22583 Special Topics in Computer Engineering (2)
Advanced Digital Systems Design and Verification

Lab 1: Build Adders, Subtractors and Multipliers

February 26, 2024

1 Objectives

- Use structural description in Verilog
- Test Arithmetic Circuits in Verilog
- Experiment with the usage of instantiation and tasks

2 Tools

Use one of the following Verilog simulator (or any one of your choice):

- Intel® Quartus®
- www.edaplayground.com
- Download Verilog from: http://bleyer.org/icarus/

3 Design Activity

3.1 Part 1

You need to design and test a circuit that repeatedly adds an input A to itself as shown in Figure 1. This circuit is often called an accumulator . The circuit includes a carry out from the adder, as well as an overflow output signal. If the input A is considered as a 2's-complement number, then overflow should be set to 1 in the case where the output sum produced does not represent a correct 2's-complement result. Note that you have to create a structural (**non-behavioral**) implementation for this lab.

3.2 Part 2

Extend the circuit from Part 1 to be able to both add and subtract numbers. To do so, introduce an add_sub input to your circuit. When add_sub is 1, your circuit should subtract A from S, and when add_sub is 0 your circuit should add A to S as in Part 1.

3.3 Part 3

The circuit shown in Figure 2 can be used to multiply two 4 bits numbers. Use this approach to implement an 8 x 8 multiplier circuit with registered inputs and outputs, as shown in Figure 3.

4 Design Flow

Strictly follow the following order:

1. Write the Verilog code for the circuits described in Parts 1-3 above. Using the Structural Description: Create a separate file for each module.
2. Write Verilog testbenches to test the three circuits.

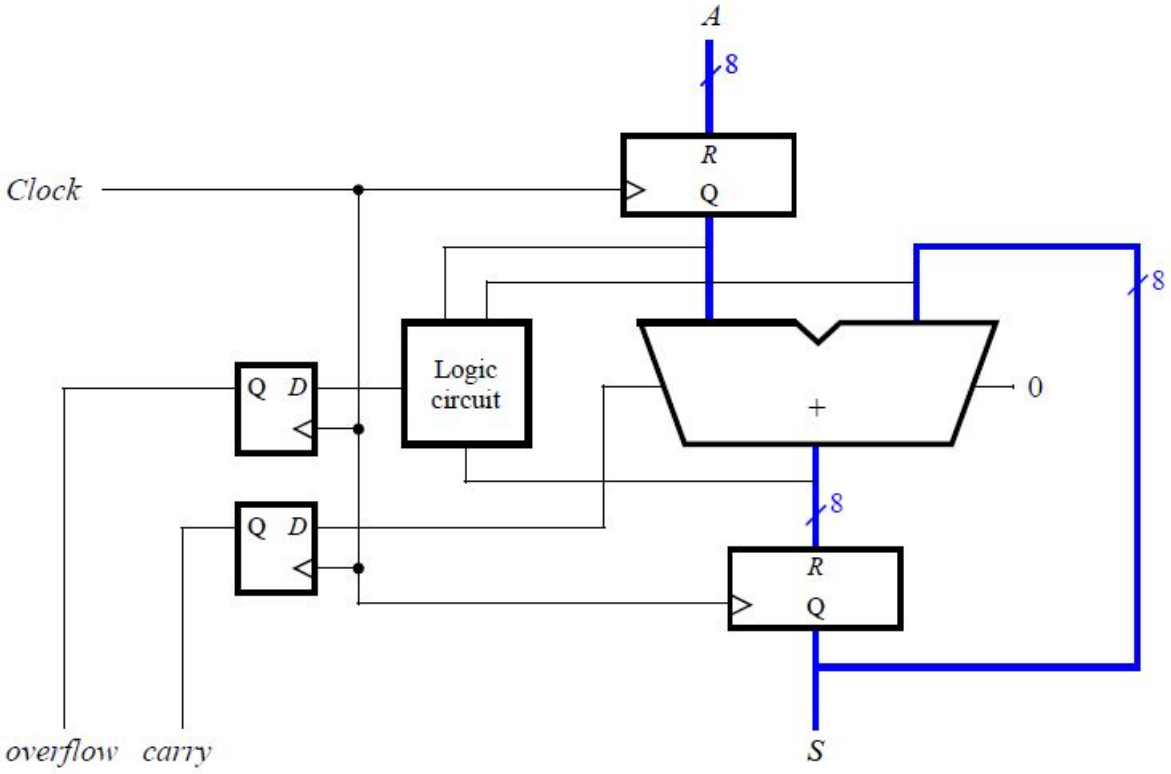

Figure 1: An eight-bit accumulator circuit

3. Perform Functional Simulation (Run Simulation → Run Behavioral Simulation) and take snapshots of the results.

5 Deliverables

You will need to submit to E-learning (as a .zip file) including the following:

- The Verilog code: Your design should be modular.
- Verilog testbenches: test each module separately.
- Run logs: the run result for each test case (text and waveforms)

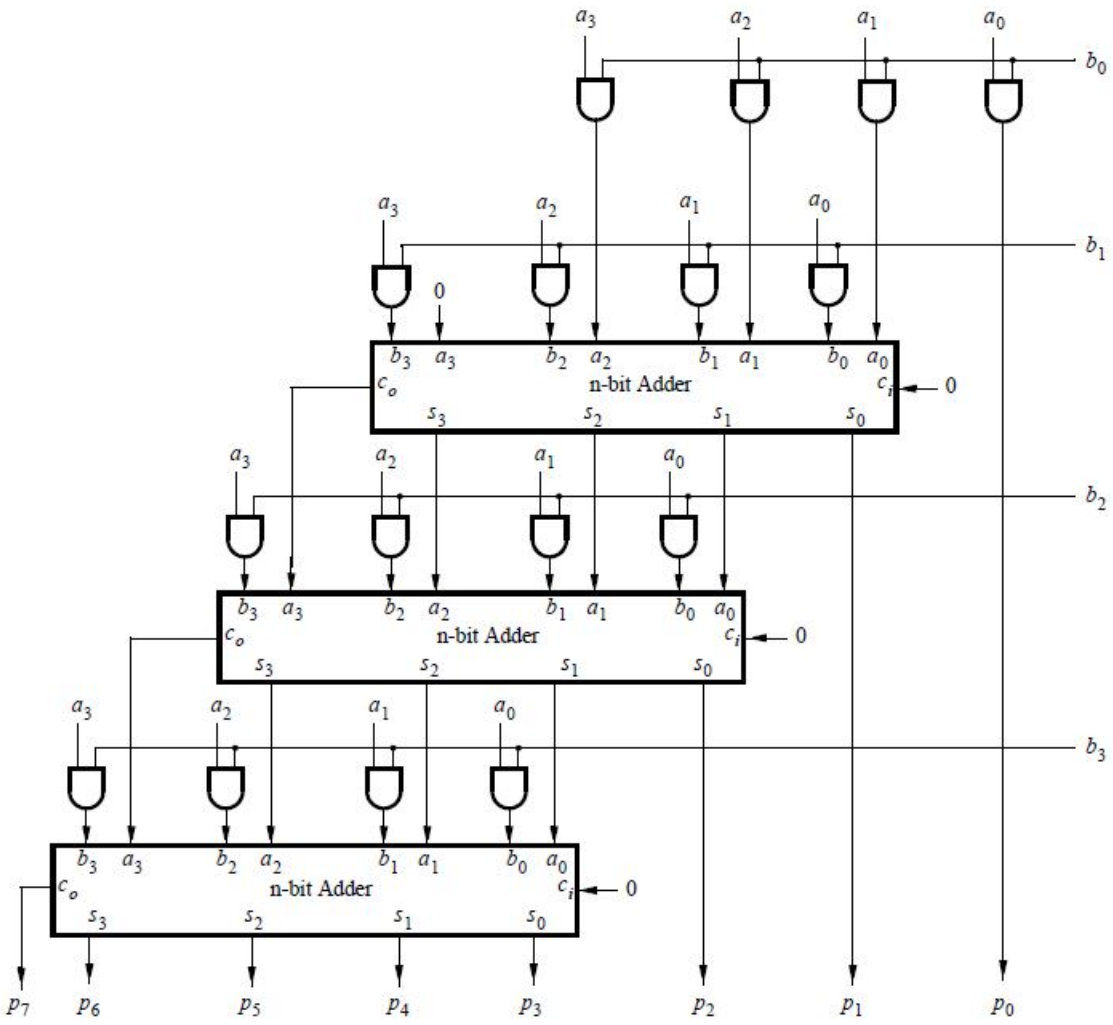

Figure 2: An array multiplier implemented using n-bit adders

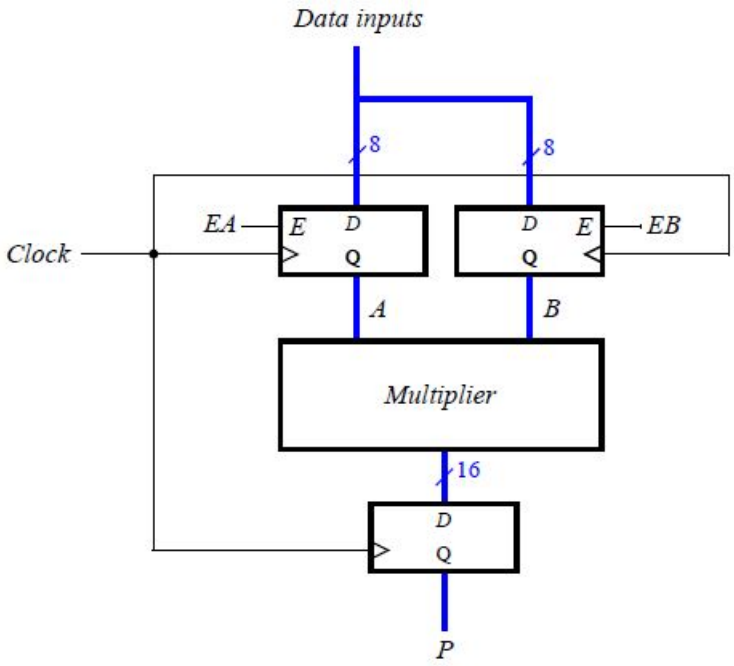

Figure 3: A registered multiplier circuit..