Electrical, Computer & Energy Engineering
UNIVERSITY OF COLORADO BOULDER

ECEN 5613 Lab-3 Report

Course Name: Embedded System Design
Student Name: ***RUSHI JAMES MACWAN***

This submission is created by Rushi James Macwan. Credits and courtesy to the TAs (Tristan and Kiran) for their immense help and support.

Lab-3 Part-1

SPLD Code Comments:

In this lab, a few input and output signals have been added to the SPLD code. The new additions are as follows:

Input Signals:

1. 8051 /WR (write) signal
2. 8051 /RD (read) signal

Output Signals:

1. NVSRAM /CE (Chip Enable) signal (Notation: CE_n_2)
2. NVSRAM /WE (Write Enable) signal (Notation: WE_n_2)
3. NVSRAM /OE (Output Enable) signal (Notation: OE_n_2)

In context of the above information, the new signals showcase the below explained input-output logical relation:

Inputs	**Outputs**
/WR – This is the 8051 write signal (active low) which goes low whenever a data write is performed to an external memory location.	**/CE (NVSRAM)** – This is the NVSRAM chip enable signal. This signal must go low only when the /EA signal is high (because external data read/write operations are to be performed), A15 signal is low (to play within the memory map range of 400H-7FFFH) and either /WR or /RD is low. This logic can be seen implemented in the SPLD code.
/RD – This is the 8051 read signal (active low) which goes low whenever	**/WE (NVSRAM) –** This is the NVSRAM write enable signal which must follow the

a data read is performed to an external memory location.	8051 /WR signal in order to let the 8051 write data to the external XRAM.
	/OE (NVSRAM) – This is the NVSRAM output enable signal which must follow the 8051 /RD signal in order to let the 8051 read data from the external XRAM.

Schematic Diagram:

The schematic diagram has been included in the submission with relevant modifications.

Wiring Description:

Wire Colour	Purpose
RED	5V Power (Vcc)
BLACK	Ground (GND)
BLUE	Control Signals (e.g. ALE)
YELLOW	Address Bus (A0-A7 and A8-A15)
GREEN	Multiplexed Address and Data Bus (AD0-AD7)
ORANGE	Buffered Data (D0-D7)
GREY	Other necessary connections across the board

Lab-3 Part-2

MSP432 Code:

For this lab, the MSP432 code essentially meets the following three purposes:

1. Provides an echo feature that sends all the received characters to the terminal emulator (TE)
2. A PWM generator that provides a user defined PWM and prints the same on the user's demand to the TE
3. A Temp sensor that provides accurate temp sensing based on the internal temp sensor and prints the same on the user's demand to the TE

The code contains the following architecture:

Source Files:

1. **main.c** – This file contains the main functions that run the entire show. A variable "MODE" determines which of the above purposes has to be served.
2. **init.c** – This file contains all the hardware initialization functions that sets up the clock sources, ADC, PWM pins, timers, etc.
3. **IRQ.c** – This file contains all the interrupt request handlers that act as the interrupt service routines. The main.c file responds to the IRQ handlers present in this file.

4. **print.c** – This file contains all the TE printing functions that I have written myself from scratch. This functions allow an efficient representation of the data on the TE.
5. **temp_init.c** – This file contains the temp sensor initialization functions that refresh the output variables providing data to the functions in the print.c source file.

Header Files:

1. **main.h** – This file contains all the function prototypes and introduces some of the major details of the entire CCS project file.

Submission Questions:

1. What operating system (including revision) did you use for your code development?

   I used the Windows 10 Operating System for my code development.

2. What compiler (including revision) did you use?

   I used the CA51 Compiler kit used by the Keil uVision5 IDE.

3. What exactly (include name/revision if appropriate) did you use to build your code (what IDE, make/makefile, or command line)?

   I used Keil uVision5 IDE for my code development.

4. Did you install and use any other software tools to complete your lab assignment?

   No.

5. Did you experience any problems with any of the software tools? If so, describe the problems.

   There was a trouble in assembling the paulmon21.asm and extra.asm files. As a result, I had to use the command line and AS31 assembler to generate its .hex files.

Conclusion:

As part of this lab, I have pointed out some of the key learnings that I had:

1. Ability to work with MSP432 Firmware Development and Debugging
2. RS-232 communication
3. External XRAM interfacing