California Polytechnic State University, Pomona

ECE 4304.01

Lab 5

Adrian Alarcon

Isaac Elizarraz

Jacob Swenke

03/17/21

Architecture

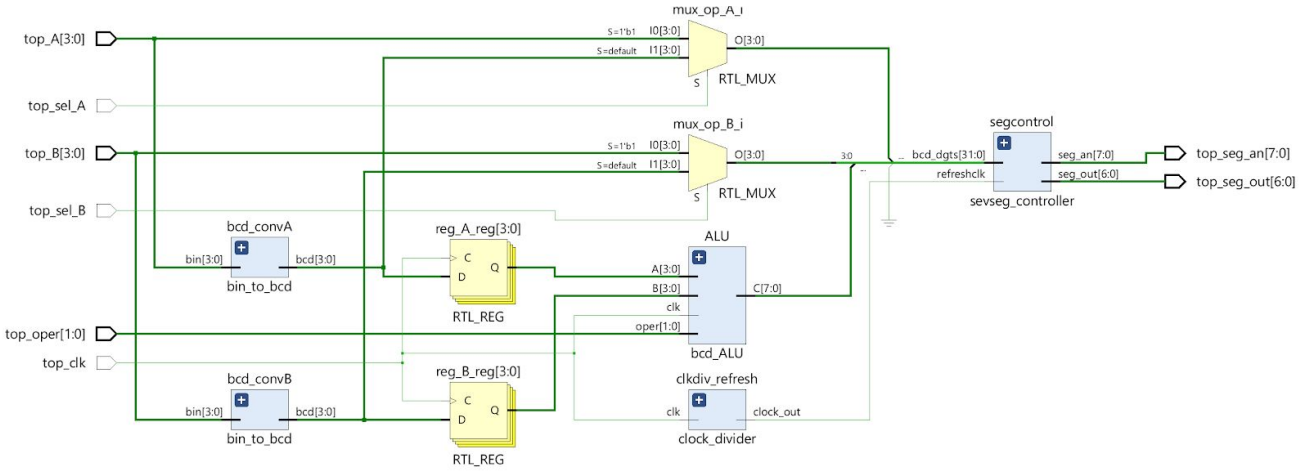

Fig. 5.1 - Elaborated Design

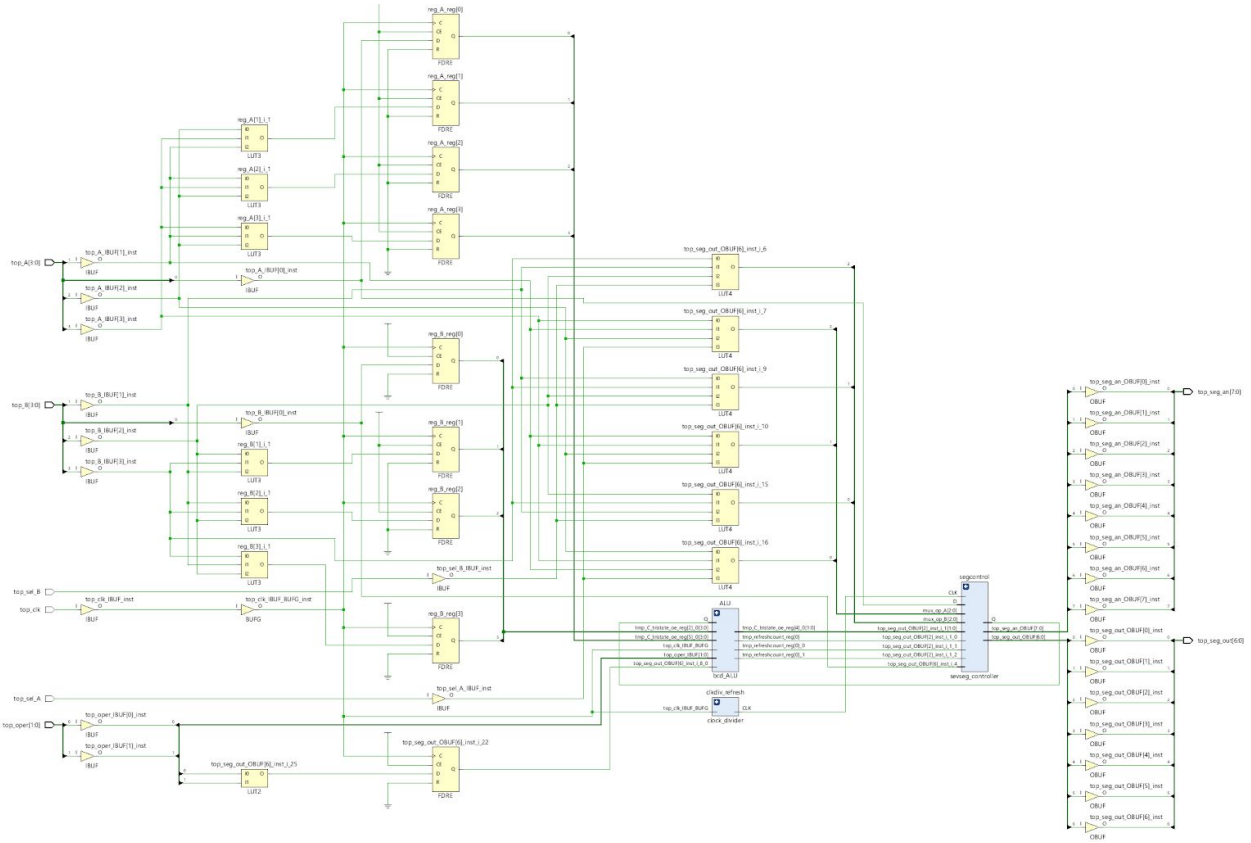

Fig 5 .2 - Implemented Design

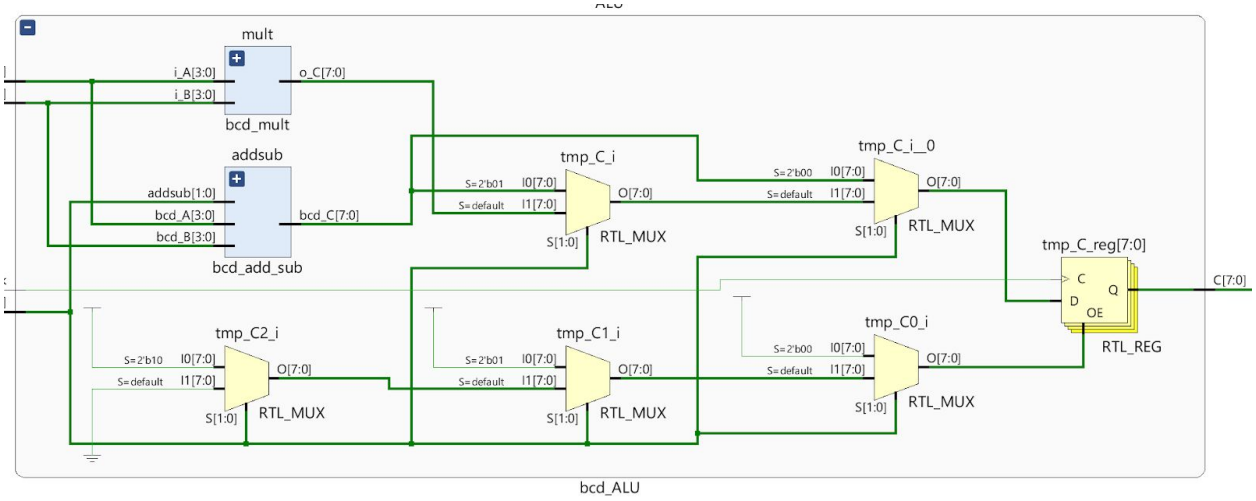

Fig. 5.3 - Elaborated Design of ALU

Trick of the Codes

We implemented a 2-bit select that is used to select which mode the system will be in. “00” is the default state of adder and it will add A+B; “01” the system is set in subtractor mode and will subtract A-B; “10” will set the ALU to act as a multiplier, it will multiply A x B. “11” currently does nothing.

The ALU is composed of the multiplier and adder/subtractor. It is then meshed with the signal tmp_C. It is then outputted with an 8-bit C which is the Sum, Difference, or Product depending on which mode the system is in.

The add/sub module was done with two 4 bit bcd adders and other logic to make sure the carryout was handled correctly. The 4 bit bcd adders are ripple carry adders constructed from 4 full adders each. By making a 2x1 mux before the input into the first adder, we were able to switch whether the addition was A+B or A-B (using B’s 9’s complement in the adder).

The Multiplier is a 4-bit Array Multiplier and outputs an 8-bit product. The binary product is then fed through an assortment of full adders and half adders and outputs an 8-bit binary product that goes through the binary to BCD decoder which allows it to output a BCD output.

An interesting point in this lab, or “trick”, was our necessity to use two different binary-to-bcd decoders that allowed us to separate our operations easily between the ALU’s subtraction and addition. Obviously, creating two different entities will use more resources in the lab, and although this may not be the desired result in terms of efficiency and power, it was a trick that we had to implement in order to guarantee success in the lab. This allowed us to focus on more areas of the lab, such as the divider portion, which we were unable to finish so we scrapped it from the project ultimately. Ideally, continuing forward we will aim to optimize our labs.

In order to successfully output our results to the FPGA for this lab, we again had to utilize a seven segment controller which would cycle through a register of the seven segment compatible values to output to various selected seven segs.

Corner Cases

- N/A

Resource Information

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!								93	53	0.0	0	0
✓ impl_1	constrs_1	write_bitstream Complete!	4.730	0.000	0.222	0.000	0.000	0.123	0	92	53	0.0	0	0

Figure 5.4 - Design Runs

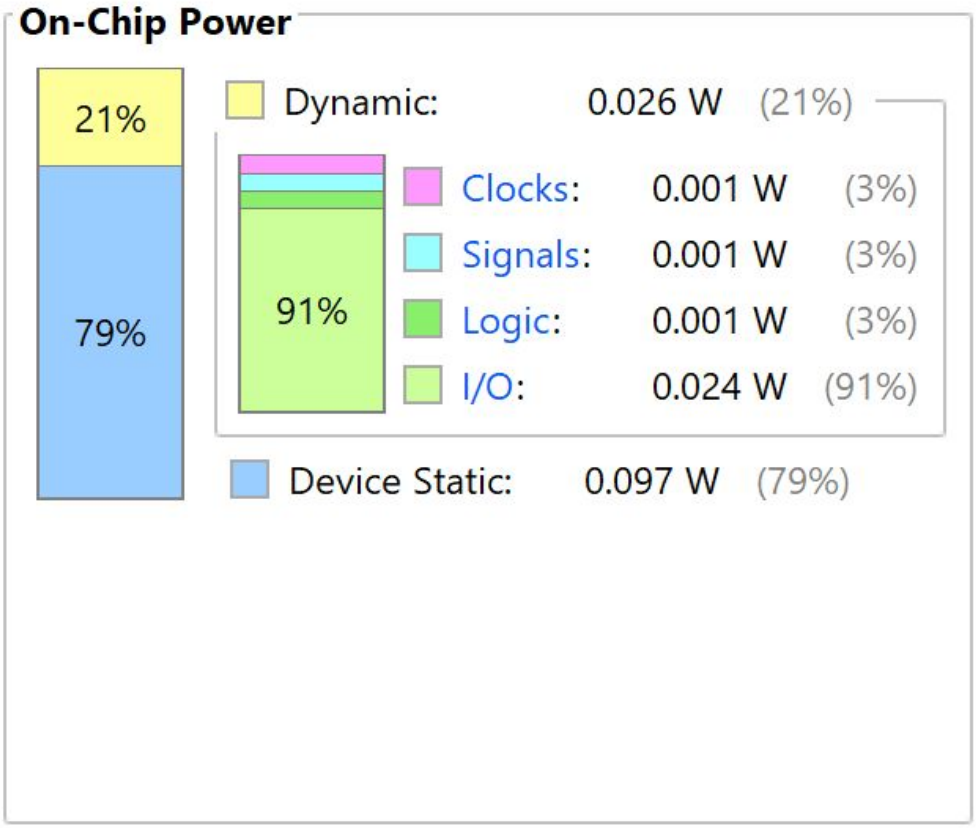

Fig. 5.5 - On-Chip Power