Midterm Exam

ECE 448
Spring 2008

Tuesday Section

(15 points)

Instructions
Zip all your deliverables into an archive <last_name>.zip and submit it through WebCT no later than Tuesday, March 4, 10:00 PM EST.

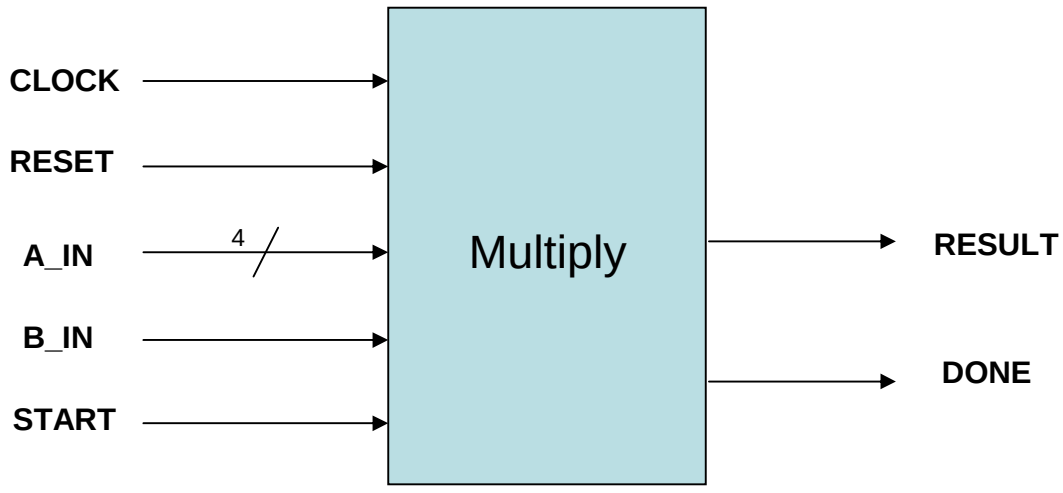
Introduction

Design a system that performs a bit-serial multiplication of two 4-bit unsigned numbers. The algorithm begins when the input START is set to a '1'. When the algorithm is complete, a DONE signal goes high, indicating the completion of the multiplication.

The pseudocode for the multiplication algorithm is shown below:

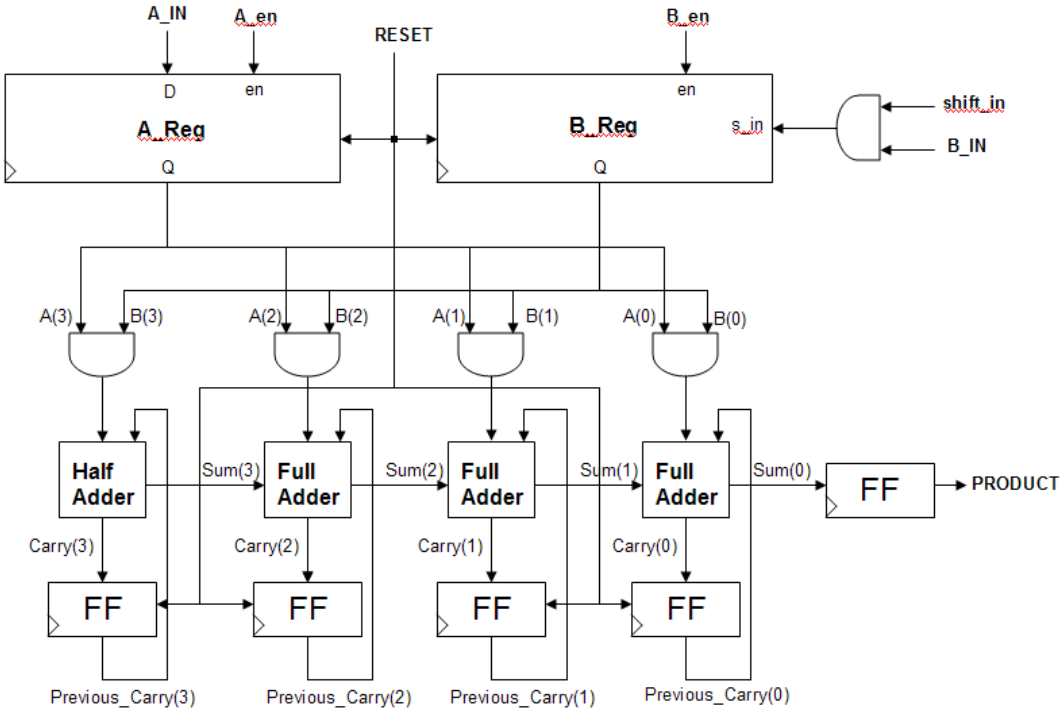
```
A_Reg = A_IN
B_Reg = 0
For i in 0 to 7 loop
        If I < 4
                B_Reg = B_Reg[2:0] & B_IN (i);      -- Least significant bit first.
        Else
                B_Reg = B_Reg [2:0] & '0'
        For j in 3 downto 0 loop
                If j == 3
                        Sum(j) = (A_Reg(j)) and B_Reg(j)) xor Previous_Carry(j)
                        Carry(j) = A_Reg(j) and B_Reg(j) and Previous_Carry(j)
                        Previous_Carry(j) = Carry(j)
                Else
                        Sum(j) = (A_Reg(j) and B_Reg(j)) xor Previous_Carry(j) xor
                                Sum(j+1)
                        Carry(j) = Majority ((A_Reg(j) and B_Reg(j)), Previous_Carry(j),
                                Sum(j+1))
                        Previous_Carry(j) = Carry(j)
                End if;
        End loop;
End loop;
```

The interface circuit is shown below:

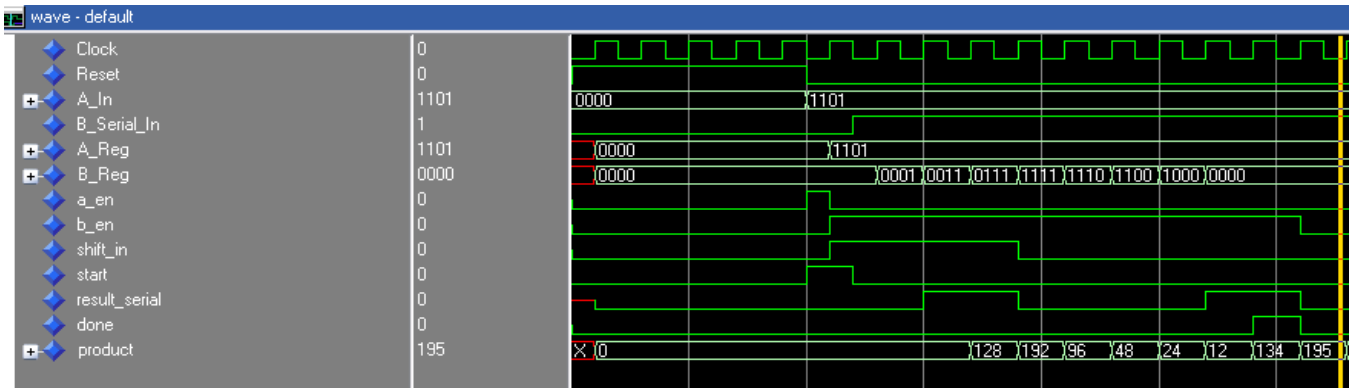


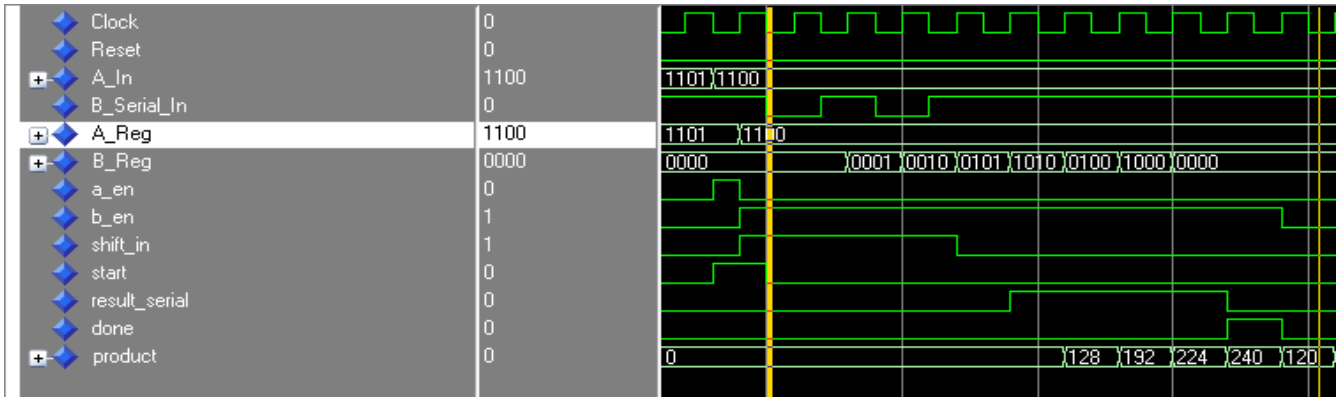
Signal Name	Type	Size	Function
CLOCK	Input	1	20 ns, 50% duty cycle clock
RESET	Input	1	Reset signal
A_IN	Input	4	Multiplicand A, parallel input
B_IN	Input	1	Multiplier B, serial input starting from LSB and ending with MSB
START	Input	1	Starts the computation algorithm, high for one clock cycle
RESULT	Output	1	Serial output of result, starting with LSB and ending with MSB
DONE	Output	1	Signals the end of computation

The execution unit for the circuit is shown below. Note that all inputs that are in capital letters are external inputs. Inputs that are written with lowercase letters are inputs from a control unit.



The timing waveforms for two successive computations are shown below:





Design Requirements

The combinational portion of the circuit should be described using the dataflow VHDL code, and the sequential portion of the circuit should be described using the synthesizable behavioral code. Your code should infer a circuit that requires a minimum amount of FPGA resources.

Tasks

Perform the following tasks:

1. Write a VHDL code of the execution unit of the described above circuit (shown in the block diagram above).
2. Write a testbench verifying the operation of your execution unit.
3. Perform functional simulation of your circuit and use it to debug your VHDL code.
4. Design a control unit of your circuit. If you do not know how to do it, go to Step 6.
5. Write a testbench verifying the operation of your entire code.
6. Synthesize your circuit and save the RTL schematic.
7. Implement your circuit.
8. Perform timing simulations of your circuit.
9. Run the static timing analysis of your circuit.
10. Based on the circuit block diagram and the implementation reports, determine the most critical path in your circuit and its length.

Deliverables

1. VHDL code of your entire circuit fulfilling the requirements specified in the Design Requirements section above
2. VHDL code of your testbenches
3. RTL schematic of your circuit
4. Timing waveforms from the functional simulation demonstrating the correct operation of your circuit.
5. Description of the critical path in your circuit
6. Timing waveforms from the timing simulation demonstrating the delay of the circuit most critical path
7. FPGA resource utilization
8. Minimum clock period of your circuit.