Lab 2: Verilog and FPGA Refresher

Objectives:

The objective of this lab is to review some Verilog and FPGA basics and to familiarize you with Vivado Design Suite and the Nexys A7 Trainer Board.

This lab is organized as follows:

1. Introduction
2. Experiments: **to be conducted on the Nexys A7 using Vivado (should be conducted during the lab session and shown to the instructor)**
    a. **Experiment 1: An 8-bit ripple carry adder (RCA)**
    b. **Experiment 2: Seven Segment Display Negative Number**
    c. **Experiment 3: Addition of two signed numbers**
3. Deliverables: **Online Submission regulations.**

Introduction

An adder is a digital circuit that performs addition of numbers. Adders are not only used in ALUs (Arithmetic Logic Units) in PCs but are also used in address calculation, Program counters increments, Stack pointers increments etc.

There are many types of adders but our focus will be on the ripple carry adder-subtractor (RCA) that works on binary representation of the numbers. Then we will extend our experiment to handle signed numbers. Negative numbers are represented as two's complement and the most significant bit shows the sign (1 negative, 0 positive)

Experiments:

Note: Students should be working in pairs to implement the following experiments

Experiment 1: An 8-bit ripple carry adder (RCA), Use module parameter to make it N-bit

a. **Create a new RTL project targeting the Nexys A7 board using Vivado.**
b. **Write a new Full-adder module in Verilog**
c. **Design and implement an 8-bit RCA using full adders and generate blocks. The 16 switches of the Nexys A7 board should be used to provide two 8-bit inputs to the adder. The 9-bit result (including carry out) should be displayed on one of the two 4-digit seven segment display using the driver module you implemented in last Lab.**
d. **Create the necessary constraint file.**
e. **Program the Nexys A7 board and test it.**
f. **Record both the utilization and the delay of the critical path. You can find the delay in the timing report after you have completed the implementation step.**

Note: You need to show your instructor the result of this experiment within Lab.

Experiment 2: Seven Segment Display Negative Number

Last lab, we implemented Four Digit Seven Segment. This Lab we will modify Four Digit Seven Segment to be (1 Sign + 3 Digit) Seven Segment.

a. **Create a new RTL project targeting the Nexys A7 board using Vivado.**
b. **Design and implement a (1 Sign + 3 Digit) 7-segment display driver module controlled by the 8 input switches. The Nexys A7 has 16 switches, but we choose to use only8 switches to specify the binary value we want to visualize on the 7-segment display. (what is the maximum and min number that could be displayed)**
    a. **Copy files from Lab1 (Experiment#3) don't modify on the same files.**
    b. **A positive number should show no sign.**
    c. **A negative number will show a negative sign (which segment should be on?)**
    d. **Eighth bit of the input indicates its sign (1 negative, 0 positive)**
    e. **Negative numbers are provided in two's complement format. i. (ex. -7 1111 1001, 7 0000 0111)**
    f. **what should be modified to handle negative numbers?**
c. **Create a constraint file attaching:**
    a. **The inputs of the module to switches 0 to 7 (eight bits only)**
    b. **The outputs of the module to the cathodes and anodes of the 7-segment LEDs.**
    c. **The 100 MHz clock input to pin E3**
d. **Program the Nexys A7 board and test it**
e. **Record the utilization (number of LUTs, FFs, and IO ports) consumed by the driver circuit you designed. You can find this information after you have completed the implementation step in the utilization report.**

Experiment 3: Addition of two signed numbers

a. **Create a new RTL project targeting the Nexys A7 board using Vivado.**
b. **Use Modules From Experiment#1 and Experiment#2 (Copy them, don't override).**
c. **Using the designed 8-bit RCA From Experiment#1, modify it to handle 8-bit signed numbers. Negative numbers are represented as two's complement and the most significant bit shows the sign (1 negative, 0 positive).**
d. **The 8-bit (without carry) result should be displayed on one of the two 4-digit seven segment display using the driver module you implemented in Experiment#2.**
e. **Create the necessary constraint file.**
f. **Program the Nexys A7 board and test it.**
g. **Record both the utilization and the delay of the critical path. You can find the delay in the timing report after you have completed the implementation step.**

Deliverables:

General Online Submitting Notes:

For Verilog descriptions or constraint files, please attach just the files (not the whole project). No photos or screenshots accepted.

The submission should be a zipped folder with the following Structure

a. **YourID_Report.pdf**
b. **Exp1 : Folder containing your codes for experiment#1**
c. **Exp2 : Folder containing your codes for experiment#2**
d. **Exp3 : Folder containing your codes for experiment#3**

Lab2 Report Requirements

Report of Lab2 should include:

1. **[0 pts] Your name(s), student(s) ID, Team Name.**
2. **[2 pts] A technical summary of experiments conducted in the lab (Steps, Results, components (a screen shot with Schematic design might help), code functionality, etc...]**
3. **[4 pts] Any Verilog descriptions or constraint files you wrote for the lab.**
4. **[2 pts] Compare the utilization and delay of experiment 3 vs experiment 1, please add your comments about the results.**
5. **[2 pts] Results recorded in the last step of experiment 1 & 2 & 3 (Photos or screen shots)**

Figures:

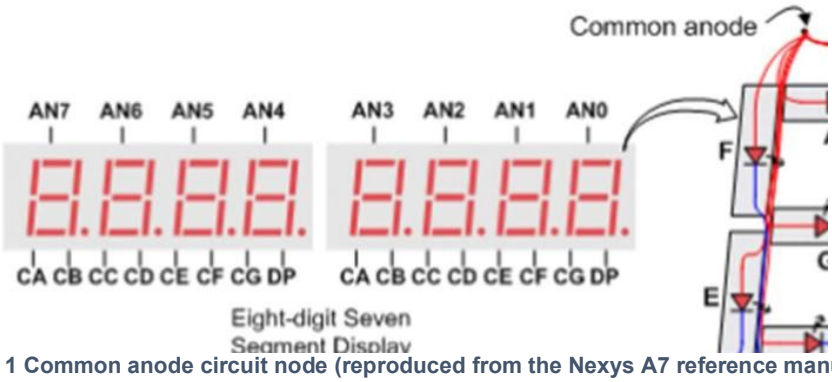

Figure 1 Common anode circuit node (reproduced from the Nexys A7 reference manual)

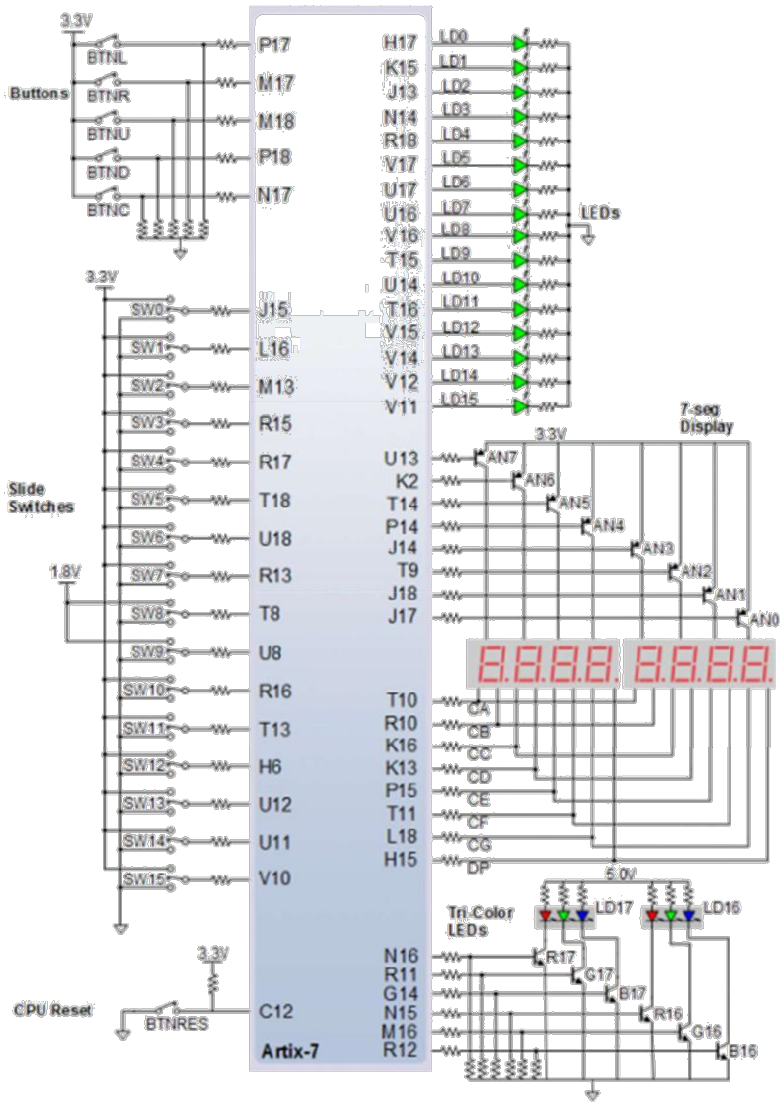

Figure 2 General purpose I/O devices on the Nexys A7 (reproduced from the Nexys A7 reference manual)