EHD Final Project

Finding x^n

Name: Ayush Singhania(20171031) and Kadiyala Abhiram (20171011)

1. **Timing Analysis:**

   Worst Slack Negative Slack: 0.696ns

   Worst Hold Slack: 0.037ns

   Worst Pulse Width Slack: 4.52ns

   Maximum Operation Frequency : 90 MHz

2. Utilization Analysis:

   Slice LUTs : 472

   Slice Registers : 651

   LUT as Logic : 412

   LUT as Memory : 60

   LUT Flip Flop : 267

   DSPs : 6

3. Power Analysis:

   Total On Chip Power: 1.676 W

   Dynamic Power : 1.535W

   Static Power : 0.141W

4. Basic Algorithm:

The basic idea is that we split the work using the binary representation of the exponent.

Let's write n in base 2, for example:

$3^{13}=3^{1101}= 3^8 \cdot 3^4 \cdot 3^1$

Since the number n has exactly ⌊log2n⌋+1 digits in base 2, we only need to perform O(logn) multiplications, if we know the powers a1,a2,a4,a8,…,a⌊logn⌋.

So we only need to know a fast way to compute those. Luckily this is very easy, since an element in the sequence is just the square of the previous element.

So to get the final answer for 313, we only need to multiply three of them (skipping 32 because the corresponding bit in n is not set):

$3^{13} = 6561 \cdot 81 \cdot 3 = 1594323$

The final complexity of this algorithm is O(logn): we have to compute logn powers of a, and then have to do at most logn multiplications to get the final answer from them.

5. Steps Followed in Project:

   a. Write and test the Verilog code for exponentiation

   b. Create and package a new Ip for exponentiation

   c. Create a block diagram with Zynq processor and new IP and do the necessary connections

   d. Run Synthesis

   e. Run Implementation

   f. Check timing analysis and adjust the clock frequency

   g. Generate Bitstream

   h. Launch SDK

   i. Program the SDK

   j. Run the program on the ZedBoard and observe results using PuTTY terminal Window