

CSE 3201 F13: Laboratory Exercise 5

Note these exercises are excerpted from "Laboratory Exercise 6, Adders, Subtractors, and Multipliers" from Altera Corporation. Sections have been removed and shorted as required. For the original file, please see

ftp://ftp.altera.com/up/pub/Altera_Material/12.0/Laboratory_Exercises/Digital_Logic/DE2-115/verilog/lab6_Verilog.pdf

Multipliers

The purpose of this exercise is to examine arithmetic circuits that multiply numbers. Each circuit will be described in Verilog and implemented on an Altera DE2-series board.

Figure 1a gives an example of paper-and-pencil multiplication $P = A \times B$, where $A = 11$ and $B = 12$.

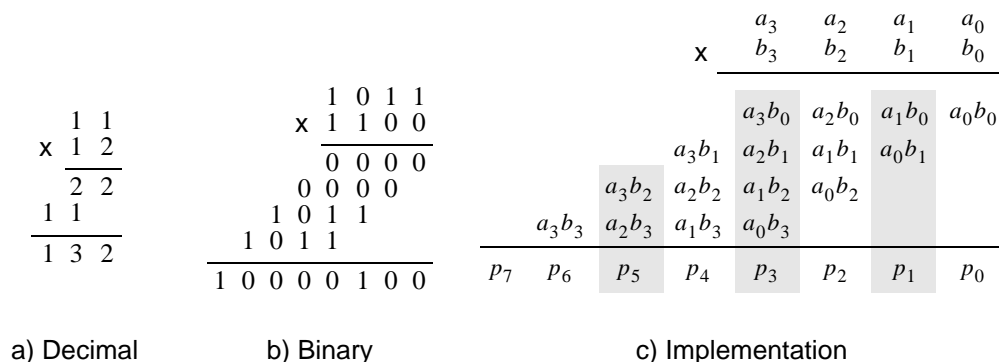


Figure 1: Multiplication of binary numbers.

We compute $P = A \times B$ as an addition of summands. The first summand is equal to A times the ones digit of B . The second summand is A times the tens digit of B , shifted one position to the left. We add the two summands to form the product $P = 132$.

Part *b* of the figure shows the same example using four-bit binary numbers. To compute $P = A \times B$, we first form summands by multiplying A by each digit of B . Since each digit of B is either 1 or 0, the summands are either shifted versions of A or 0000. Figure 1c shows how each summand can be formed by using the Boolean AND operation of A with the appropriate bit of B .

A four-bit circuit that implements $P = A \times B$ is illustrated in Figure 2. Because of its regular structure, this type of multiplier circuit is called an *array multiplier*. The shaded areas correspond to the shaded columns in Figure 1c. In each row of the multiplier AND gates are used to produce the summands, and full adder modules are used to generate the required sums.

Part A

In Figure 2, an array multiplier was implemented using full adder modules. At a higher level, a row of full adders functions as an n -bit adder and the array multiplier circuit can be represented as shown in Figure 3.

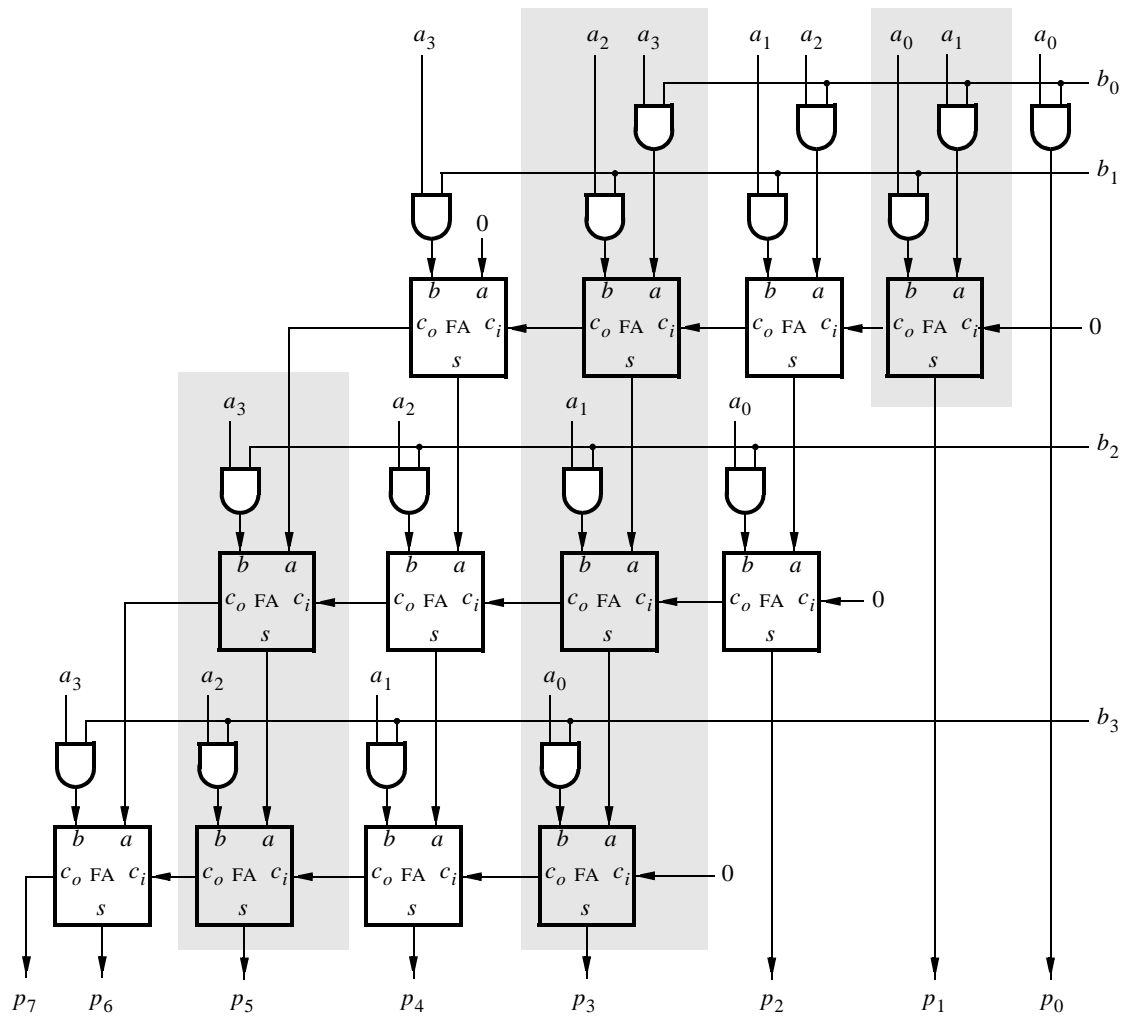


Figure 2: An array multiplier circuit.

Each n -bit adder adds a shifted version of A for a given row and the partial sum of the row above. Abstracting the multiplier circuit as a sequence of additions allows us to build larger multipliers. The multiplier should consist of n -bit adders arranged in a structure shown in Figure 3. Use this approach to implement an 8×8 multiplier circuit.

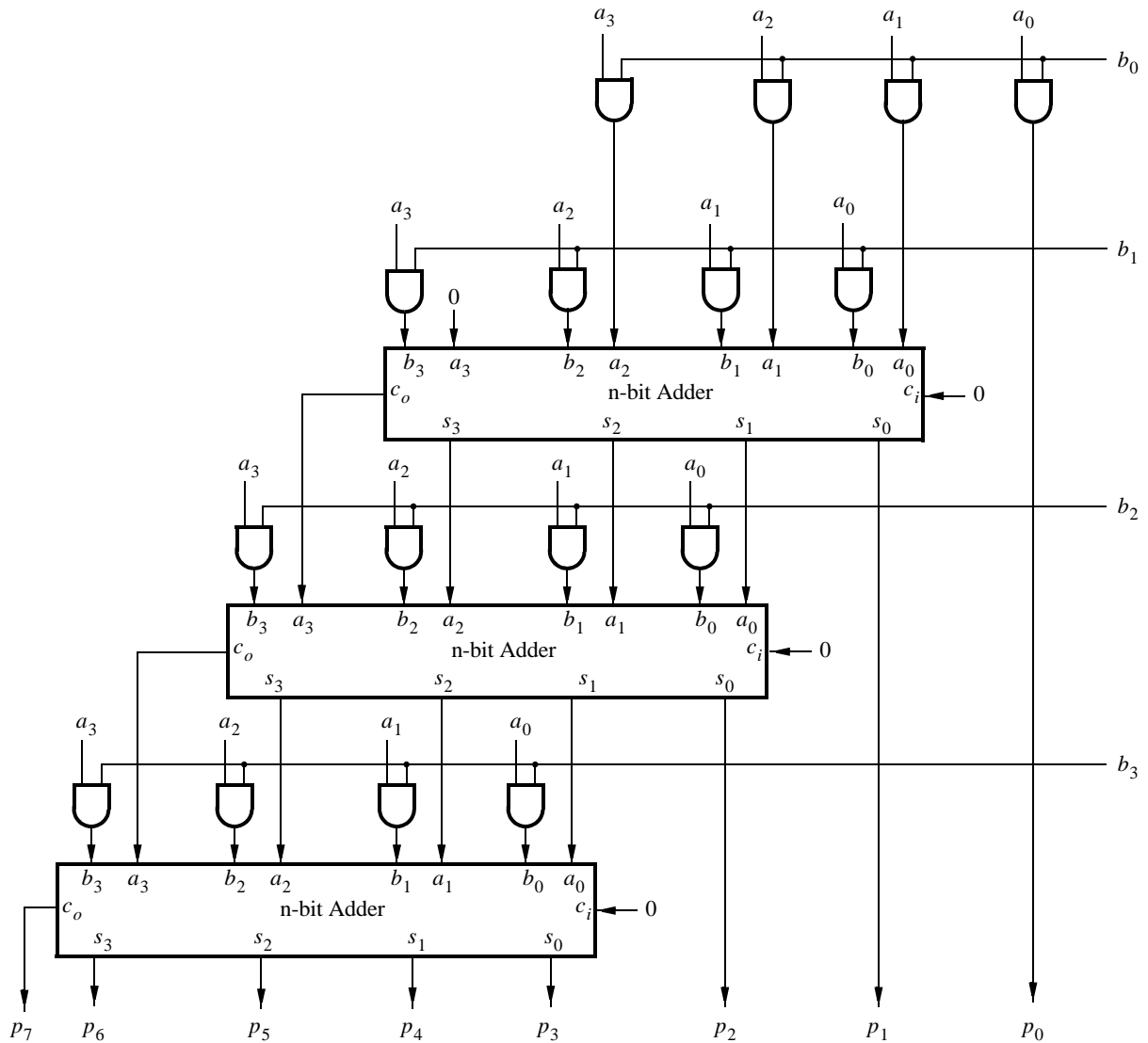


Figure 3: An array multiplier implemented using n -bit adders.

Perform the following steps:

1. Create a new Quartus II project.
2. Write the required Verilog file, include it in your project, and compile the circuit.
3. Use functional simulation to verify your design.
4. Augment your design to use switches SW_{15-8} to represent the number A and switches SW_{7-0} to represent B . The hexadecimal values of A and B are to be displayed on the 7-segment displays $HEX7-6$ and $HEX5-4$, respectively. The result $P = A \times B$ is to be displayed on $HEX3-0$.
5. Assign the pins on the FPGA to connect to the switches and 7-segment displays.
6. Recompile the circuit and download it into the FPGA chip.
7. Test the functionality of your design by toggling the switches and observing the 7-segment displays.
8. How large is the circuit in terms of the number of logic elements?

9. What is the longest delay from input to output (latency) or this circuit?

Part B

Part A showed how to implement multiplication $A \times B$ as a sequence of additions, by accumulating the shifted versions of A one row at a time. Another way to implement this circuit is to perform addition using an adder tree.

An adder tree is a method of adding several numbers together in a parallel fashion. This idea is illustrated in Figure 4. In the figure, numbers $A, B, C, D, E, F, G,$ and H are added together in parallel. The addition $A + B$ happens simultaneously with $C + D, E + F$ and $G + H$. The result of these operations are then added in parallel again, until the final sum P is computed.

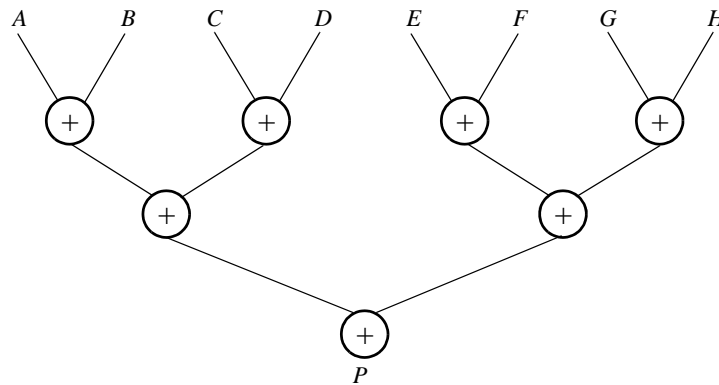


Figure 4: An example of adding 8 numbers using an adder tree.

In this part you are to implement an 8×8 array multiplier that computes $P = A \times B$. Use an adder tree structure to implement operations shown in Figure 3. What is the longest delay from input to output (latency) for this circuit?

Pre-Lab

Before entering the lab ensure that for each design you have at a minimum: 1) Truth tables, maps, Boolean expressions and other design aids. 2) Fully documented Verilog source and 3) Test patterns and/or a testing strategy

If you are not prepared for the lab you will not be allowed to start. The two-hour lab time slots are strictly enforced and you must be prepared in order to complete the lab in the allotted time.

Evaluation

Lab demonstration, in-lab explanations and answers, debug and test approach

Copyright ©2011 Altera Corporation, modified R. Allison October, 2012, modified again by S. Magierowski, Oct. 2013.