CSE 3215 Embedded Systems Laboratory

Lab1: CodeWarrior & the Dragon12

Objective

The objective of this lab is to introduce you to developing projects for the Dragon12 using CodeWarrior. Upon completion of this lab you will be able to:

- Create Assembler and C projects in CodeWarrior using predefined stationeries that target the MC9SDP256B processor and the Dragon12.

- Debug projects using the Dragon12's on board resident monitor program D-Bug12.

To meet these objectives you will:
- Create an Absolute Assembler project based on the flowchart shown in Figure 1 and debug your code on the Dragon12 using D-Bug12 by setting, breakpoints, and examining both register and memory contents.

- Create a project written in C based on the flowchart shown in Figure 2 and again debugging your code on the Dragon12 using D-Bug12.

Reference Material

Reference Guide For D-Bug12 Version 4.x.x
 DB12RG4 April 04,2005.pdf

Dragon12 Schematics
 Dragon12 Schematics. PDF

HCS12 Reference Manual
 SC12CPUV2.pdf

MC9SDP256 Advance Information
 MC9SDP256.pdf

Motorola HC12 Assembler
 Manual_Assembler_HC12.pdf

Pre-Lab

1. Run through the online tutorials on "*Creating Projects in CodeWarrior*", "*Using D-Bug12*" and "*Downloading files onto the Dragon12 using HyperTerminal*".

2. Determine from the Dragon12 schematics what I/O port of the MC9SDP256B is connected to LEDS PB[7..0] and what I/O port is connected to the momentary pushbuttons SW2 to SW5.

           LEDS: SW2 to SW5:

3. Determine, again from the Dragon12 schematics the polarity of the momentary pushbutton switches, i.e. are they active low or active, by examining the schematics. Note switches are shown on schematics in the non-active position.

           Polarity:

4. Can you determine from the schematics if there is anything that would prevent your program from detecting a transition on any of the momentary pushbutton switches?

5. Refer to the HCS12 Reference manual and determine what instruction you would use to test the contents of a memory location to see if the value was positive, negative or equal to zero.

6. Refer to the **Reference Guide For D-Bug12 Version 4.x.x** and determine how many hardware breakpoints you can set in D-Bug12 and what command you would use to set the breakpoints?

In Lab Procedure

Part 1

1. Create an absolute assembler project based on the flowchart shown in Figure 1.1 using the _CSE3215_ABS_ASM stationery.

2. Connect the serial cable labeled COM1 to connector P1 on the Dragon12.

3. Verify that the Dragon12 is operating in the EVB mode by ensuring that both switch settings on SW7 to the ON position,SW7 is located on the bottom right corner of the Dragon12.

4. Power up the Dragon12 by plugging the AC adapter in to connector JK3. If the EVB LED is not illuminated adjust SW7 until it is then press the Reset button SW7.

5. Open up HyperTerminal and create a connection to COM1 with the following parameters:

> Bits Per second: 9600
> Data bits: 8
> Parity: None
> Stop bits:1
> Flow control: None

6. Download the ***"HCS12_Serial_Monitor.abs.s19"*** file onto the Dragon12 and run your program. Note you need to know the starting address of your program. The starting address can be obtained by examining either the source file or the listing file. The listing file is located in the …\bin directory of your project.

7. Use D-Bug12 to remove any runtime errors you may have. Note: Pressing the Abort button, SW8 on the Dragon12 halts the program and displays the current contents of the processors registers. At this point you can set/delete break points, display/modify memory locations, single step through your program or resume execution.

Part 3

1. Create a C based project based on the flowchart in Figure 3 using the _CSE3215_MS_C stationery.

2. Download the Monitor.abs.s19 file onto the Dragon12 using HyperTerminal.

3. The starting address for C based programs is different from Assembler based programs. Open the Monitor.Map file and search for the *"Entry point"* to find your programs starting point. The Monitor.Map file is located in the "Linker Map" group of your projects workspace.

Evaluation

1. Prelab (including preliminary code) 20%
2. Lab demonstration, in-lab explanations and answers, debug and test approach 60%
3. Program/design documentation (code should be well documented but no report is required – submit electronically within 24 hours on Prism) 20%

You should be able to demonstrate the following for parts1 & 2:

-   Set/remove breakpoints where ever specified by the lab instructor.
-   Display/modify memory contents.
-   Indicate in both the source and listing files where your program is when halted by the Abort button.
-   Verify that your programs function as per the specifications listed.

```
                              ┌─────────┐
                              │  Entry  │
                              └────┬────┘
                                   │
                                   ▼
                          ┌──────────────────────────┐
                          │ Set LED Port data direction │
                          │ register to output          │
                          └──────────┬──────────────────┘
                                     │
                                     ▼
                          ┌──────────────────────────┐
                          │ Clear contents of LED Port │
                          └──────────┬──────────────────┘
                                     │
                                     ▼
                          ┌──────────────────────────┐
                          │ Set Pushbutton Port data   │
                          │ direction register to input │
                          └──────────┬──────────────────┘
```

Yes — SW1 > 190 / No

Turn LED7 on

Turn LED0 off

Yes — SW1 < 62 / No

Turn LED7 off
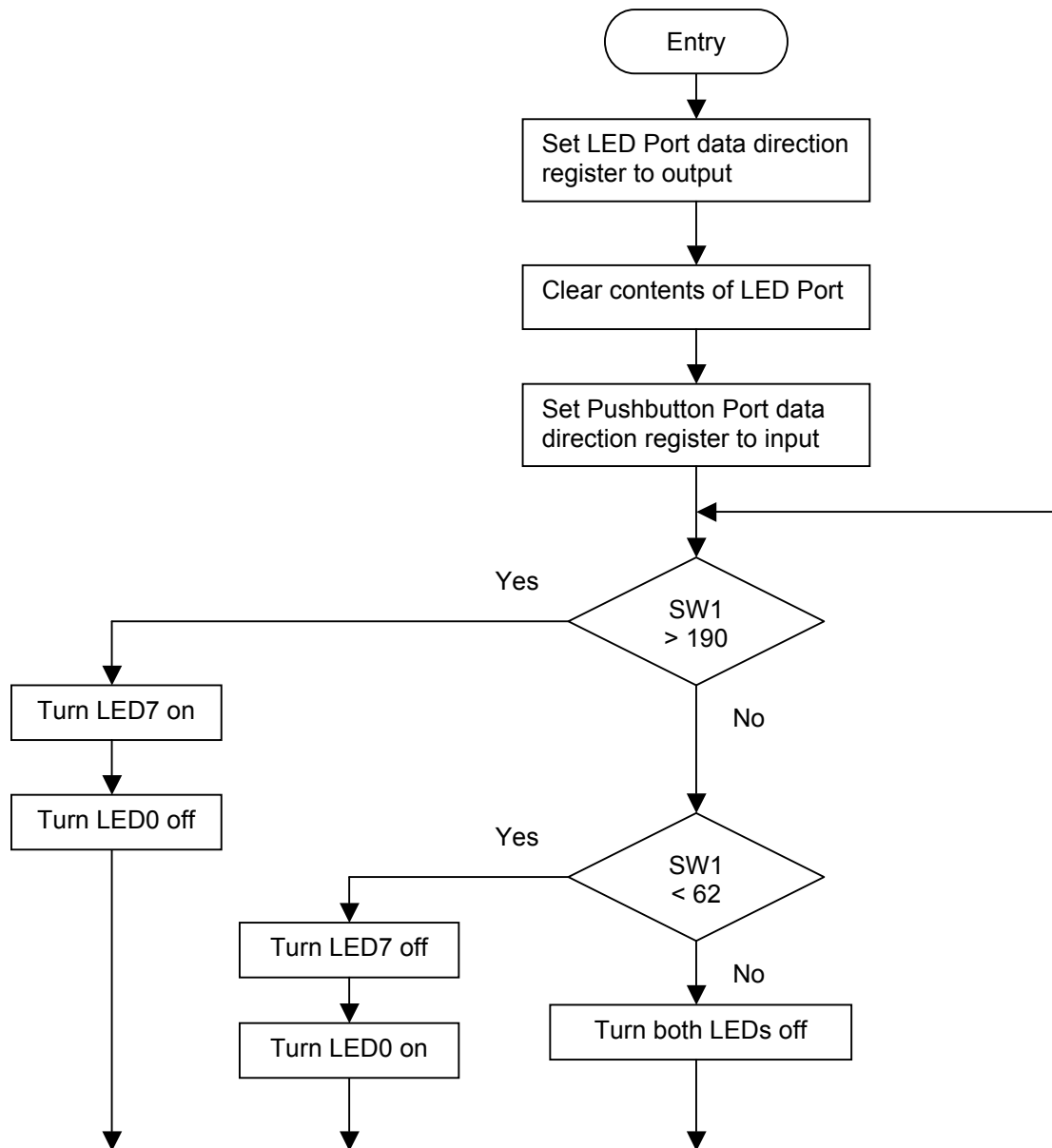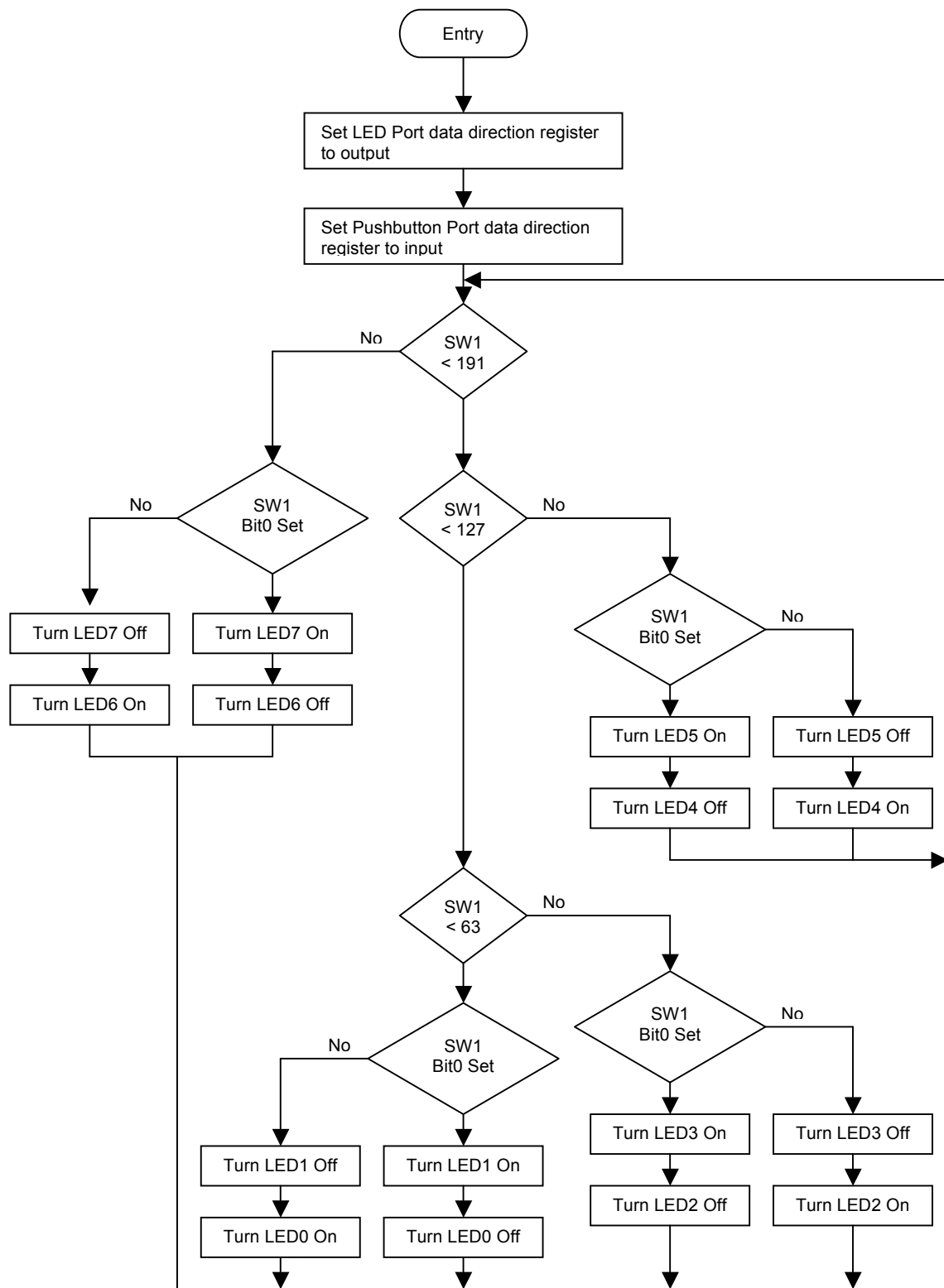
Turn LED0 on

Turn both LEDs off

Figure 1. Decision Making

Figure 1. Decision Making

Figure 2. Decision Making