COE 405, Term 062

Design & Modeling of Digital Systems

HW# 4

Due date: Saturday, May12, 2007

Q.1. It is required to model an **N-bit Serial Multiplier**. The VHDL Entity description of the n-bit multiplier is given below:

```
Entity multiplier is
        Generic (n: Positive := 4);
        Port(clk: IN std_logic; dataready: IN std_logic; A : IN   std_logic_vector (n-1
        downto 0); B : IN      std_logic_vector (n-1 downto 0);
        busy, done: OUT std_logic; result: OUT   std_logic_vector ( 2*n-1 downto
        0));
End multiplier;
```

The circuit receives two n-bit operands when the input *dataready* becomes 1. This causes the multiplication process to begin and the *busy* flag to become active. Using the **add-shift** method, the multiplier takes one or two clock cycles for each bit of the multiplicand. When the process is completed, *done* becomes 1 for one clock cycle and *busy* returns to 0. The circuit receives two operands from its *X* and *Y* inputs and produces the result on its 2n-bit *result* output.

(i) Develop a behavioral model of the n-bit Serial Multiplier. Divide your multiplier into a data-path and control unit and model each one separately using behavioral modeling style.

(ii) Write a test bench for testing the 4-bit Serial Multiplier assuming that the input arguments are read from an input file and that the output will be stored in an output file. Use TEXTIO package for this purpose. Apply the following values for testing the correct operation of a 4-bit Serial Multiplier:

Input A	Input B
5	2
2	2
7	0
0	7
15	10
10	15
15	15
15	1

1	15
15	2
15	4
15	8

The output should be stored in the output file using the following format:

Input A	Input B	Result
5	2	5*2=10

(iii) Synthesize the modeled Serial Multiplier in (i) using Xilinx Project Navigator and report on the total equivalent gate count for design after mapping and the longest delay in the design based on Post-Map static timing report.

(iv) Develop a dataflow model of the n-bit Serial Multiplier. Divide your multiplier into a data-path and control unit and model each one separately using dataflow modeling style.

(v) Verify the correct functionality of your dataflow model using the test bench developed in (ii).