Lab 7. For week starting on 07/04.

In the class we discussed two methods of making the designs faster. In pipelining, we add additional registers between parts of a design to utilise them at the same time, without disturbing others. One can form segments to determine the location of these registers. The individual partitions are called stages. For a balanced design (i.e. design which has stages requiring same amount of time to complete), the throughput multiplied by number of stages. Latency remains same, though.

Loop unrolling works very similar to unrolling loops in a programming language. Consider a sequential computation using one functional unit again and again to reach the end of the task. Such a task can be made faster by providing increased number of functional units. For example, instead of calculating single bit at each step of division algorithm (considered below) one can generate two bits at the expense of replicating one-bit-calculation circuit. This will increase the throughput by 2 times.

Pipelining increases throughput by utilizing the idle hardware, loop unrolling does so by providing extra hardware.

Preparation: “git pull” again, you should be getting a folder named “pipeline”.

Task 1. The code for 16-bit adder discussed in class is provided. It is made by chaining 4 4-bit adders (therefore, is a ripple adder).
A. How many clock cycles are needed by the adder to finish one addition?
B. The example shows how to insert a single stage of registers to increase the throughput by 2 times. Show this by writing a testbench for the module and tracking the signals.
C. Increase the number of stages to increase the throughput even more.
D. Calculate the number of registers needed in each case for register banks.

Task 2. The restoring division algorithm was discussed in class, it is the classic division algorithm. We would like to calculate quotient (q) and remainder (r) , given x and y (x<y), such that x = q*y + r. X and y are n-bit numbers and q and r are p bit numbers.
A. Run the testbench provided for restoring division and assert the correctness of A. How long does it take to get the result?
B. Draw the circuit diagram corresponding to the code provided (this was drawn in class). Connect two such units to calculate 2 bits of q in one clock cycle. What extra circuitry is needed to implement the algorithm correctly?
C. Write a verilog code for the same and test it.