In class, we have discussed Pipelining. Pipelining is an implementation technique in which multiple instructions are overlapped in execution. Today, pipelining is nearly universal.

A pipeline is a set of data processing elements connected in series, so that the output of one element is the input of the next one. In most of the cases we create a pipeline by dividing a complex operation into simpler operations.

In this lab, we are going to design a 4-bit pipelined adder, and an 8-bit pipelined multiplier as well.

Be sure to include your code, schematics, simulation outputs and summarize your findings. Please follow the detailed lab report guidelines for the class.

**Part 1 Pipeline Adder:**

In lab1, we have designed a 4-bit carry-ripple adder using 4 one-bit full adders. Now, we need a 4-bit pipelined version, while each stage contains only one full adder.

Hint: Use D-flip flop (DFF) to help implementing the pipeline.(For more information, see [http://en.wikipedia.org/wiki/Flip-flop_(electronics)#D_flip-flop](http://en.wikipedia.org/wiki/Flip-flop_(electronics)#D_flip-flop))

Q1: Design a pipelined four-bit adder using VHDL or Verilog.
Q2: Simulate and compare the result with four-bit adder in lab1 by using the same test vectors.

**Part 2 Pipeline multiplier**

A binary multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. It is built using binary adders.

A variety of computer arithmetic techniques can be used to implement a digital multiplier. Most techniques involve computing a set of partial products, and then summing the partial products together. This process is similar to the method taught to primary schoolchildren for conducting long multiplication on base-10 integers, but has been modified here for application to a base-2 (binary) numeral system [1].

Here is an example of unsigned binary multiplication:
Looking at the truth table for multiplying 2 bits, the output of the table matches the output of an AND gate’s truth table. Thus, the multiplication process of 2 bits could be achieved using a single AND gate:

\[
\begin{array}{c|cccc|c|cccc|c|cccc|c|}
 & A3 & A2 & A1 & A0 & \text{Inputs} \\
\hline
x & B3 & B2 & B1 & B0 \\
\hline
+ & B0 \times A3 & B0 \times A2 & B0 \times A1 & B0 \times A0 \\
\hline
& C & B1 \times A3 & B1 \times A2 & B1 \times A1 & B1 \times A0 \\
\hline
+ & B2 \times A3 & B2 \times A2 & B2 \times A1 & B2 \times A0 \\
\hline
& C & B3 \times A3 & B3 \times A2 & B3 \times A1 & B3 \times A0 \\
\hline
\end{array}
\]

Internal Signals

\[
\begin{array}{c|c|c|c|c|}
 & \text{sum} & \text{sum} & \text{sum} \\
\hline
C & \text{sum} & \text{sum} & \text{sum} \\
\hline
Y7 & Y6 & Y5 & Y4 & Y3 & Y2 & Y1 & Y0 & \text{Outputs} \\
\hline
\end{array}
\]

Q3: Design a pipeline 8-bit x 8-bit multiplier using full adders and AND gates. (Hint: Usually we can use nine stages for this multiplier). You may use either VHDL or Verilog.
Q4: Try running 1, 5, 10, 100, 500, and 1000 multiplications. Discuss the speedups.
Q5: Discuss the advantages and the disadvantages of Pipelining.

Due Date: Nov 17th