Assignment 11

Multiplier

Embedded Logic Design

October 31, 2015

1 Description

1.1 Booth's Multiplier

Implement the Booth's multiplication algorithm using Verilog HDL in which the multiplicand and multiplier are 4 bits wide each. The values for the multiplier and multiplicand are programmed using switches SW7...SW4 and SW3...SW0 respectively. The result is shown using the LD7...LD0. The module must have the following interface:

```
module multiplier(product, multiplicand, multiplier, clock, start, busy);
```

`product` 8 bit value, output

`multiplicand` 4 bit value, input

`multiplier` 4 bit value, input

`clock` 1 bit value, input: Represents the clock (FPGA clock)

`start` 1 bit value, input: If this signal is set, the multiplier and multiplicand values are considered valid and multiplier starts. If the `start` signal is set, while a multiplication is underway, the multiplier restarts with the current values of multiplier and multiplicand.

`busy` 1 bit value, output: As long as the multiplication is underway, this line is asserted. It goes low, after the multiplier finished its execution and is idling.

Use a modeling that you seem fit and remember to use either the top-down or bottom-up approach for the required modules. Also keep in mind that small modules are easier to test for correctness. Document the utilization in terms of flip-flops, latches, IOBs, LUTs and required clock cycles to obtain a result.

Before you start implementing a module, understand how the Booth's multiplication algorithm works. Keep in mind that you have several simple, yet powerful, tools such as cascading, module instantiation, etc, at your disposal resulting in less coding and a cleaner solution.

1.2 Dedicated Multiplier

Using data-flow modeling, implement the same functionality as in section 1.1. However this time, make sure that you utilize the dedicated hardware multiplier integrated on the FPGA. Check that your approach is correct by analyzing the floor plan of your design ("Place & Route" → "Analyze Timing / Floorplan Design (PlanAhead)"). More information about PlanAhead is summarized in its documentation: `http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_4/PlanAhead_Tutorial_Design_Analysis_and_Floorplanning_for_Performance.pdf`

1. How many dedicated multipliers does the FPGA have?
2. Where are they located physically?
3. Which one does your design use? (e.g. show in a screenshot)

What is the resource requirement of your design in terms of flip-flops, latches, IOBs, LUTs now? Are there any differences? How can you explain those differences?

2 Deliverables

All files must be submitted to `nanu.iiitd.edu.in` via `git` or `subversion`. Late submissions are not evaluated nor will be submissions through `https://www.usebackpack.com` or mail. Your repository has to contain:

- Verilog code of all modules and test benches of section 1.1
- Verilog code of all modules and test benches utilizing the dedicated hardware multiplier (refer to section 1.2)
- Create a documentation in PDF (no proprietary format by e.g. Microsoft®) with the following content:
  - The resource utilization of both multiplier implementations and the comparison
  - Answers/Screenshots to the questions above.

2.1 Remarks

If you encounter a problem, ask Google, DuckDuckGo, Bing, etc. first. The TAs will not type the question that you have, into the mask in the search engine for you. Required resources, textbooks, etc. are available on the ELD course website of `https://www.usebackpack.com` or in the Internet (datasheets, AVR library documentation, etc.)