ECEN – 5613 – EMBEDDED SYSTEM DESIGN
LAB_2 WRITEUP – SRICHARAN KIDAMBI

CONTENTS:

PART - I

1. Assembly program to blink an LED via External Addressing a 32KB NVRAM.
2. Writing SPLD WINCUPL firmware to enable access to external device.

PART – II

1. Disable External Addressing and Enabling the BLINK – LED operation using the 4KB ROM of the Microcontroller.
2. DEBUGGING the circuit via a DEBUG latch by attaching values for every jump to main and ISR routine in the above Assembly Program.
3. Basics of ARM programming in Embedded C to perform
    a. BLINK LED with 300ms
    b. TOGGLE BETWEEN LED in a fixed time and enabling a switch which controls the toggle operation or not.
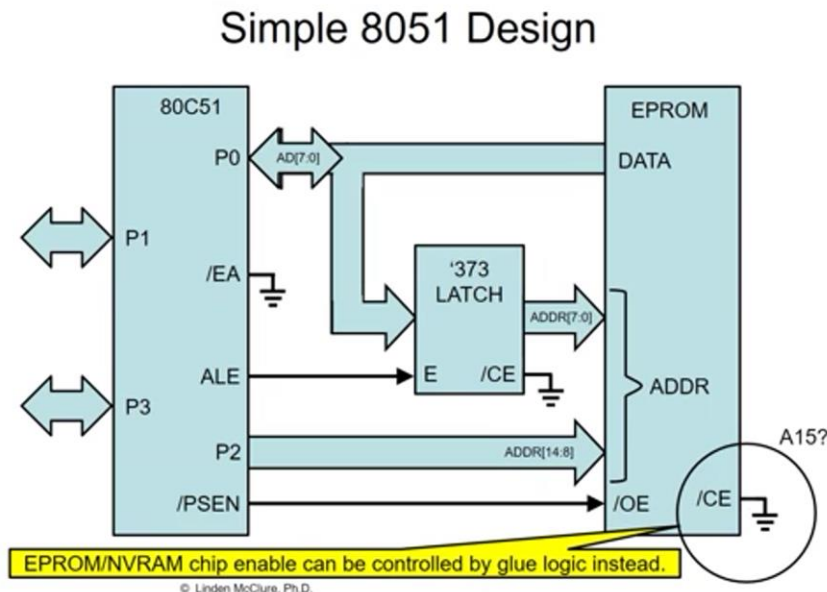

PART – I

Objective: To write a basic assembly level programming for 8051 and configuring basic timers and interrupts to blink an LED at 0.39 second interval.

Tools Required:

       i. M48Z35Y – NVRAM: A 32KB RAM.
      ii. ATF16V8B – SPLD to control the configuration of NVRAM.
     iii. Siemens Microcontroller (C51).
     iv. Soldering and Wire wrapping Tools.

Procedure:

1. Solder the RAM components and ensure the interconnections between MCU and RAM are neatly designed and labelled according to the below diagram.

Simple 8051 Design

EPROM/NVRAM chip enable can be controlled by glue logic instead.

© Linden McClure, Ph.D.

2. Write a basic SPLD configuration to ensure that the output enable of the NVRAM (/CE) is not grounded, rather controlled by A15 signal. The program should be like the below.

CE = A15

Please find the full code in the SRICHARANLAB2PART1.PLD file.

3. Once you successfully designed, compiled, simulated, and dumped to the SPLD code to the SPLD device. Start with writing the Assembly program.

4. Firmware Details of Assembly Program:
   a. Timer0 operated in Mode 1
   b. ISR vector addressing to 000Bh.
   c. 16- Bit counter UP Counter loaded with a value of approximately 39ms.
   d. Needed value = (65536 – load value) * 1.085us.
   e. The above operation is made to function for 10 times after which the interrupt is told to address the main logic.
   f. Once addressed, the timer is again restarted with initial values.

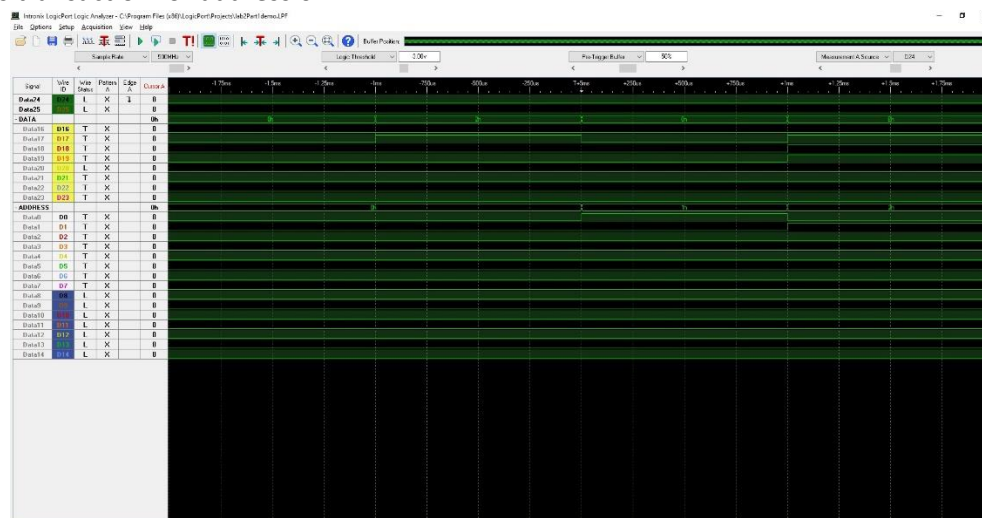   Program found in LABPart.asm in the attached WINZIP file.

5. The LED should toggle at 0.39 msec intervals now.
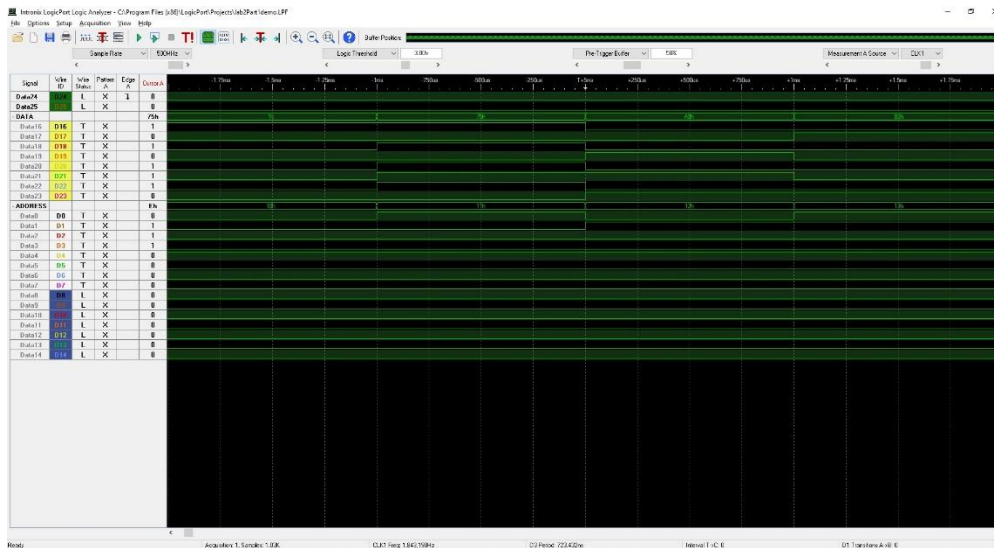
Observations

LOGIC Analyzer is used to view the LAB results for this part.

STATE MODE RESULTS
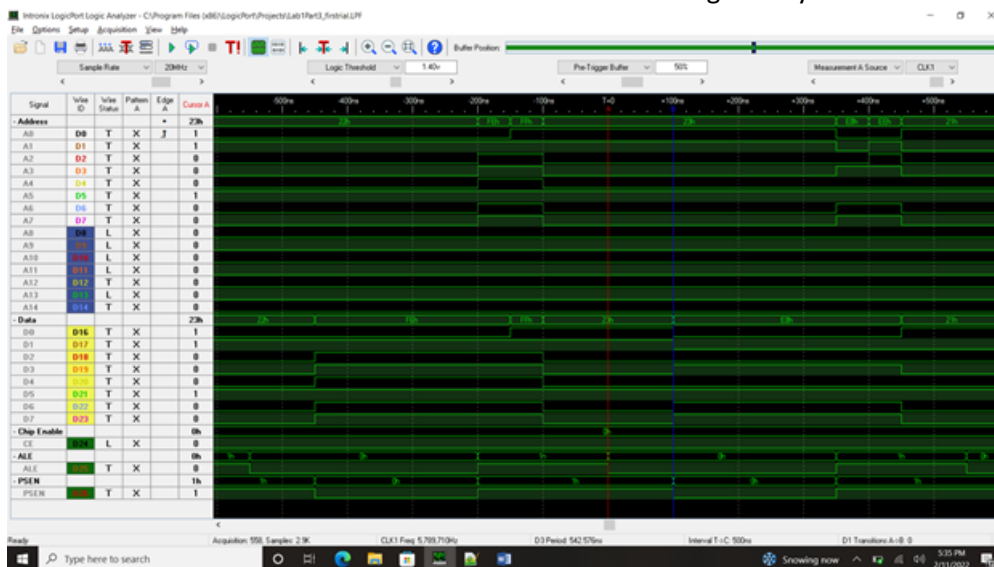
Address transaction for address 0.

Address transaction for address E.

You can compare this with data present on LABPart.lst file.

TIMING MODE RESULTS

Timing Mode uses the clock of the MCU rather than the clock of Logic analyzer to view the results.

The above screenshot mentions that PSEN drops a bit later than ALE so that the bus cycle transaction happens seamlessly. Failing which, the MCU can be a cycle ahead before reaching the data from the previous cycle.

Theoretical Time = (1/11.0592) – 25ns = 90.4 – 25 = 65.4 ns

T_{llpl} time = PSEN – ALE ~= 100ns – Practical situations take a little bit more time. This may be due to:
1. Improper hardware setup.
2. Minor defects with microcontroller as well as LOGIC Analyzer.

KEY LEARNINGS from LAB2_Part1

1. Understood how to configure timers and interrupts on assembly level language for 8051.
2. Obtained good insights on 8051 bus cycle and bus transactions.
3. How to calculate the time between ALE drop and PSEN drop in a program memory read cycle.
4. How to effectively program memory into an NVRAM and verify the contents using chip programmer to ensure that dumping is done correctly.
5. Learnt how to use a logic analyser and verify accurate results.
6. Learnt the difference between timing mode and state mode of logic analyser.

RESULT: The requirements are fulfilled, hardware and software devised properly, and parameters verified successfully.

PART – II

OBJECTIVE:

1. To disable external addressing and ensure that the assembly program mentioned above is working with internal memory of AT89C52 Atmel microcontroller.
2. Add a debug latch and verify its working by adding values to it.
3. Write basic ARM programs in embedded C to understand timers and interrupts on ARM Cortex M4 processor.

TOOLS REQUIRED:

1. MAX 232
2. RS 232
3. AT89C52
4. SN74LS374 DEBUG LATCH
5. 5 1uF Capacitors
6. 3 Pin header for EA
7. 5.1K to act as a pull-down resistor for PSEN.

PROCEDURE:

1. Ensure that connections are made right from the MAX232 to RS232 as well as MAX232 to Microcontroller.
2. Ensure that the Microcontroller is in bootloader mode. i.e RESET address = 00H and SRAM enabled.
3. Make connections according to below diagram.
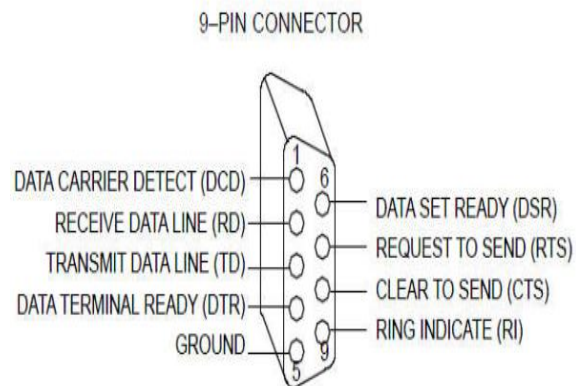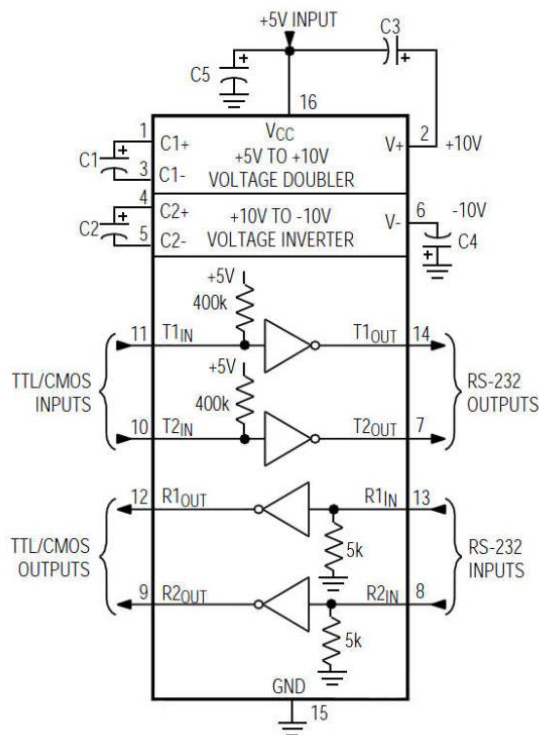
4. Keep the MCU in reset mode. (Hold the reset signal, press the PSEN signal, leave the reset first, PSEN second – The system is now in bootloader mode.

5. Program the device using FLIP and ensure that LED blink is functioning properly.
6. Add a debug latch and add values to the latch each time the data passes through that.
7. Find the assembly code for this in Supplement.asm file.
8. Find the values using logic analyser.
9. Program the basic interrupts and timer program in ARM processor – Programs found in folder lab2 and lab2_2.
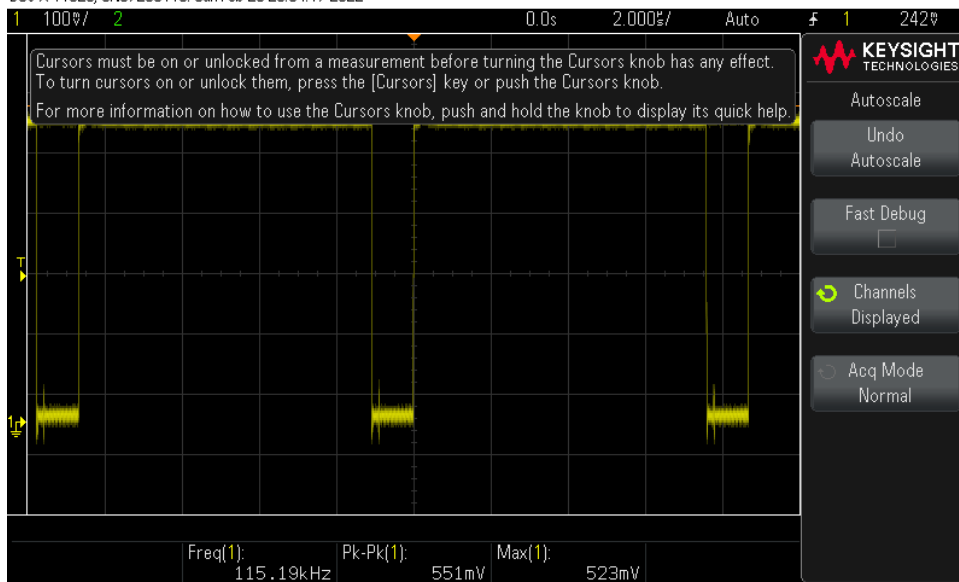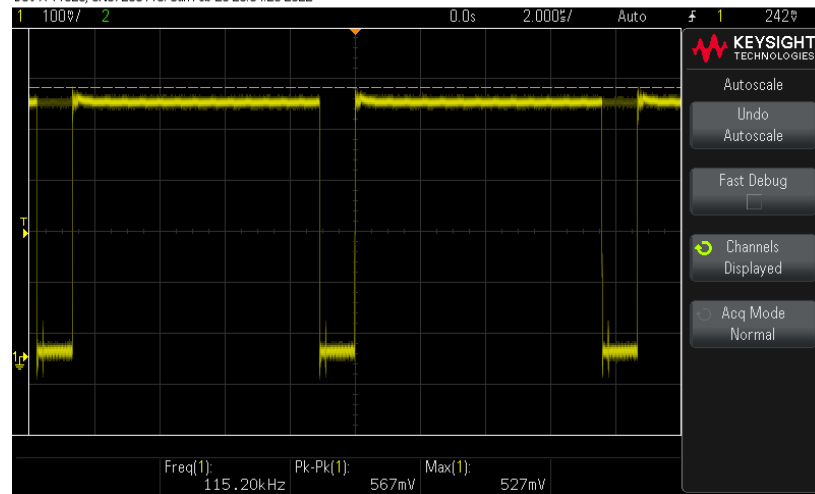
OBSERVATIONS

LOGIC ANALYZER OUTPUT
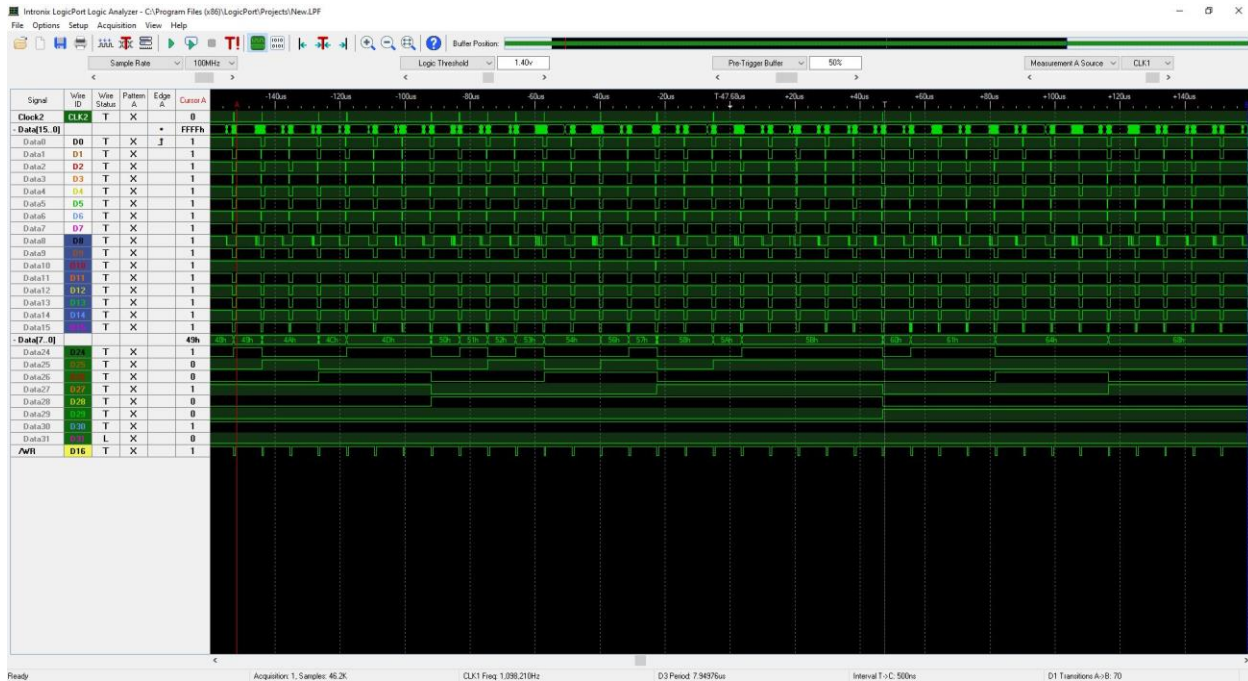
DSO OUTPUT

MCU INPUT – A15

SPLD INPUT – A15

SPLD INPUT - WR
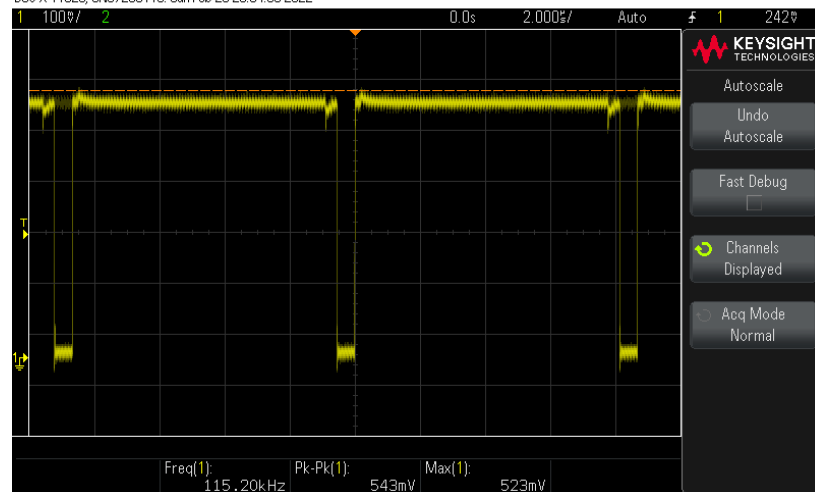
SPLD OUTPUT – !A15 & !WR
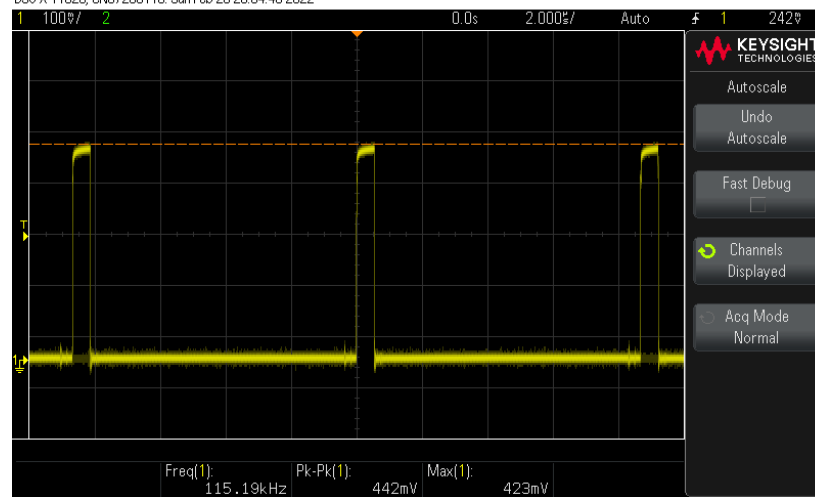
KEY LEARNINGS

1. I am now comfortable with configuring basic timer and interrupts in ARM cortex M4 embedded device.
2. Soldering and wire wrapping comes handy.
3. Configuring DPTR and assigning it to address is something I am comfortable with after doing this lab work.
4. Understood how debug latches work, MAX232, Data terminal equipment to data communication equipment.
5. Obtained an idea on the scenarios where a C51 processor is implemented and an AT89C52 processor is implemented.
6. Understood how to program with flip.
7. Understood the concept of timing analysis, setup and hold time calculations.

RESULT: All requirements designed, coded, validated for this lab and executed successfully.

SIGN OFF SHEETS

You will need to obtain the signature of your instructor or TA on the following items in order to receive credit for your lab assignment. Signatures are due by **Friday, February 11, 2022 (Part 1 Elements)** and **Friday, February 18, 2022 (Part 2 Elements)**.

Print your name below, sign the honor code pledge, and then demonstrate your working hardware & firmware in order to obtain the necessary signatures.

Student Name: SRICHARAN KIDAMBI·S

Honor Code Pledge: "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work. I have clearly acknowledged work that is not my own."

Student Signature: _(signature)_

Signoff Checklist

Part 1 Required Elements

- ☑ Schematic of acceptable quality, correct memory map, SPLD .PLD file
- ☑ Pins and signals labeled, decoupling capacitors, and two 28-pin wire wrap sockets present on board
- ☑ NVRAM (as EPROM substitute), decode logic, and LED functional
- ☑ Understands device programmer.
- ☑ Demonstrated ability to use logic analyzer to capture bus cycles and view fetches from NVRAM. Shows detailed knowledge of both state and timing modes. Captures latched address lines A[15:0], data lines D[7:0], ALE, /PSEN, and NVRAM chip select signal on the logic analyzer display.
- ☑ Shows and discusses logic analyzer screen captures:
- ☑ Assembly program and timer ISR functional:

                                  Moones MD   02/16/22
                                    TA signature and date

Part 2 Required and Supplemental Elements

- ☑ AT89C51RC2, RS-232, and FLIP functional
- ☑ 74LS374 debug port functional
- ☐ Understands timing analysis, setup/hold/propagation
- ☑ ARM code build process, LED program, version control

                                    sskharooh   02/18/22
                                    TA signature and date

Instructor/TA Comments: ☐   ☐   ☐

FOR INSTRUCTOR USE ONLY **Part 1 Elements**	Not Applicable	Poor/Not Complete	Meets Requirements	Exceeds Requirements	Outstanding
Schematics, SPLD code	☐	☐	☐ ✓	☐	☐
Hardware physical implementation	☐	☐	☐ ✓	☐	☐
Part 1 Required Elements functionality	☐	☐	☑	☐	☐
Sign-off done without excessive retries					
Student understanding and skills	☐	☐	☑	☐	☐
Overall Demo Quality (Part 1 Elements)	☐	☐	☐ ✓	☐	☐

FOR INSTRUCTOR USE ONLY **Part 2 Elements**	Not Applicable	Poor/Not Complete	Meets Requirements	Exceeds Requirements	Outstanding
Schematics, SPLD code	☐	☐	☐ ✓	☐	☐
Hardware physical implementation	☐	☐	☐ ✓	☐	☐
Part 2 Required Elements functionality	☐	☐	☐ ✓	☐	☐
Supplemental Elements functionality	☐	☐	☐	☐	☐
Sign-off done without excessive retries			☑		
Student understanding and skills	☐	☐	☐ ✓	☐	☐
Overall Demo Quality (Part 2 Elements)	☐	☐	☐ ✓	☐	☐

NOTE: This signoff sheet should be the top/first sheet of your submission.

Lab 2 Part 1 Signoff (02/16/22)

(+) Assembly code working. with the blinking LED frequency verified

(+) Demonstrated data outputs on logic analyzer using state & timing modes.

(-) Decoupling capacitors & labels missing on the board.

(-) Wire wrapping is slack.

(+) TLPL value shown & verified.

Lab 2 Part 2 Signoff (02/18/22)

(+) Neat schematic & FLIP is functional.

(+) ARM code functional. Add headers & make code modular

(-) Need to do supplemental part

You will need to obtain the signature of your instructor or TA on the following items in order to receive credit for your lab assignment. Signatures are due by **Friday, February 11, 2022 (Part 1 Elements)** and **Friday, February 18, 2022 (Part 2 Elements).**

Print your name below, sign the honor code pledge, and then demonstrate your working hardware & firmware in order to obtain the necessary signatures.

Student Name: SRICHARAN KIDAMBI S

Honor Code Pledge: "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work. I have clearly acknowledged work that is not my own."

Signoff Checklist   **Student Signature:** _____

Part 1 Required Elements

- ☐ Schematic of acceptable quality, correct memory map, SPLD .PLD file
- ☐ Pins and signals labeled, decoupling capacitors, and two 28-pin wire wrap sockets present on board
- ☐ NVRAM (as EPROM substitute), decode logic, and LED functional
- ☐ Understands device programmer.
- ☐ Demonstrated ability to use logic analyzer to capture bus cycles and view fetches from NVRAM. Shows detailed knowledge of both state and timing modes. Captures latched address lines A[15:0], data lines D[7:0], ALE, /PSEN, and NVRAM chip select signal on the logic analyzer display.
- ☐ Shows and discusses logic analyzer screen captures:
- ☐ Assembly program and timer ISR functional:

_____
TA signature and date

Part 2 Required and Supplemental Elements

- ☐ AT89C51RC2, RS-232, and FLIP functional
- ☑ 74LS374 debug port functional
- ☑ Understands timing analysis, setup/hold/propagation
- ☐ ARM code build process, LED program, version control

Manas MD   02/19/22
_____
TA signature and date

Instructor/TA Comments: ☐   ☐   ☐

FOR INSTRUCTOR USE ONLY **Part 1 Elements**	Not Applicable	Poor/Not Complete	Meets Requirements	Exceeds Requirements	Outstanding
Schematics, SPLD code	☐	☐	☐	☐	☐
Hardware physical implementation	☐	☐	☐	☐	☐
Part 1 Required Elements functionality	☐	☐	☐	☐	☐
Sign-off done without excessive retries	☐	☐	☐	☐	☐
Student understanding and skills	☐	☐	☐	☐	☐
Overall Demo Quality (Part 1 Elements)	☐	☐	☐	☐	☐

FOR INSTRUCTOR USE ONLY **Part 2 Elements**	Not Applicable	Poor/Not Complete	Meets Requirements	Exceeds Requirements	Outstanding
Schematics, SPLD code	☐	☐	☐	☐	☐
Hardware physical implementation	☐	☐	☐	☐	☐
Part 2 Required Elements functionality	☐	☐	☑	☐	☐
Supplemental Elements functionality	☐	☐	☐	☐	☐
Sign-off done without excessive retries	☐	☐	☐	☐	☐
Student understanding and skills	☐	☐	☐	☐	☐
Overall Demo Quality (Part 2 Elements)	☐	☐	☐	☐	☐

NOTE: This signoff sheet should be the top/first sheet of your submission.

Comments

(+) Debug latch functional.

(+) Verified rollover case.