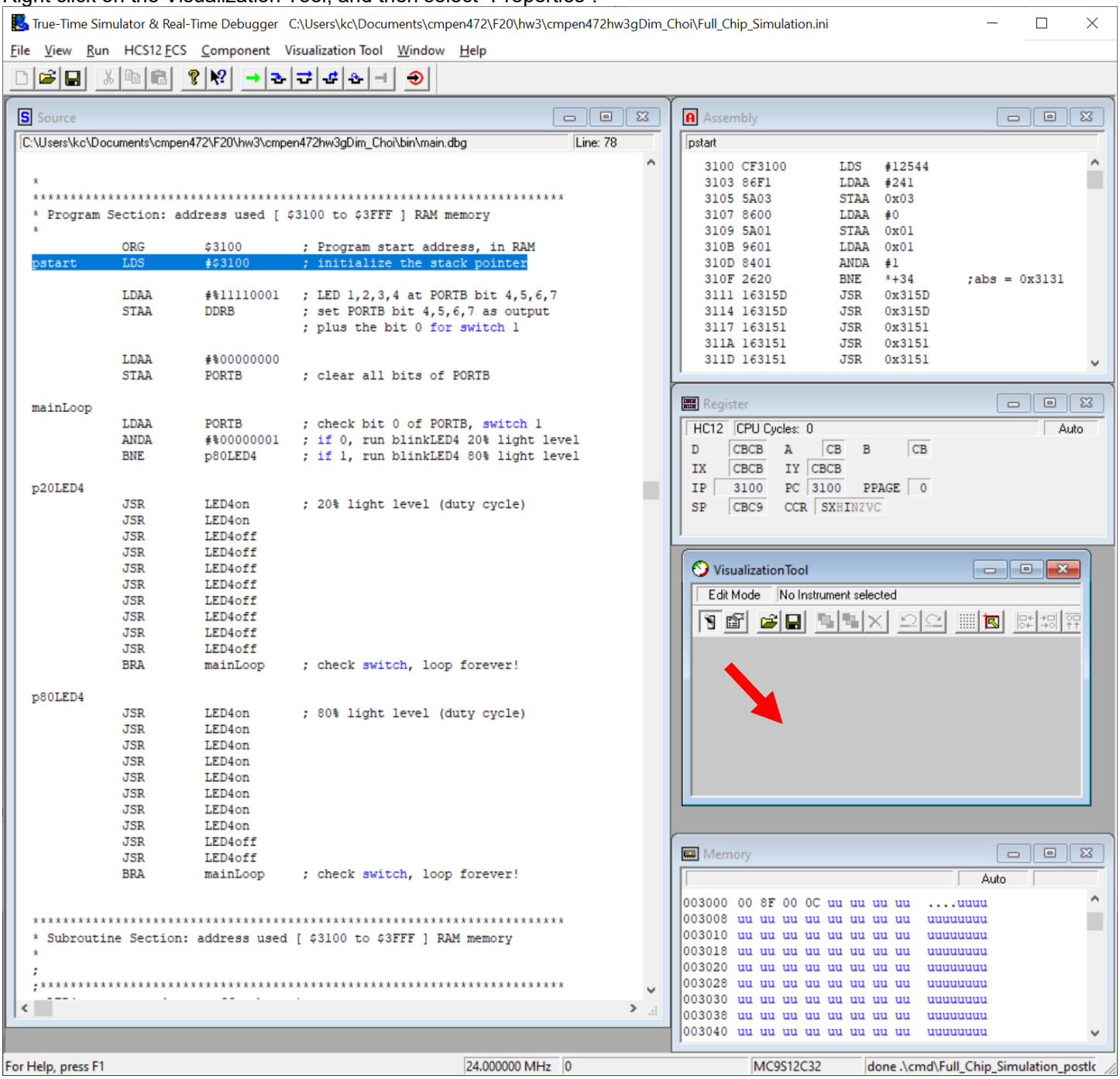

True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw3\cmpen472hw3gDim_Choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Memory Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw3\cmpen472hw3gDim_Choi\bin\main.dbg
Line: 78
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
LDAA #%11110001 ; LED 1,2,3,4 at PORTB bit 4,5,6,7
STAA DDRB ; set PORTB bit 4,5,6,7 as output
; plus the bit 0 for switch 1
LDAA #%00000000
STAA PORTB ; clear all bits of PORTB
mainLoop
LDAA PORTB ; check bit 0 of PORTB, switch 1
ANDA #%00000001 ; if 0, run blinkLED4 20% light level
BNE p80LED4 ; if 1, run blinkLED4 80% light level
p20LED4
JSR LED4on ; 20% light level (duty cycle)
JSR LED4on
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
JSR LED4off
BRA mainLoop ; check switch, loop forever!
p80LED4
JSR LED4on ; 80% light level (duty cycle)
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4on
JSR LED4off
JSR LED4off
BRA mainLoop ; check switch, loop forever!
* Subroutine Section: address used [$3100 to $3FFF] RAM memory
Assembly
pstart
3100 CF3100 LDS #12544
3103 86F1 LDAA #241
3105 5A03 STAA 0x03
3107 8600 LDAA #0
3109 5A01 STAA 0x01
310B 9601 LDAA 0x01
310D 8401 ANDA #1
310F 2620 BNE *+34 ;abs = 0x3131
3111 16315D JSR 0x315D
3114 16315D JSR 0x315D
3117 163151 JSR 0x3151
311A 163151 JSR 0x3151
311D 163151 JSR 0x3151
Register
HC12 CPU Cycles: 0 Auto
D CBCB A CB B CB
IX CBCB IY CBCB
IP 3100 PC 3100 PPAGE 0
SP CBC9 CCR SXHINZVC
Memory
Counter1 3000 - 3001 Auto
003000 00 8F 00 0C uu uu uu uu .....uuuu
003008 uu uu uu uu uu uu uu uu uuuuuuuu
003010 uu uu uu uu uu uu uu uu uuuuuuuu
003018 uu uu uu uu uu uu uu uu uuuuuuuu
003020 uu uu uu uu uu uu uu uu uuuuuuuu
003028 uu uu uu uu uu uu uu uu uuuuuuuu
003030 uu uu uu uu uu uu uu uu uuuuuuuu
003038 uu uu uu uu uu uu uu uu uuuuuuuu
003040 uu uu uu uu uu uu uu uu uuuuuuuu
For Help, press F1
24.000000 MHz
0
MC9S12C32
done .\cmd\Full_Chip_Simulation_postlc

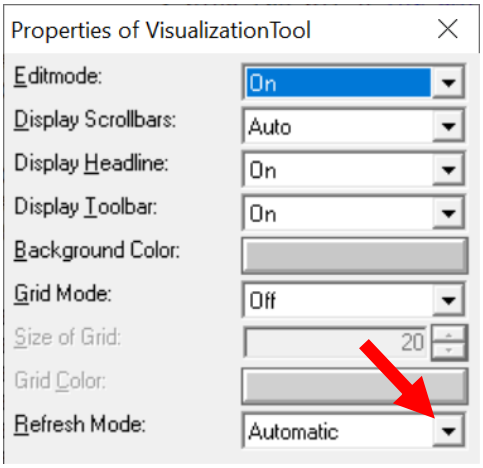
Add Component, Open . . . . Then select “Visualization . . .”

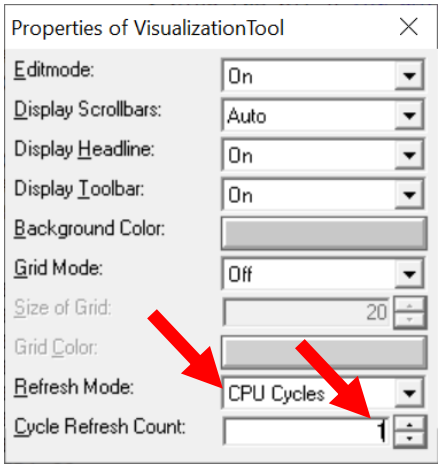




Right click on the Visualization Tool, and then select “Properties”.

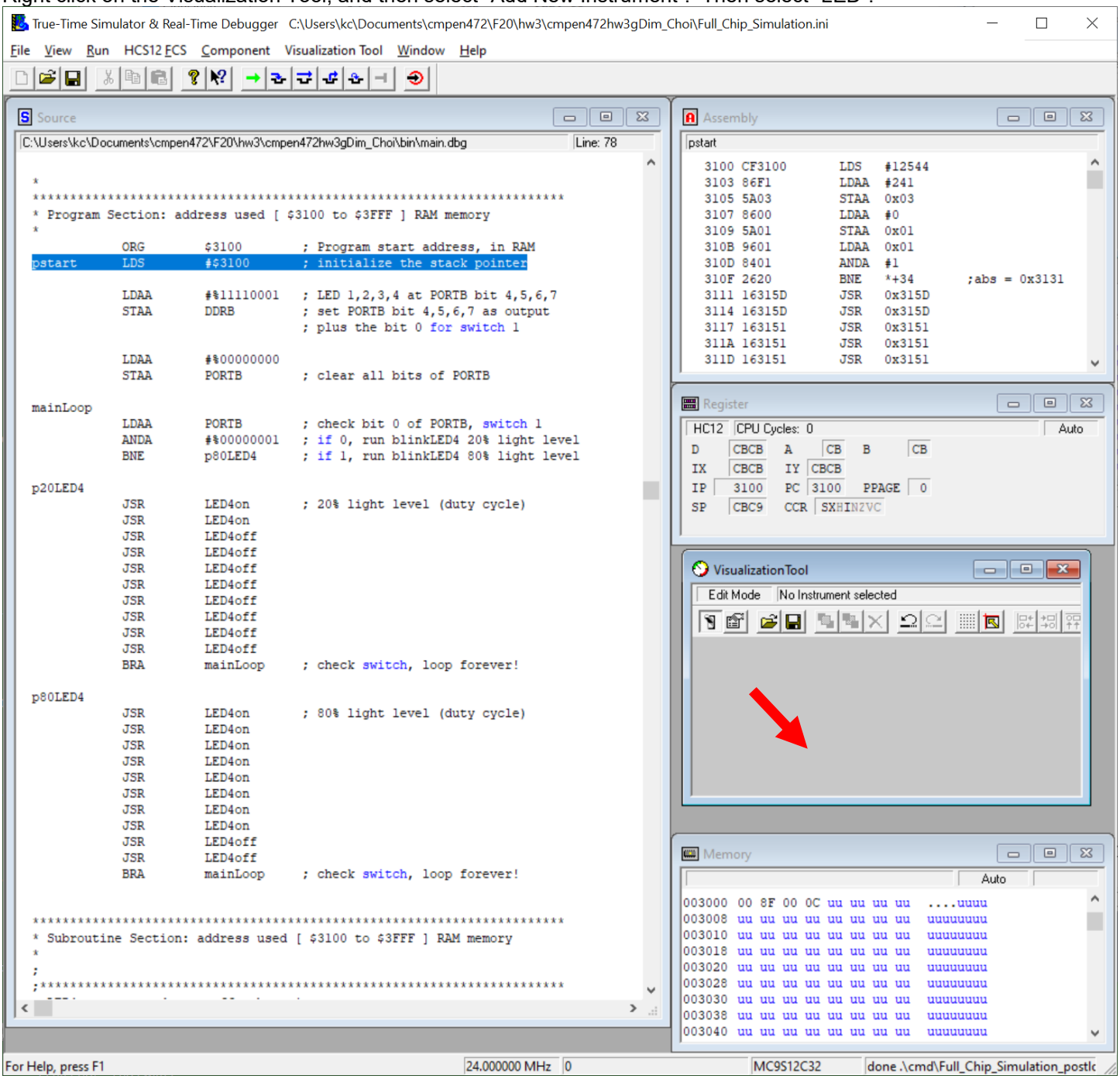




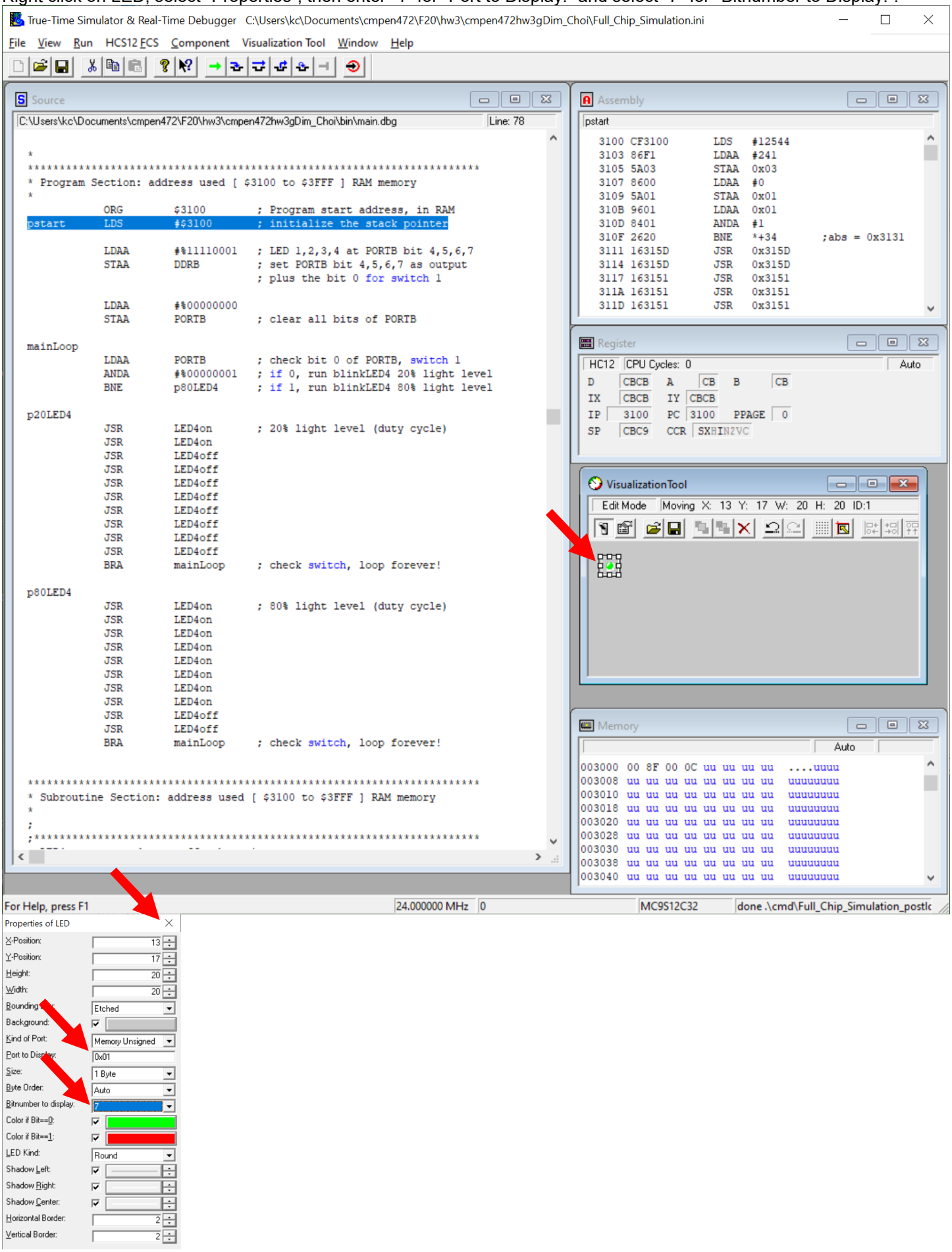


Enter “1” for Cycle Refresh Count.

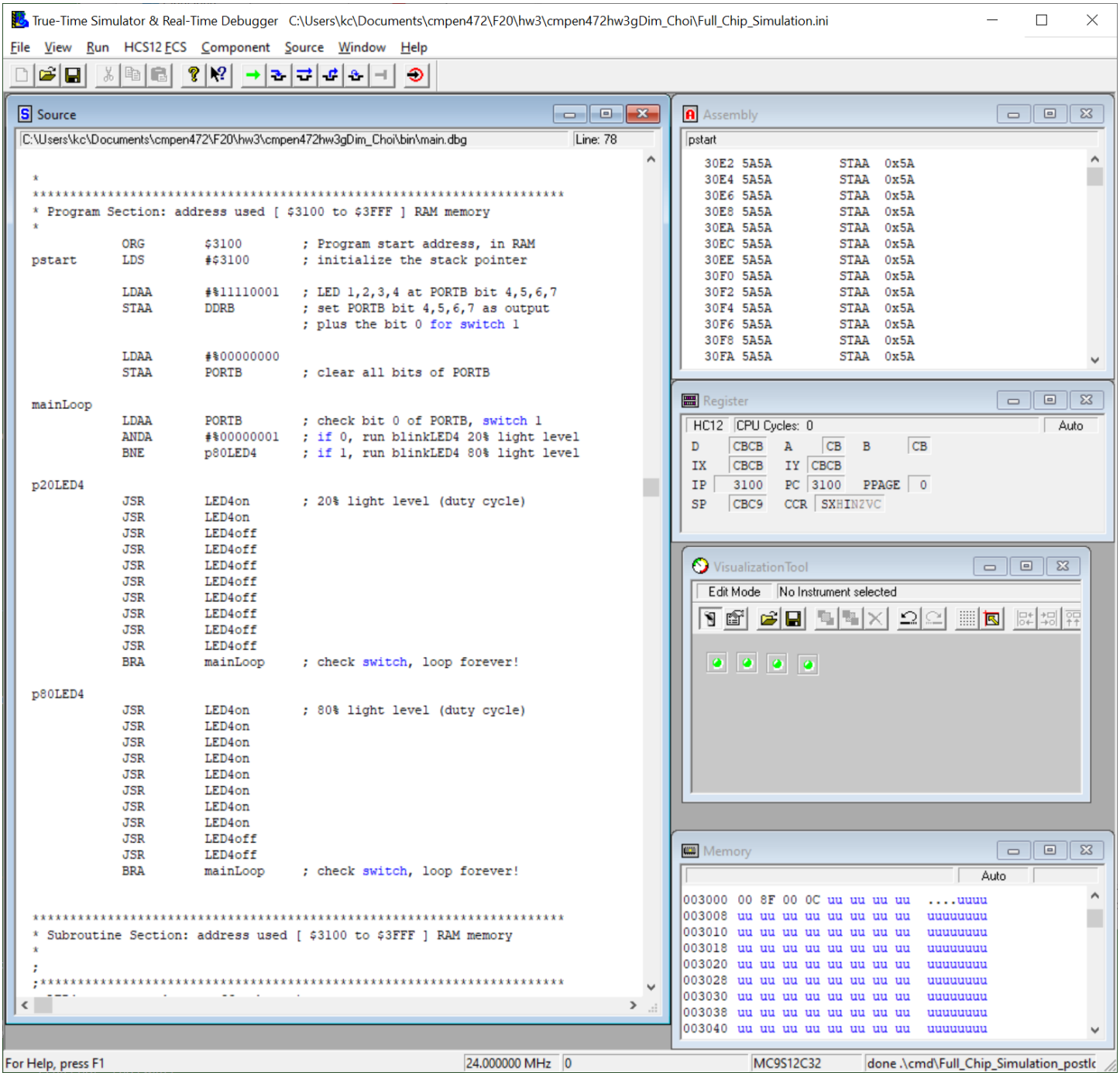
Right click on the Visualization Tool, and then select “Add New Instrument”. Then select “LED”.



Right click on LED, select “Properties”, then enter “1” for “Port to Display:” and select “7” for “Bitnumber to Display:”.



In a similar way, add LED 1, 2, and 3. That is, connect LED 1, 2, 3, and 4 to PORTB bit 4, 5, 6, and 7, respectively.



Right click on the Visualization Tool, and then select “Add New Instrument”. Then select “DIL Switch”.
Be sure to connect the DIL Switch to PORTB through their “properties”. (Please see Homework 2 sample program.)

Now change your Visualization Tool from “Edit Mode” to “Display Mode”, by right clicking the Visualization Tool and select the checked “Edit Mode”. The DIL Switch at PORTB will not work on “Edit Mode”, they work only on “Display Mode”.

Now the Debugger/Simulator is ready to simulate Homework 3 Sample program and you want to SAVE the setting.
Click on the “File” menu and select the “Save Configuration”.

Now run your program and observe LED 4 blinking.

While the program is running, click the Switch 1 in the Visualization Tool (DIL Switch bit 0). And observe the LED 4 blinking duty cycle, changing from 20% to 80%.

Now you can modify the Homework 3 Sample program to finish the Homework 3 program’s full specification.