REPORT FOR LAB 2

In the lab 2 assignment, I was able to understand the following:-

Part 1:-

1. The lab starts with an assembly level program that helped to learn and apply the knowledge of Timers and Interrupts of 8051. The program helped develop my assembly coding skills.The code required calculation for the ISR timing loop to be 500msec.We know that timer can count to a maximum of FFFFH in timer mode 1 and that for 1.085microsec for every machine cycle it will take about 71 msec to calculate a total of 65535 counts i.e we take 71msec for one timer cycle of FFFFH.For 500ms we will need to loop the timer 7 times before it blinks LED in the ISR.But 71*7=497msec.We are short of 3 msec.So we adjust the count such that we get exactly 500msec.Thus max count -desired count+1 gives us the desired F52FH value that helps us get a 500msec delay for the LED.This program helped me to learn about TMOD and TCON register along with Interrupt flag register.Here I used timer mode 1 in order to access 16 bit timer.The mode 0 was an 8 bit timer so the number of iterations required would have been more.

2.The lab proceeds with the use of NVRAM and interfacing it with the logic analyzer to verify the output.For the LED program,we interfaced the NVRAM with Siemens microcontroller.I was able to learn how to make connections to NVRAM.I learned about the state and timing mode in logic analyzer.In the state mode,I was able to verify the values in the memory with my edsim51 code.I also calculated the maximum and minimum ISR time and

3.For the LED circuit design,after analyzing a few circuits from the LED circuits PDF I decided to use the Design B since it used a transistor with the base connected to port pin.This is because circuit B model,LED should glow only when the Port P1.0 is set.

4. For answering the timing mode question regarding the Tplpl I looked through the Siemens data sheet which mentions that minimum value of tllpl=tclcl-25. Since we are using a 11.0592Mhz frequency we need to calculate tclcl as 1/tclcl=11.0592.Therefore tclcl=90.042nsec.Therefore,tllpl(min)=90-25=65nsec. Therefore we need to get minimum of 65 nsec and we get an output of 100nsec.

Part 2

1.In the second part we had to switch from the Siemens C51 to Atmel AT89C51RC2 which required us to firstly set the program to start from 0000h and clear any garbage that might be present in the atmel chip using Phyton programming.

2.Then for transferring codes to the ATMEL chip using RS 232 we set connections for the same using MAX 232 driver that works using charge pump circuit. For this,five 1uF capacitors were used to connect between C1+/-,C2+/-,V+ and VCC,V- and GND and VCC and GND .I connected the capacitors as per the pin diagram in the slides and made sure that the RS 232 RTS CTS were shorted along with DSR,DTE and DCE.The FLIP was used to transfer the code

according to the steps mentioned on the website.The bootloader push button with a resistor were also implemented.

3.In the supplemental part,I was able to figure out the code but failed to get a logic analyzer output for the TA sign offs.The latch 74374 consists of a clock pin,OE pin and I/O pins.For performing the write operation,we use WR pin of microcontroller as the clock for the debug circuit,OE is connected to ground since it is an active low pin.Address lines A0-A7 were connected as inputs and outputs were connected to logic analyzer to check output.

The code was as follows:

```
ORG 0000H ;START OF PROGRAM
LJMP MAIN ;MAIN PROGRAM
ORG 000BH ;TIMER INTERRUPT ADDRESS
LJMP ISR ;JUMP TO ISR
MAIN:
MOV R3,#80H          ;set up counter for ISR
MOV R4,#00H          ;set up counter for main loop
MOV DPTR,#1000H  ;move address 1000h to data pointer
MOV A,R4     ;move counter to A register
MOVX @DPTR,A ;move value to address pointed by Data pointer
INC R4           ;increment counter
MOV TMOD,#01H ;INITIALIZE MODE 1
MOV TH0,#0F5H ;SET TIMER TO 0
MOV TL0,#02FH ;SET TIMER TO 0
SETB TR0 ;START THE TIMER
MOV R0,#08H
MOV IE,#82H ;SET INTERRUPT FLAG
HERE:SJMP HERE

ISR:
MOV A,R3         ;move the counter to register A
MOV DPTR,#4000H ;move the address 4000h to data pointer
MOVX @DPTR,A     ;move the count to the external memory using MOVX instruction
INC R3 ;increment count
DJNZ R0,X ;decrement And jump if it is not zero
SETB P1.2 ;set port p1.2
CPL P1.0 ;complement led
MOV R0,#08H ;reset the resistor to 8 for 500ms
MOV TH0,#0F5H
MOV TL0,#02FH
X:CLR TF0 ;clear the overflow flag
CLR P1.2
RETI ;return from ISR to main
```