=============================================================

**Lab \_\_8\_\_ Report**

**Title \_\_\_\_\_\_\_\_\_** **I/O and Interface**

**Semester \_\_\_\_\_Fall 2016\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_Dec 10\_\_\_**

**by**

**Name \_\_\_\_\_\_\_Nadim Sarras\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ SID \_\_\_008382608\_\_**

**Name \_\_\_\_\_\_\_\_\_\_Kevin Manan\_\_\_\_\_\_\_\_\_\_\_\_ SID \_008355464**

**Lab Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Performed by (print name)** | **Checked by (print name)** | **Successfully Completed** | **Partially Completed\*** | **Failed or Not Performed\*** |
| Kevin Manan | Nadim Sarras | x |  |  |
| Nadim Sarras | Kevin Manan | x |  |  |

**\* Detailed descriptions must be given in the report.**

**San Jose State University**

**Department of Computer Engineering**

**CMPE 140 Lab Report**

**System on Chip Application of Enhanced Single-cycle MIPS Processor for Factorial Calculations**

**Purpose**

Lab 8 applies the previous lab’s Enhanced Single-cycle Mips Processor and introduced the implementation of the processor to a system on chip design with it’s own dedicated factorial accelerator. There is a difference between having a dedicated module processing the factorial calculation compared to the MIPS processor processing the results. The analysis of this lab will depict such difference and will allow us to decide which method is more efficient dependent of its’application.

**Materials & Software**

1. Vivado 2015.2
2. Verilog HDL
3. Nexys DDR4 FPGA Board
4. MIPs Greencard

**Task**

The tasks were split into 3 major portions: the factorial accelerator module, GPIO, and the top level that implements all the modules. All 3 modules will have their own individual address decoder that determines control signals to allow functionality of the circuit.

*Factorial Accelerator:*

The factorial accelerator for this lab will receive a value and calculate the factorial using the module the team had created in lab 2. This module does the arithmetic for the outputs: Done, Err, and Factorial Value.

*GPIO:*

The GPIO is the method of I/O for our System on Chip device. Inputs and outputs are connected to switches and LED’s respectively.

*Top level:*

The top level is the linking of all of our modules combined. The decoder for this level decides which module to operate and connects the factorial accelerator, GPIO, and the MIPS processor with the peripheral devices together.

**Discussion & Results**

*Schematic:*

The schematic consists of the factorial accelerator, GPIO, and top level that connects the modules together. The schematics below were based off the ones given in lecture, and the only edits made were to adjust our sign extensions done for the multiplexor. Based off the results from the test bench and FPGA portion, we can conclude that the schematic we created works.

*System on Chip:*

The System on Chip’s modules were very simple, however, once connected the team did face many issues. One such issue that still exists in our project would be unnecessary loops prior to the calculation. This will not affect the results, however, this unnecessary recursion requires extra clock cycles to complete. The biggest issues that the team faced conducting the lab would be setting the wires correctly, because when the bits do not align correctly Vivado grounds the wires. The address wire that has different connections between the modules were unfortunately grounding and until the team traced the wires assuring that the modules receive the correct number of bits, the address wires from the MIPS “aluout” output went directly to ground. Once this issue was fixed, there were  fewer issues and the rest of the connections were done smoothly. The test bench shown below in the appendix using the input value 5 matched the output value 120 and displayed success. Ultimately, the team was able to accomplish the System on Chip portion of the lab relatively fast once these issues were fixed.

*FPGA Board:*

The FPGA board was completed once our System on Chip test bench is correct. The 4 switches determines the “n” value that we will use to input the value we want to factorial. The Done signal was connected to the LED and the value of the factorial is outputted to the 8-segment LED’s. The FPGA portion of this lab is completed without any errors and was tested using the values 0, 1, 2, 4, 7. The outputs of these values tested were correct and since the test bench was operationable, the FPGA portion of this lab is successful.

*System on Chip Vs. MIPS Single-Cycle Processor:*

After implementing the System on Chip on both the FPGA and the simulation sources, it is clear that the System on Chip design is much more efficient than the Single-Cycle Processor. The System on Chip design utilizes a factorial accelerator which is able to run the computations of a factorial a lot quicker than the MIPS Single Cycle design alone. In addition, when the MIPS Single-Cycle Processor runs calculations it continuously loads values from the registers where the SoC design does not require the loading of registers as often.

Table 1: Enhanced Single Cycle vs SoC computation time

|  |  |  |
| --- | --- | --- |
|  | Enhanced Single Cycle | I/O and Interface |
| n | Computation Time (ns) | Computation Time (ns) |
| 1 | 400 | 220 |
| 2 | 450 | 240 |
| 3 | 500 | 260 |
| 4 | 560 | 330 |
| 5 | 600 | 445 |
| 6 | 625 | 460 |

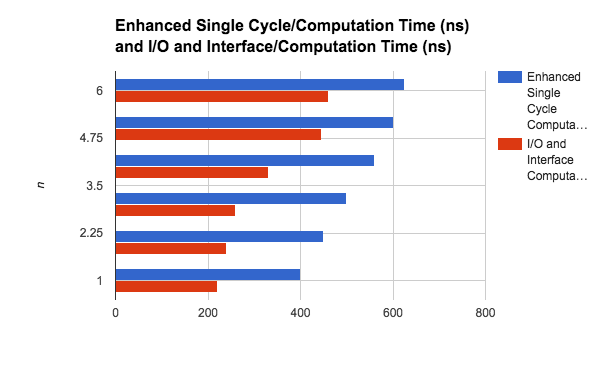


Chart 1: Computation Times comparison

**Conclusion**

Ultimately, the team was able to accomplish the lab without any major errors other than the unnecessary recursion. The recursion is an issues that does not affect the results other than time, so our System on Chip Vs MIPS Single Cycle Processor may hold different results, but the recursion only affects approximately 4 clock cycles which is insignificant, however, may result in 4 cycles slower result. The SoC is a lot faster than the MIPS single cycle processor when calculating factorials values of numbers.

**Appendix**

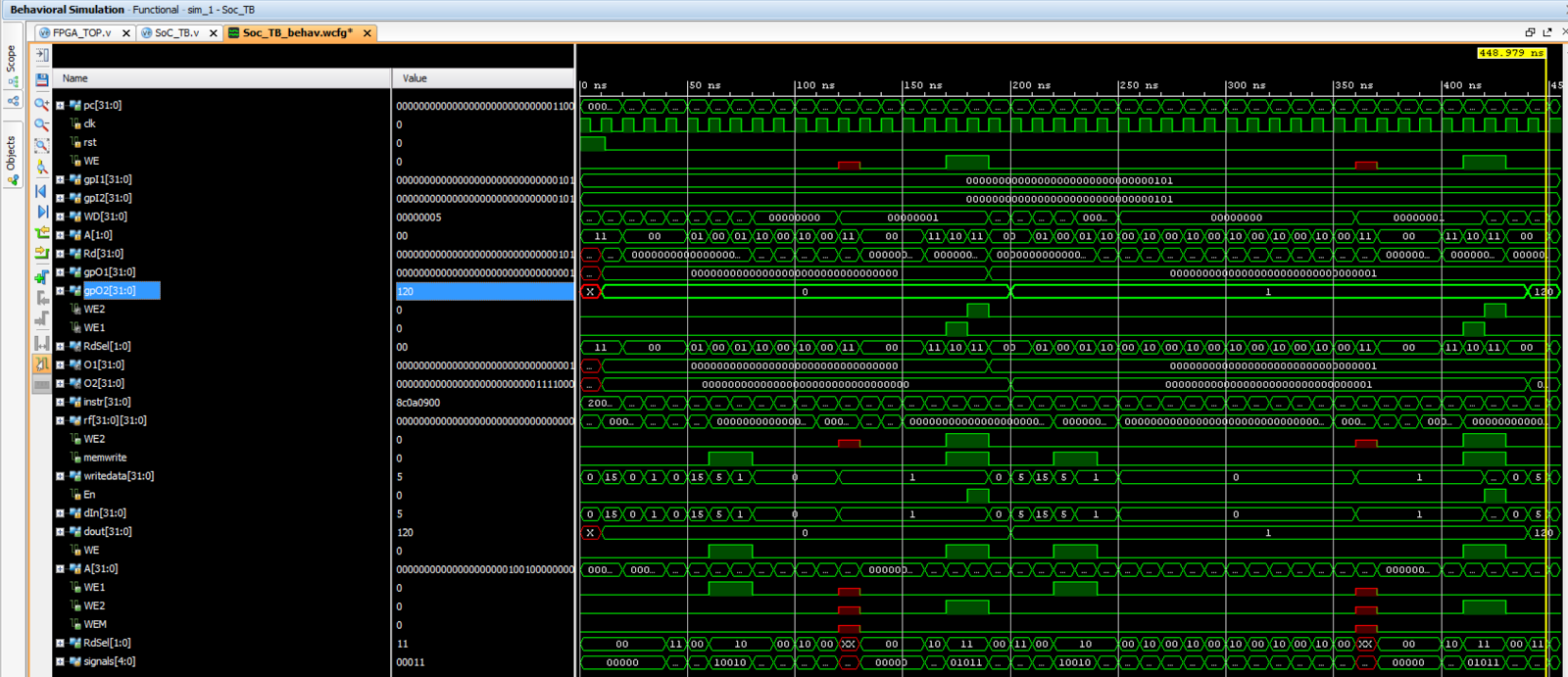


Figure 1: Test bench of the System on Chip

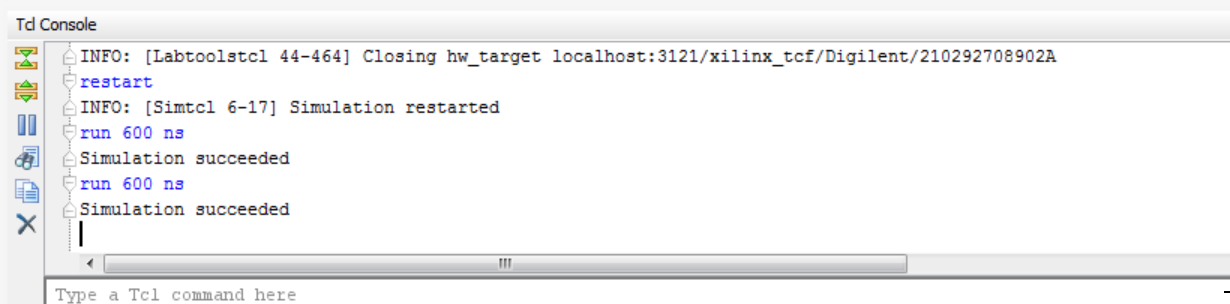


Figure 2: Display of Success if the value of the output is 120 for 5!

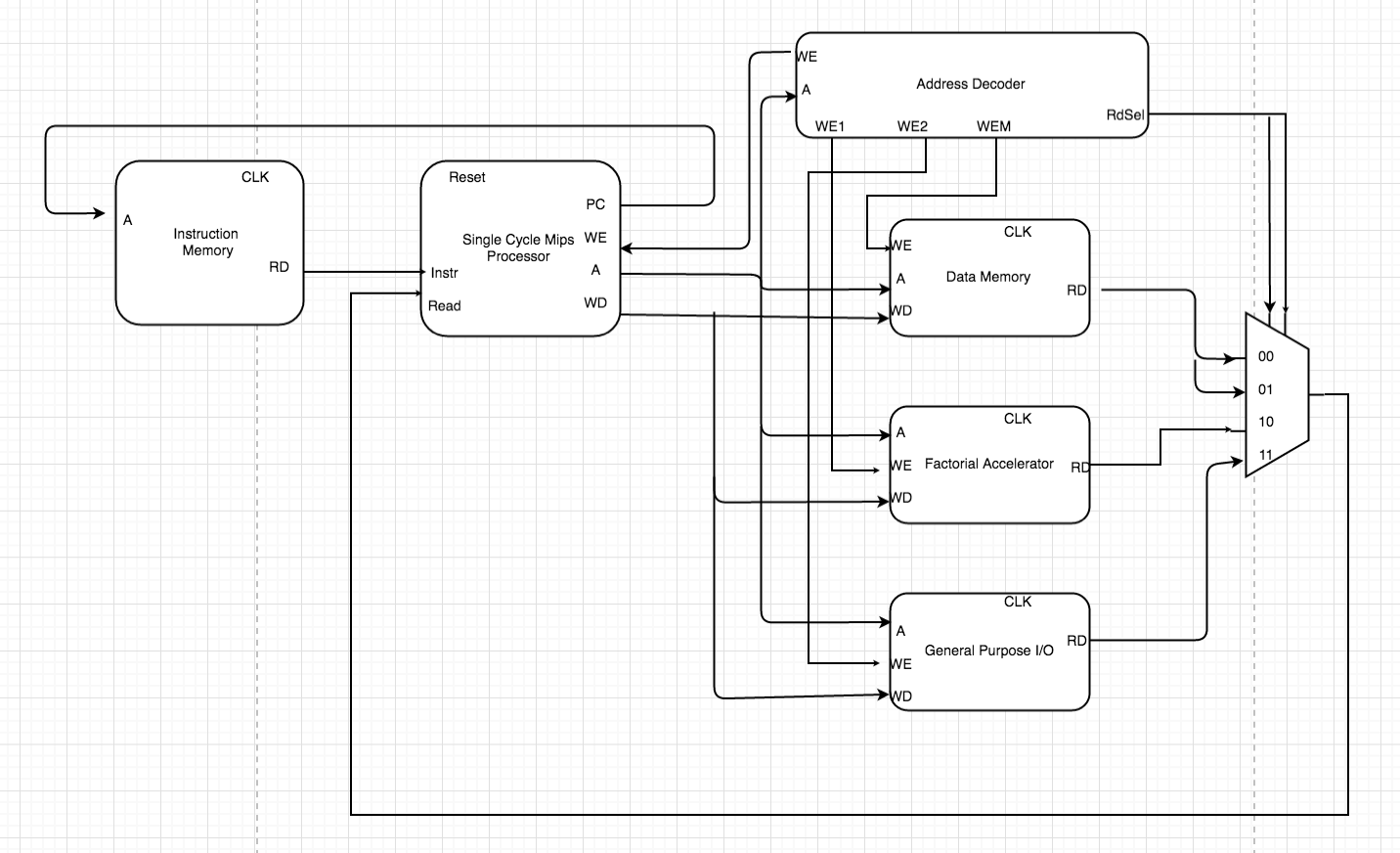


Figure 3: SOC Block Diagram

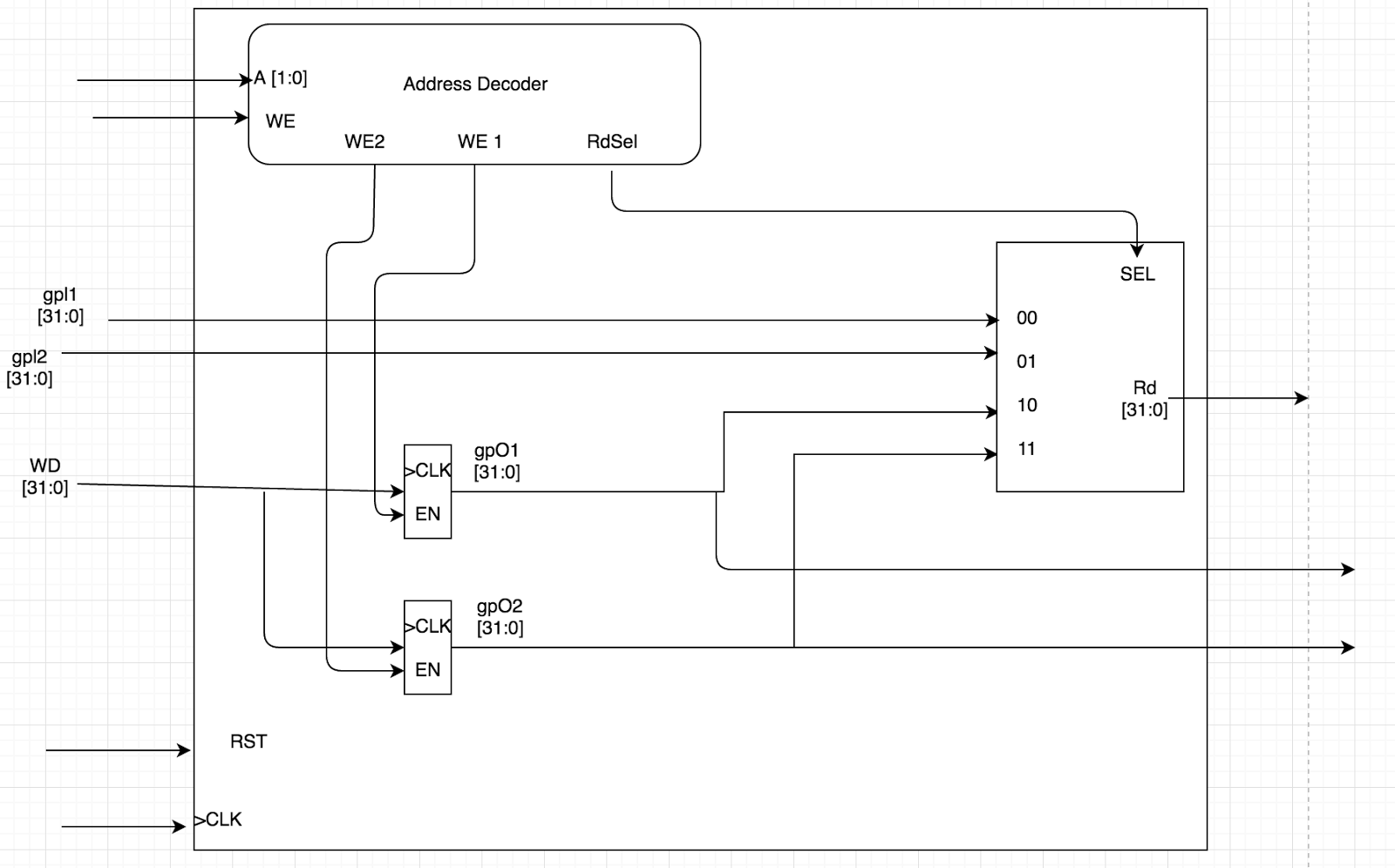


Figure 4: GPIO Block Diagram

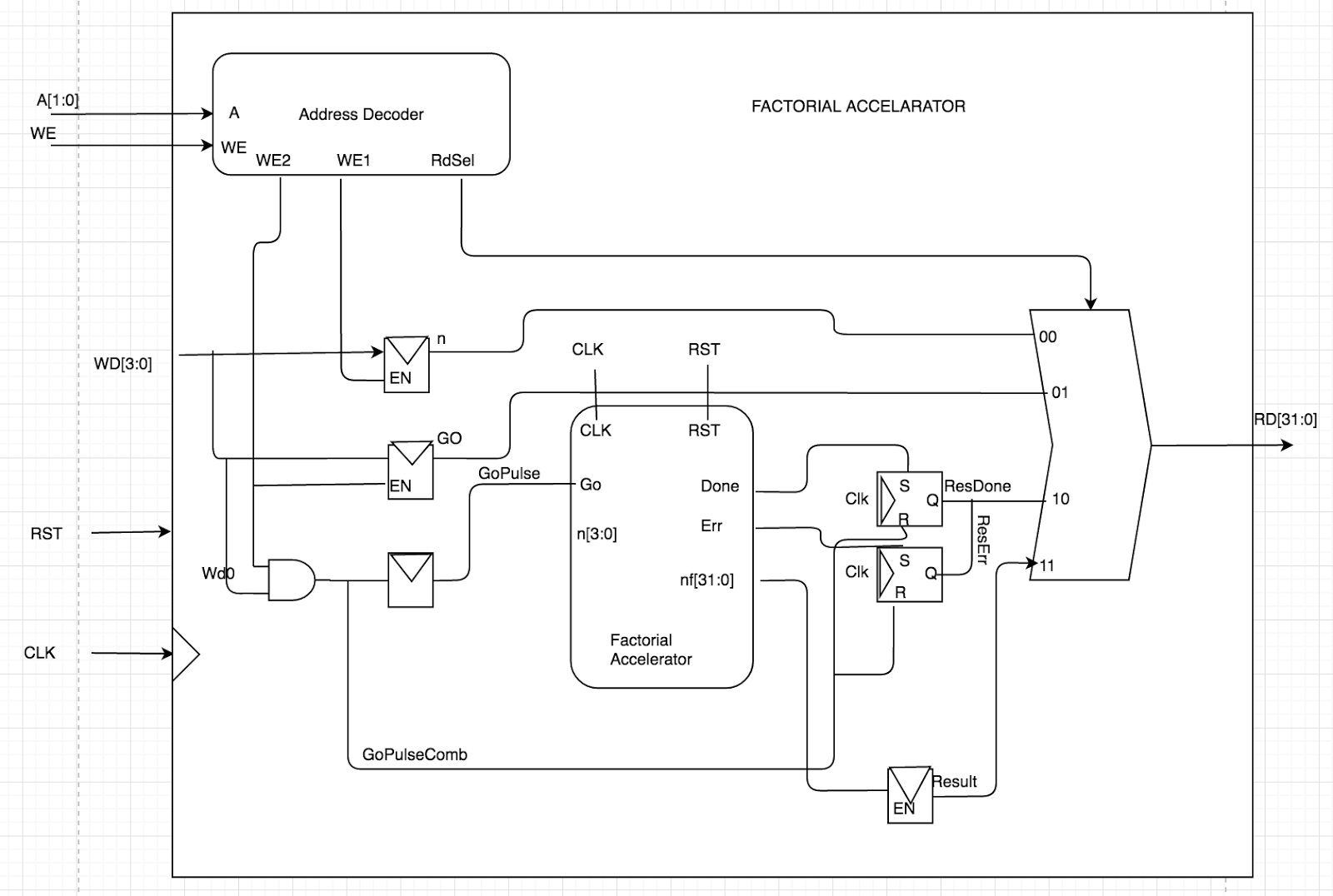


Figure 5: Factorial

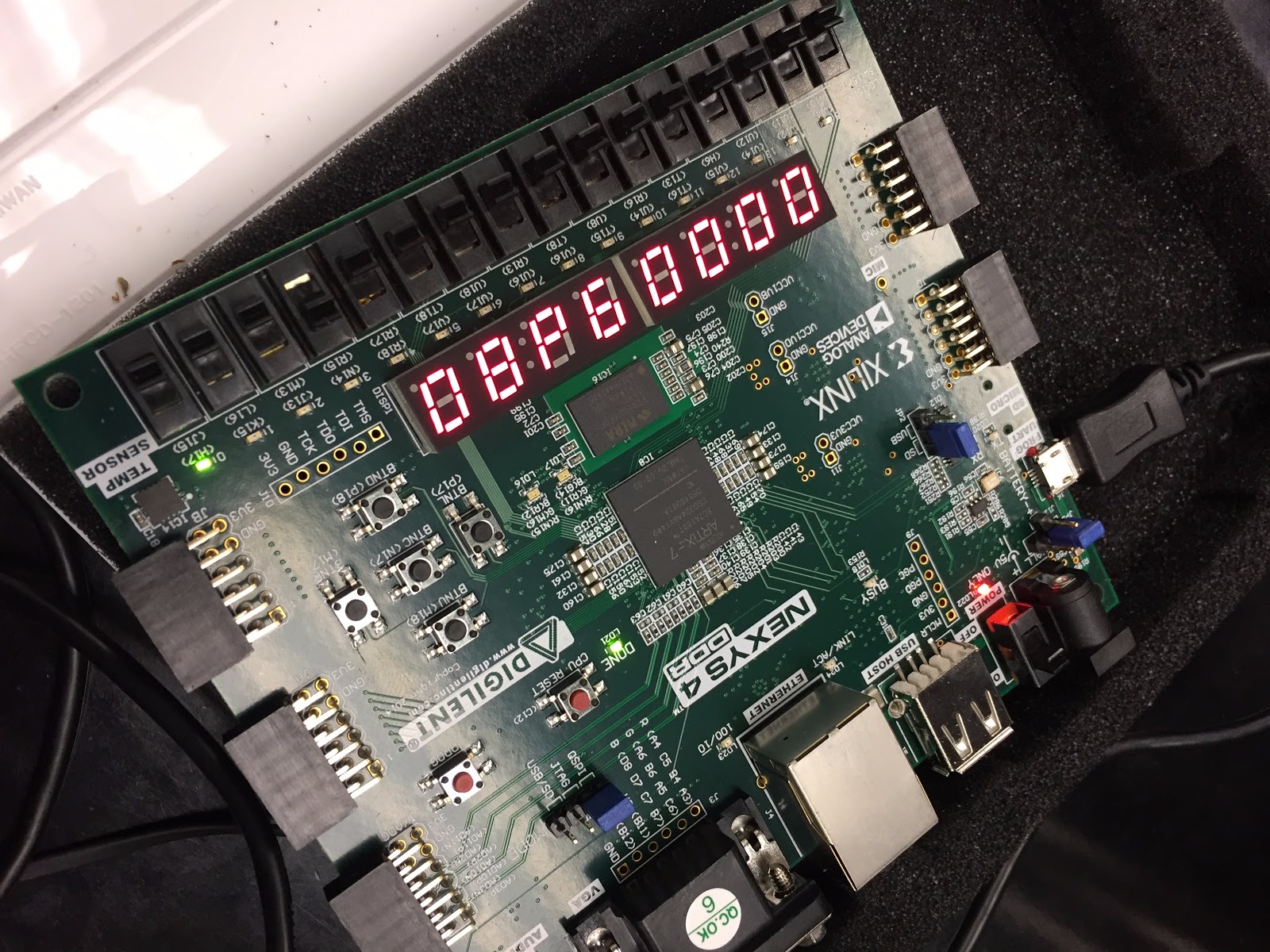


Figure 6: SoC FPGA Implementation