Text

Description automatically generated

Embedded System Design Lab 2 Report

Date: 2nd October 2021

Name: Dhiraj Bennadi

**Objective:**

* Add decode logic, an NVRAM (an EPROM substitute), and a status LED to the hardware developed in Lab #1.
* Write simple assembly programs to test NVRAM accesses and perform user I/O.
* Learn how to use timers and write ISRs in assembly.
* Learn how to use a device programmer for code storage.
* Learn how to use a logic analyzer to capture state and timing information.
* Add the AT89C51RC2 processor to your board and enable in-circuit programming via FLIP.
* Continue learning about the ARM architecture and the MSP432 dev board.

**Summary of Tasks:**

* Developed a decode logic in WinCUPL for accessing the NVRAM on which the code was stored. The logic facilitated accessing of memory map in the range of 0000H-7FFFH (32KB of address space).
* Developed a circuit to blink a LED driven by the pin on the Port of the controller.
* Developed a program to blink an LED at a rate 430ms and stored it on the NVRAM.
* Understood generation of Intel and Motorola Hex records.
* Understood the programming of NVRAM using Phyton ChipProg-48 Parallel Programmer
* Understood how to write Assembly Programs using Timers and ISRs
* Understood the Timing and State Mode analysis
* Understood the Programming of that Atmel Controller using the serial communication on FLIP Utility.
* Understood the Bootloader for the Atmel Controller
* Developed C program to Toggle and LED using the Timer A interrupt on MSP432
* Developed C program to enable or disable Toggling of LED using a Push-Button (GPIO Interrupt) using Timer A Interrupt.
* Understood the features of ARM Cortex Controllers and their comparison to 8051 controllers.
* Developed a logic to write values to 74LS374 chip when a write cycle is performed. This was verified on logic analyzer with values controlled by software for a different set of values in the main function and ISR execution loop.
* Understood the timing analysis for setup time, hold time and propagation delays in 74LS374 latch.

**Critical Readings**

1. Toggling of Pin at a rate of 430ms

To generate a delay of 430ms, the Timer 0 ISR of the 8051 was used. The Timer was configured in Mode 1 (16bit Timer) to count from values 0000-FFFFH + 1 for the timer overflow interrupt to occur. This ISR was executed 6 times by loading a value in the R0 register and decrementing it each time the ISR was triggered.

The 80C51 has a external clock of 11.5092 Mhz.

One Oscillator clock period is divided into 12 Machine cycles. The frequency of each machine cycle is 11.0592 Mhz / 12 = 0.9216 Mhz

The period of each machine cycle is 1/f = 1/0/9216 Mhz = **1.085us**

The duration of the ISR to execute is = 1.085 us \* (65535 + 1)

= **71106.56 us**

The total duration of the ISR by decrementing the R0 register each time the ISR is triggered = 71106.56 us \*6

= 426639 us

= **426 ms**

A screenshot of a computer

Description automatically generated with medium confidence

1. Total Duration the program was in the ISR subroutine

The Timer 0 starts the moment TR0 bit is set in the program.   
The duration of program spent in ISR = Total duration for the values configured – Until the first occurrence of the ISR

Duration = 426ms – 71ms

= **355ms**

Graphical user interface

Description automatically generated

1. TLLPL Calculation

The TLLPL Duration is defined as the Time for ALE low to PSEN low.

The TCLCL is defined as the duration of the instance of Clock Low level to the Next low level.

TCLCL = 1 / 11.0592 Mhz

= 0.0904 us

= **90.42 ns**

The minimum permissible duration of the TLLPL = TCLCL - 25

= 90.42ns – 25

= 65 ns

The duration of TLLPL is as follows:

Timing Mode

A screenshot of a computer

Description automatically generated

State Mode

Graphical user interface, Word

Description automatically generated with medium confidence

Duration of TLLPL = Trigger A - Trigger B

= 540ns – 450ns

= **90ns**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**

Duration of TLLPL as measured on Oscilloscope =265ns – 176ns

= 89ns

1. Selection of Resistor Value for status LED Circuit

The voltage drop needed across the LED is approximately 1.6

The voltage available at Source = 5V

The current needed for the LED is approximately 0.5mA. A current of 10ms will be sufficient to glow.

The value of the Resistor connecting the Port to the Base = 5V – 1.6 V / 10 mA

= **340 ohms**

A resistor of **330 ohms** was used as the current limiting resistor in the circuit.

The voltage available at the Pin of Port 0 of the microcontroller is usually less than 5V. For design purposes, a reasonable value of 4.5V as logic high is considered.

The voltage drop across the Base-Emitter = 4.5 – 0.7

= 3.8 V

The base current Ib is usually 93uA.

The resistor value connected to from Port Pin to Base of the resistor = 3.8V / 93 uA

= 4086

= **4 K ohms**

A resistor of **5K** is used in the circuit.

The transistor used was NPN-2222.

1. Memory Map

To access an address range of 0000-7FFFFH,

the number of lines needed are = 32767 / 2n

The value of n = 15

The number of address lines used is 14, A0 -A14

The address line A15 was used as Chip enable signal to the NVRAM

1. Values for Delay of 300ms for LED Toggling in MSP432

The value of SMCLK = 3Mhz

The prescaler used is 2, so the clock frequency is 1.5Mhz

The Timer A in MSP432 is a 16-bit timer.

By approximation, to generate a value of 30ms a value of 0x5848 is loaded into the capture/compare register of Timer A. An output pin was connected to measure the time delay.

For design, a value of **0x1E78** was used to generate a delay of 10ms for scalability.

A global variable was used to count from 0 – 29 to generate a delay of 300ms.

1. Setup Time, Hold Time, and Propagation Delay

**Setup Time (ts):** ts is defined as the minimum time required for the correct logic level to be present at the logic input prior to LE transition from HIGH-to-LOW in order to be recognized and transferred to the outputs.

Minimum value for latch LS374 – 20ns

The value obtained is **170ns**

A screenshot of a computer

Description automatically generated

**Hold Time (th):** th is defined as the minimum time following the LE transition from HIGH-to-LOW that the logic level must be maintained at the input in order to ensure continued recognition.

Minimum value for latch LS374 – 0ns

The value obtained is **770ns**

A screenshot of a computer

Description automatically generated

**Propagation Delay**:

Amount of time after the active edge of the clock required for the new value to become valid.

Minimum value for latch LS374 – No value

Calculated value = 730ns – 170ns

= **560ns**

**A screenshot of a computer

Description automatically generated**

The values for the above parameters obtained in the ISR subroutine and in the main function are approximately equal.

**ISR Debug Values**

**Setup Time (ts):** A screenshot of a computer

Description automatically generated

**Hold Time (th):**

A screenshot of a computer

Description automatically generated

**Propagation Delay**

A screenshot of a computer

Description automatically generated

**Key Learnings from TA Comments**

1. Understanding of Memory Map
2. Understanding of Clock/Dividers to eliminate overhead in ISR
3. Understanding of Debouncing to reduce ISR overload
4. Significance of setup time, hold time and propagation delay
5. Understanding of Address of Bootloader while using the Flip Utility.