THE UNIVERSITY OF WAIKATO
Department of Computer Science

COMP311 — Advanced Computer Architecture 2008
Assignment 4 - VHDL Multiplier

Lecturer:	Tony McGregor
email:	tonym@cs.waikato.ac.nz
Due:	20/10/2008
Worth:	15%

For this assignment you are going to design and simulate one of the multiplier algorithms described in chapter three of Patterson and Hennessy. You are to represent your design in VHDL and simulate it using Riviera.

A top-level block diagram for the multiplier is shown in Figure 1. The *multiplier* and *multiplicand* are read into the mult8 component using the *multiplier* and *multiplicand* inputs when the *LdData* input is asserted (active high) for one clock cycle. The Multiplier should then start multiplying the two numbers together in the following clock cycle. Once the multiplication is completed the 16-bit *product* is output on the 8-bit `ProductOut` lines over the next two clock cycles. During the first clock cycle the most significant 8 bits of the *product* are output on the `ProductOut` output while `ValidProdHi` is asserted. During the second clock cycle the least significant 8 bits of the *product* are output on the `ProductOut` output while `ValidProdLo` is asserted. A timing diagram showing this behaviour is given in Figure 2.

Your Multiplier design should be able to handle 2's compliment negative numbers. That is if an input (*multiplier* or *multiplicand*) is negative then it should be converted to being positive and the sign remembered. Once the product has been generated then it should be made negative if the signs of the *multiplier* and *multiplicand* were different.

The VHDL entity that you are to use for you multiplier is given below.

```
entity mult8 is
  port (
    LdData         : in  std_logic;
    Multiplier     : in  std_logic_vector(7 downto 0);
    Multiplicand   : in  std_logic_vector(7 downto 0);
    ProductOut     : out std_logic_vector(7 downto 0);
    ValidProdHi    : out std_logic;
    ValidProdLo    : out std_logic;
    reset          : in  std_logic;
    clk            : in  std_logic);
end mult8;
```

You should declare all signals in your design of type `std_logic` or `std_logic_vector`. You may use the addition function from `ieee.std_logic_unsigned`.

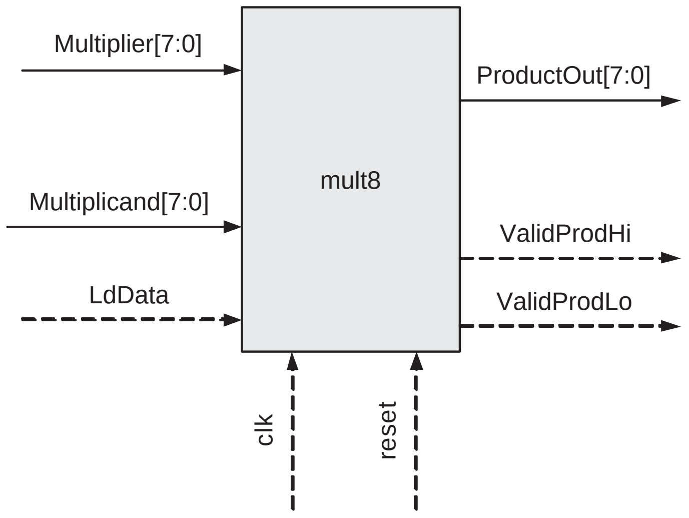

Figure 1: Top-Level Block Diagram for the Multiplier

Test your VHDL design with test-bench provided with this assignment sheet.

Submit your assignment using Moodle. Submit the following:

1. A block diagram showing the structure of you multiplier.
2. A state transition diagram defining the behaviour of the control unit for your multiplier.
3. The VHDL files for you multiplier design. Your VHDL files should be fully documented.
4. Details on how to compile and run simulations of your Multiplier design. You should also describe any assumptions that you made in your interpretation of the specification.

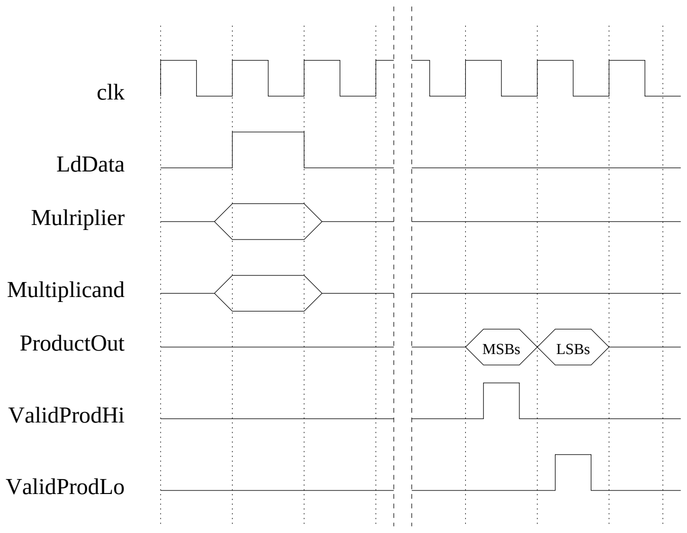

Figure 2: Timing Diagram for Multiplier operation