Lab 1: Simulation & Hardware Implementation

EE222: Microprocessor Systems

Contents

1 Acknowledgements

This lab exercise is prepared by Mohammad Azfar Tariq and Muhammad Usman under the supervision of Dr. Rehan Ahmed for the course EE-222 Microprocessor Systems. Reporting any error or discrepancy found in the text is appreciated.

2 Administrivia

2.1 Objectives

By the end of this lab you will be able to:

- Simulate and understand an Assembly code implementation
- Qualitatively and quantitatively Compare similar implementations

2.2 Deliverables

You are required to submit

- Simulated observations in a tabular form
- Quantitative and qualitative comparisons

in the beginning of next lab.

3 Hardware Resources

- ATmega16A microcontroller Unit
- Universal Programmer
- LEDs (may use from trainer kit)
- Switch or button (may use from trainer kit)
- Power source with voltage regulator (may use from trainer kit)

4 Introduction

You should not worry about nitty gritty details of code for the moment. We will simulate an altered form of the code and burn the code in microcontroller. You can find this code here[1] as well.

```
                            ; initial constants
    ldi R16, 0xFF
    ldi R17, 0xFF
                            ; set DDRB as output
    out DDRA, R17
                            ; code to toggle LEDs
toggler:
    subi R16, 0xFF
    out PORTA, R16
    rjmp idle_loop
                            ; delay loop
idle_loop :
    ldi R19, 0xFF
    ldi R20, 0x0F
    ldi R21, 0x01
idle_loop_0 :
idle_loop_1 :
idle_loop_2 :
    dec R19
    brne idle_loop_2
    dec R20
    brne idle_loop_1
    dec R21
    brne idle_loop_0
rjmp toggler
```

In the above assembly program, ldi, out, sub, etc. are assembly instructions mnemonics, toggler, idle_loop, etc. are labels and lines starting with a semi-colon are commnets.
The code is designed to count from 0x00 to 0xFF which will be displayed on 8 LEDs. There are two nested loops in the code, the outer loop (named "toggler" in the code) increments the count on each iteration and the inner loop (named "idle_loop" in the code) just iterate for some time doing nothing, wasting the time of CPU (a sort of time delay).

[1] https://github.com/Uthmanhere/EE222/blob/master/00_lab1Toggle.asm

5 Lab Tasks

5.1 Simulation

Change the value of `R20` in code to `0xFF` in line `14`. Add the breakpoint to instruction on line `9`. Initiate the simulation, observe:

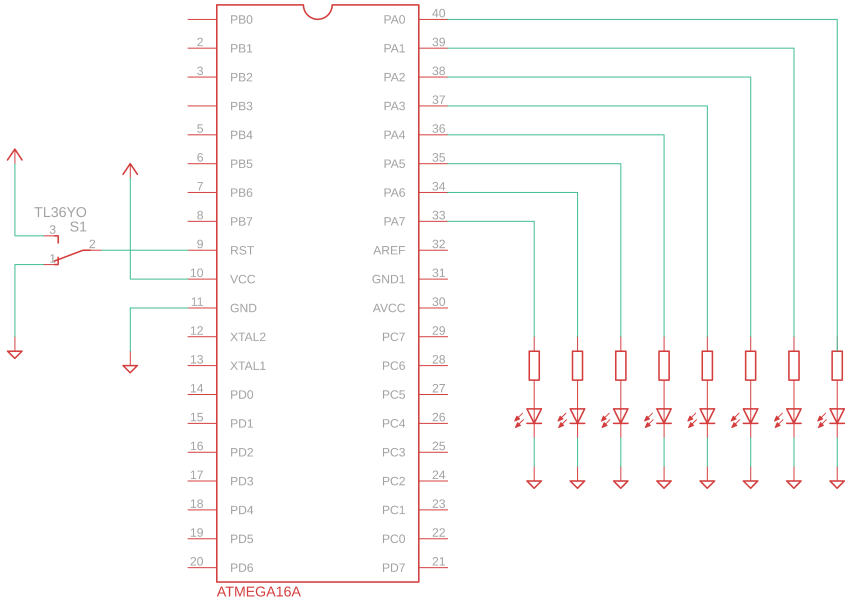
- Status Register
- Time taken in iteration
- Cycles for iteration
- Register R16
- I/O Port A

and fill the table for first **20 iterations**. The table to be filled is available in doc[2] and pdf[3] format.

`Note: Time for each iteration can be calculated from Stop Watch time and Cycle Counter`

5.2 Hardware Implementation

Patch the circuit as schematic suggests. You may utilize the resources available in the trainer kits of lab.



Burn the hex file generated in the above project using the universal programmer.

Task Observe changes and write them in submission. Time any two reasonable LEDs from this lab and previous lab. Submit these readings and defend your observations through them.

[2] https://github.com/Uthmanhere/EE222/blob/master/00_iter_table.docx
[3] https://github.com/Uthmanhere/EE222/blob/master/00_iter_table.pdf