University of California, Davis
Department of Electrical and Computer Engineering

EEC180B DIGITAL SYSTEMS II Spring Quarter 2014

LAB 2: Combinational Logic and Arithmetic Circuits in Verilog

Objective: The purpose of this lab is to use Verilog to describe combinational circuits and to design some basic arithmetic circuits.

I. Hex to 7-segment Display Code Converter

Figure 1 shows a hex to 7-segment display code converter. This code converter should be used to display hexadecimal characters (0-9, A-F) on the 7-segment display. The Altera DE2 board has eight seven-segment displays, named HEX0, HEX1,..., HEX7. Each segment is illuminated by driving its control signal to logic 0; that is, they are ***active-low***. In the *DE2_pin_assignments.csv* file, the segments are identified by the indices 0 to 6 as shown in Figure 1. Thus, you should declare the 7-bit port

output [0:6] HEX0, HEX1, …, HEX7;

in your Verilog code so that the output names match the corresponding names in the pin assignments file.

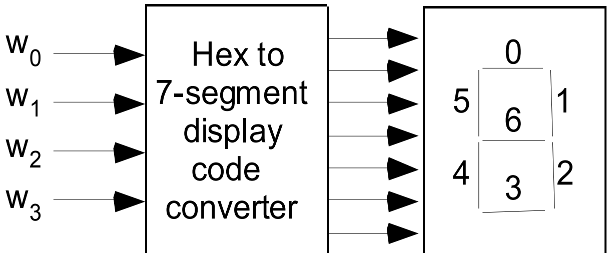

Figure 1. Code converter Block Diagram

Perform the following steps:

1. Create a Verilog module for the 7-segment code converter. Instantiate your code converter in a top-level module in order to drive the 7-segment displays on the DE2 board. Let the hex values on switches SW_{15-12}, SW_{11-8}, SW_{7-4} and SW_{3-0} be displayed on HEX3, HEX2, HEX1 and HEX0, respectively. The segments on HEX7 – HEX4 should be turned off.

2. Perform a functional simulation of your Verilog design to verify your design's correctness. Print a waveform showing the code converter outputs for each of the 16 possible inputs.

3. Compile your design in Quartus II and download the circuit to the DE2 board. Test your circuit using the toggle switches on the DE2 board.

II. **Ripple-carry Adder**

A *full adder* (FA) has inputs a, b and c_i (carry in) and produces outputs s (sum) and c_o (carry out). An n-bit ripple-carry adder can be designed by connecting full-adders in a chain, as shown in Figure 2.

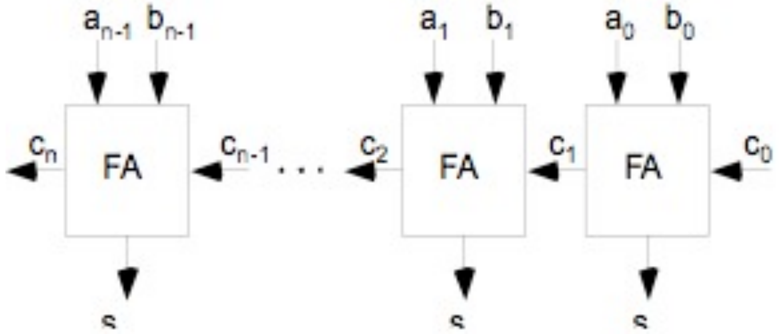

Figure 2. N-bit Ripple-carry Adder Block Diagram

Perform the following steps:

1. Write a behavioral model for a full adder in Verilog
2. Instantiate your full adder subcircuit in order to build an 8-bit ripple-carry adder. Add logic to produce an *Overflow* output, which should be set to 1 whenever the sum produced by the adder does not provide the correct signed (twos complement) value.

3. Perform a functional simulation of your Verilog design to verify your design’s correctness. Print a waveform showing your simulation results.

4. Using the Quartus II Classic Timing Analyzer, perform a timing analysis and determine the critical path in the ripple-carry adder.

Figure 3 shows the traditional procedure for performing the multiplication P=A x B, where A and B are 4-bit unsigned binary numbers. Since each bit in B is either 1 or 0, the summands are either shifted versions of A or 0000. The Boolean AND operation can be used to multiply any two binary bits. Figure 4 shows an array multiplier circuit that implements P = A x B, where A and B are 4-bit unsigned binary numbers.

				a_3	a_2	a_1	a_0
			x	b_3	b_2	b_1	b_0
				a_3b_0	a_2b_0	a_1b_0	a_0b_0
			a_3b_1	a_2b_1	a_1b_1	a_0b_1	
		a_3b_2	a_2b_2	a_1b_2	a_0b_2		
	a_3b_3	a_2b_3	a_1b_3	a_0b_3			
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Figure 3. Multiplication of Unsigned Binary Numbers

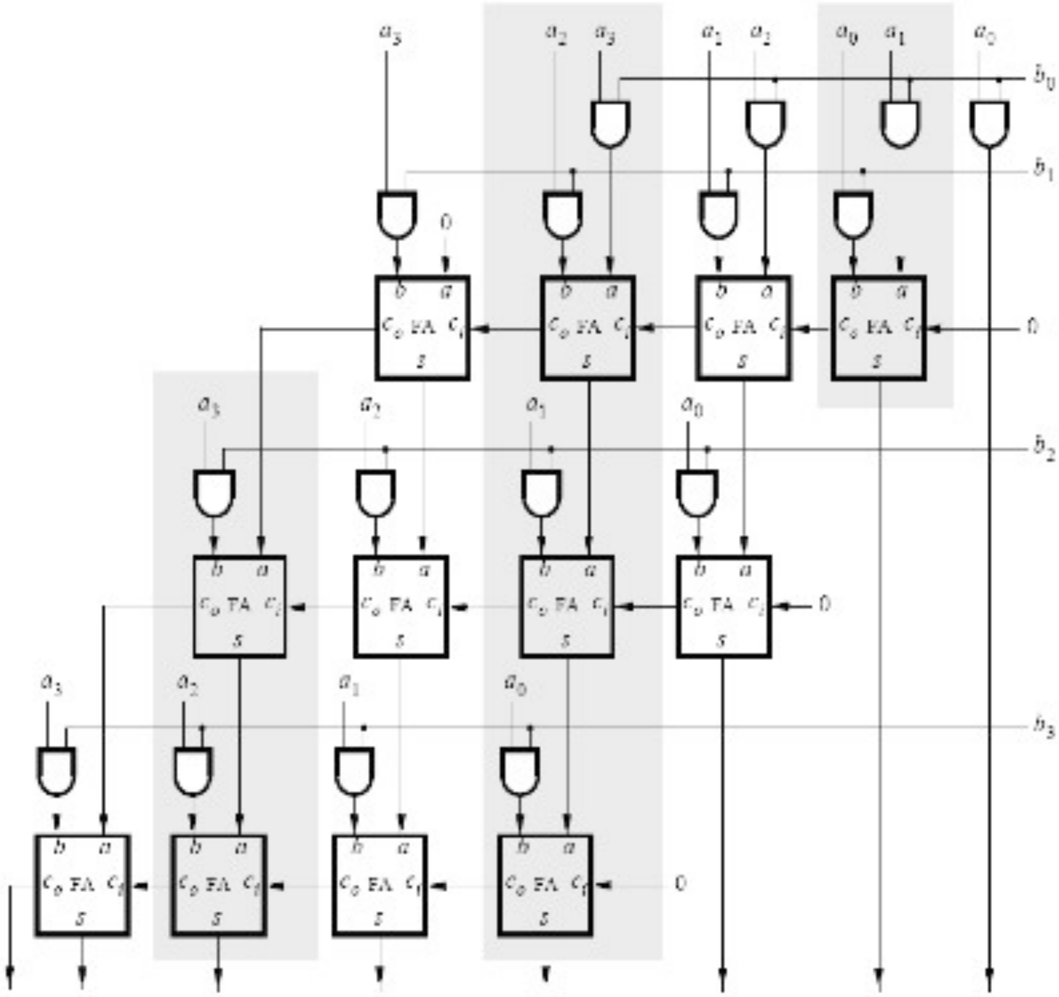

Figure 4. Array Multiplier Circuit Block Diagram

Perform the following steps:

1. Create a structural model in Verilog that describes an 8x8 unsigned array multiplier that can be implemented on the Altera DE2 board. Use switches SW_{15-8} to represent the number A and switches SW_{7-0} to represent the number B. Display the hex value of A on HEX7-6 and the hex value of B on HEX5-4. Display the result P = A x B on HEX3-0.

2. Perform a functional simulation of your design and print the simulation results.

3. Compile your design in Quartus. Download your design to the Altera DE2 board and test your circuit. **Demonstrate your circuit to your TA.**

4. Run the Quartus Classic Timing Analyzer to determine the critical path of your multiplier.

III. **Lab Report**

For your lab report, include the following:

For your lab report, include the following:

- Lab Cover Sheet with signed TA verification for successful simulations and implementation in the Altera board.
- Simulation waveforms of your functional simulations.
- Complete Verilog source code for your design and test bench for each part.
- Timing analysis for Parts II and III.

Grading Guidelines

Part 1 – 25 points
Part 2 – 25 points
Part 3 – 50 points

You are supposed to work alone in this lab. Any unauthorized collaboration will be reported to SJA.