

**ECE 448**  
**Midterm Exam**  
**Wednesday, March 3, 2010**

**Introduction**

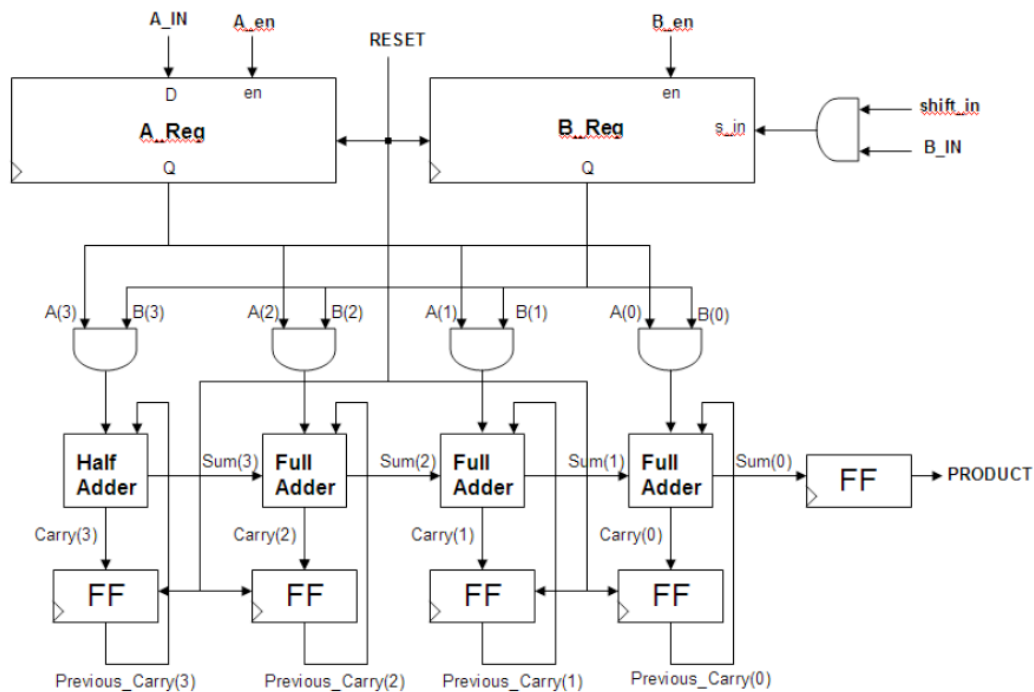
The digital circuit shown in the diagram below is called a bit-serial multiplier. This circuit is capable of multiplying two unsigned 4-bit numbers A and B, and producing the  $PRODUCT=A*B$ .

A is loaded in parallel, four bits at a time, through the input A\_IN.

B is loaded serially, one bit at a time, through the input B\_IN, starting from the least significant bit, B(0), and ending with the most significant bit, B(3), followed by four consecutive zeros.

PRODUCT is generated serially, one bit at a time, during eight consecutive clock cycles, with the least significant bit of the product generated first.

The outputs of all registers and flip-flops are set to zeros at the beginning of the execution.



The meaning of components in the diagram is given below:

A\_reg: 4-bit register with enable and asynchronous reset;

B\_reg: 4-bit shift register with enable, and asynchronous reset, shifting to the left;

FF: D flip-flop with synchronous reset;

Full Adder: one stage of a 4-bit adder;

Half Adder: simplified version of the Full Adder without the carry in input.

The clock signal is not explicitly shown, but it is assumed to be connected to the clock inputs of all sequential components, marked with small triangles.

For the circuit given in the introduction, perform the following tasks:

### Task 1 (50% of points)

Write architecture of this circuit in VHDL using mixed design style, including at least: the dataflow and behavioral design styles. Using the structural design style is optional, but if you use it you need to fully define all lower level entities [at least in terms of their architectures].

Consider applying the for-generate statement whenever appropriate.

### Task 2 (30% of points)

Write a testbench that is able to:

1. initialize all registers and flip-flops to zeros,
2. load register A\_reg with the initial value of the signal X="1010", connected to the input A\_IN,
3. perform multiplication of X by Y="1001", by shifting the least significant bit of Y to the input B\_IN, with shift\_in set to '1' for 4 clock cycles; and then setting shift\_in to '0' for the following 4 clock cycles,
4. collect the PRODUCT bits in an 8-bit signal Z, starting from the least significant bit, and ending with the most significant bit.

Use clock signal operating at the frequency of 100 MHz.

### Task 3 (20% of points)

Based on your knowledge of the internal structure of Spartan 3E FPGAs, and assuming that your circuit is implemented using CLB slices only, determine for each component shown in the diagram:

1. How many logic cells (Logic Cell =  $\frac{1}{2}$  of a CLB slice) are needed to implement this component?
2. Which part of the logic cell (Multipurpose Look-up Table - MLUT, Carry&Control Logic - C&C, or Storage Element - SE) is used to implement this component)?
3. In case an MLUT is used for implementation, please determine its mode of operation (ROM, RAM, or shift register (SR)).

Estimate the total number of CLB slices required to implement the entire circuit.

What other component of the Spartan 3 FPGAs could be used to perform the same operation?

What would be the difference in the operation of these two alternative implementations in terms of:

1. number of clock cycles
2. use of CLB slices
3. dependence between the amount of resources used and the width of operands, k, for  $2 \leq k \leq 16$ .