UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital System Design with Hardware Description Language
ECE 846 – Spring 01-02
Homework 1: RTL Design with VHDL
Due Date: Esfand 27

Description:

RTL Processing Element Design.

We have an unsigned 8-bit multiplier (*inA[7:0]*, *inB[7:0]*, *outW[15:0]*) that we want to use for approximating multiplication of two 16-bit operands, *opndA[15:0]* and *opndB[15:0]*. The result is to become available on the 32-bit output, *rsltW[31:0]*. A simple and very inaccuruate way of doing the multiplication is to take eight most significant bits of *opndA* and *opndB*, multiply them using the 8-bit multiplier and use the result as the most significant 16 bits of *rsltW* with 16 zeros to its right.

Alternatively, instead of taking eight bits of *opndA* and *opndB* from the left most bit, we can start with the first 1 from the left of the operands and take the most eight valuable bits to its left. This method ignores several of the right-hand side bits of the operands. If the most valuable bit of an operand is not found in the eight-most significant bits, the right-most bits of that operand will be used as is, and no bits are ignored. We refer to this multiplier as a valuable 16-bit approximate multiplier (VAM16).

For the implementation of this multiplier, you need to identify the first 1 from the left of each operand, if the 1 is in the upper eight bits, use the next eight bits for multiplication, remember the number of ignored bits of each operand, perform multiplication of two eight bit operands, and then add zeros to the right of the result equal to sum of the number of ignored zeros of the two operands.

Design Phase:

A. Design the described approximate multiplier (VAM16) using the Huffman model style. The VAM16 receives its inputs (*opndA[15:0]* and *opndB[15:0]*) via its input 32-bit bus, simultaneously. This happens after VAM16 receives a complete positive pulse on its *start* input signal. Finding the first 1 from the left of each operand must be implemented sequentially. The unsigned 8-bit multiplier is an 8-bit array multiplier. When calculation is complete, the VAM16 issues a *ready* pulse, and puts its 32-bit result on the output bus.
   - Draw the complete datapath.
   - Draw a state diagram that shows the behavior of your controller. In each state, show the control signals that are issued.
   - The unsigned 8-bit array multiplier must be implemented using *generate* statement.

B. Implement your design in VHDL. Your top module must include a datapath and a controller.

C. You are to provide an effective testbench for the design developed, and then test your design with at least 4 sets of data. A *python* code is provided to model the functionality of the approximate multiplier. For justification of your result, you must use this *python* code and compare your result with the *python* result.

Deliverables:

A. All VHDL codes, with appropriate names.

B. A complete report containing answers to all parts of each question. Your report should include enough design illustration, description, actual data, and output justification. Note that your reports should be well-organized.

Attention:

A. Your design must be synthesizable.

B. Your report should include adequate design illustration, description, actual data, and output justification. Note that reports should be written in English, in IEEE format, and well-organized.

C. For drawing schematics, use Visio or other graphic tools.

D. Make sure you do an independent work and what you submit as your final report includes none but your own work.

E. Compress all files and documents mentioned in the Deliverables section into a zip file and upload on elearn. The name of the zip file must be in this format "YourFirstName-YourLastNme-HW#1".