Experiment 7. Multiplier Design

Masud ul Hasan · Alauddin Amin · Ahmad Khayyat – Version 151, 12 October 2015

Table of Contents

1. Objectives
2. Materials Required
3. Background
    - 3.1. Approach
    - 3.2. Inputs
    - 3.3. Outputs
    - 3.4. Multiplier Circuit
    - 3.5. Buses and Bus Taps
    - 3.6. Using the Seven-Segment Display
4. Tasks
    - 4.1. Planning Your Implementation
    - 4.2. Testing the Circuit
5. Grading Sheet

1. Objectives

- Learn more about sequential circuits.
- Find the proper library components for the design of a simple 4x4 multiplier circuit.
- Implement and verify the operation of the multiplier circuit.
- Learn how to use buses and bus taps.
- Learn how to use a seven-segment display.

2. Materials Required

- An FPGA prototyping board.
- Design and simulation software tools.
- A seven-segment display interface module.

3. Background

In this experiment, we would like to design and build a circuit that performs the computation:

$$P = A \times x$$

3.1. Approach

P is computed by adding A to itself x times. For example, if $A = 4$ and $x = 3$, then P is computed by adding 4 to itself 3 times, i.e. $P = 4 + 4 + 4$. Alternatively, if $A = 3$ and $x = 4$, then P is computed by adding 3 to itself 4 times, i.e. $P = 3 + 3$
$3 + 3$. Basically, P is computed according to the following pseudocode:

```
P <- 0
```

```
while x ≠ 0
do {
    P <- P + A
    x <- x - 1
}
done <- 1
return P
```

3.2. Inputs

x The first operand (multiplier); a 4-bit unsigned number.

A The second operand (multiplicand); a 4-bit unsigned number.

start Start signal; initializes the circuit:

1. $P \leftarrow 0$
2. Prepares for loading the initial values of *A* and *x* into the appropriate circuit components.

clk Clock signal connected to a debounced push button. Each time the push button is pressed, a *clk* pulse is produced, decrementing the counter by 1 and storing a new value ($P + A$) into *P*.

3.3. Outputs

P The resulting product: $P = x * A$.

done Indicates the completion of the computation.

3.4. Multiplier Circuit

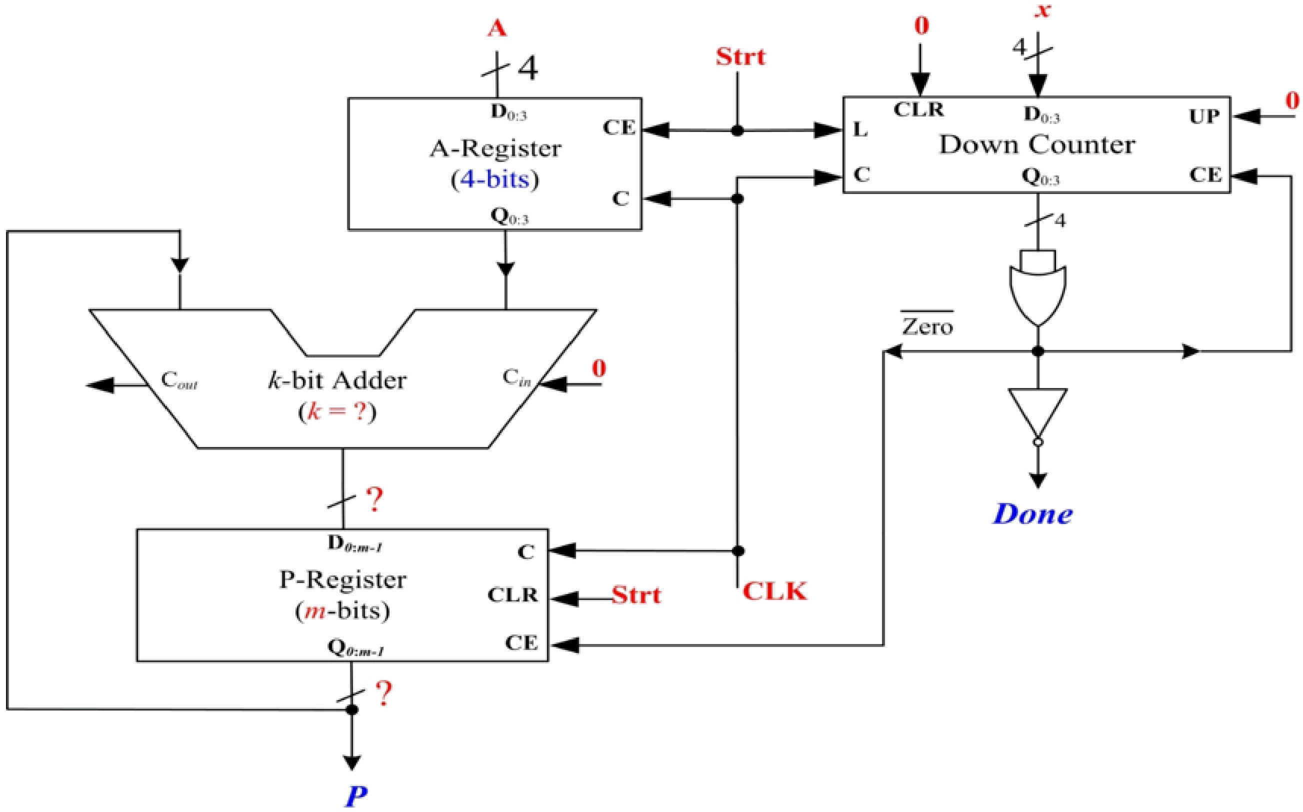

Figure 1. The 4x4 Multiplier Circuit

The multiplier circuit is shown in The 4x4 Multiplier Circuit figure. It consists of the following:

***A*-register**

A 4-bit register which holds the multiplicand (*A*).

***P*-register**

An *m*-bit register (you need to specify the value of *m*). The final product, *P*, is accumulated in this register.

Counter *C*

A 4-bit down counter used to control the number of additions to be performed. The counter counts from an initial value of *x* down to 0. The operation is stopped when the count reaches 0. This zero condition (*zero*) is detected using a 4-input OR gate.

Adder

A *k*-bit adder (you need to specify the value of *k*)

3.5. Buses and Bus Taps

A group of two or more signals that carry closely related values is considered a *bus*. A bus is usually used to wire multiple bits of signals. Buses are used to add clarity to a schematic drawing, and to conveniently name multiple wires.

> *Creating a Bus in Xilinx Schematic Entry*
>
> In Xilinx schematic design entry tools, a bus can be created by simply renaming a wire:
>
> 1. Draw a wire on the schematic where you want the bus to be
> 2. Right-click on the wire and select *Object Properties*. A pop-up window will appear.
> 3. Set the *name* field using the format: `bus_name[MSB:LSB]`, where `MSB` and `LSB` determine how many signals there are in the bus.
>
> For example, the name `product[3:0]` would create a 4-bit bus named *product*. The *most significant bit* (MSB) of this bus would be the signal `product[3]`, and the *least significant bit* (LSB) would be `product[0]`. The bus is represented by a thick line on the schematic drawing.

Bus taps represent *branches* of a bus, which can be used to wire a specific signal from a bus to a destination.

> *Creating Bus Taps in Xilinx Schematic Entry*
>
> 1. Click on the *Add Bus Tap* button on the toolbar to set the cursor to the bus tap mode.
> 2. Click on the bus at the point where you want to add the bus tap.
> 3. Draw a wire from the bus tap.
>
> To select which signal this wire would represent, use the same procedure of Creating a Bus in Xilinx Schematic Entry, while replacing the `[MSB:LSB]` part with the specific signal number in the bus, e.g. `product[2]`.

> In order for the bus taps to work correctly, it must be connected to the bus, and the name of the wire and the name of the bus must match exactly (case sensitive).
>
> For example, if we need bit 2 of the bus `product`, `product2` is a wrong name. The correct name is `product[2]`.

> *Bus taps* is Xilinx-specific terminology.

3.6. Using the Seven-Segment Display

You can use a seven-segment display to show numeric values, such as the result of the multiplication computation.

> *Seven-Segment Interface Module and UCF File*
>
> A seven-segment interface Verilog module is provided for you for use in this experiment, along with the required pin assignments (UCF file).
>
> Study the module’s ports in order to determine the appropriate connections for using it. For example, there are seven outputs of this module that represent the seven inputs of the seven-segment display that control its seven segments.

For the purpose of this experiment:

- Use the seven-segment displays to show the output of the multiplier (the product) in hexadecimal form.
- use the left-most decimal point on the seven-segment display panel to reflect the value of the *done* signal in the multiplier circuit.

4. Tasks

4.1. Planning Your Implementation

1. Determine the sizes of the *P* register and the adder (*m* and *k* in The 4x4 Multiplier Circuit figure).
2. From the Xilinx library, select the proper elements needed to build the above circuit.
3. Determine how you will connect your outputs to the FPGA board indicators, e.g. LEDs or seven-segment display.
4. Determine how you will apply the inputs, e.g. through switches and/or push buttons. Justify your choice of the specific board input device for each input signal.

4.2. Testing the Circuit

1. Determine the proper sequence in which you can apply inputs, and the the input values, in order to test the circuit.

   > For any inputs to take effect, you need to pulse the clock.

2. What happens if you keep pulsing the clock after getting the multiplication result? Record and

explain your observations.

3. Try different values of A and x (including 0 for each of A and x).

5. Grading Sheet

Task	**Points**
Selection of proper components	15
Multiplier circuit demonstration	40
Showing the product on the seven-segment displays	20
Lab notebook and discussion	25

Version 151
Last updated 2015-10-24 00:52:58 AST