California State Polytechnic University, Pomona

Lab 5: Signed Alien Calculator

Group F:

Danny Manzo, Bronco ID: 014451190

Maher Mateen, Bronco ID: 013891189

ECE 3101L – Signals and Systems Laboratory

Professor Mohamed Aly

18 July 2022

1. **How to Do Signed Arithmetic:**

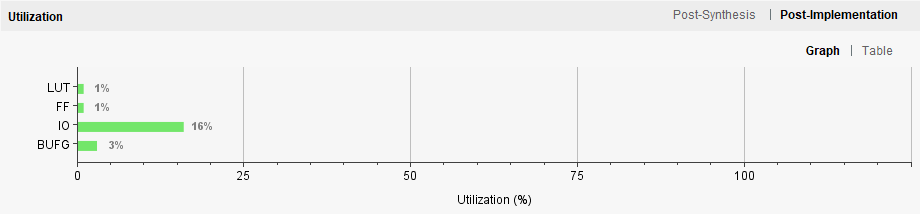
To perform signed binary addition, we get the 2’s complement form of the two signed binary numbers and we add them the same way for unsigned binary. Meanwhile, for signed binary subtraction, we take the 2’s complement of the number that is supposed to be subtracted, then we follow the same rules as signed binary addition. Lastly, to perform signed binary multiplication, we multiply the magnitude of the two inputs, then use the original sign bits to determine the sign of the output: if the sign bits are the same, the output is positive, if the sign bits are different, the output is negative.

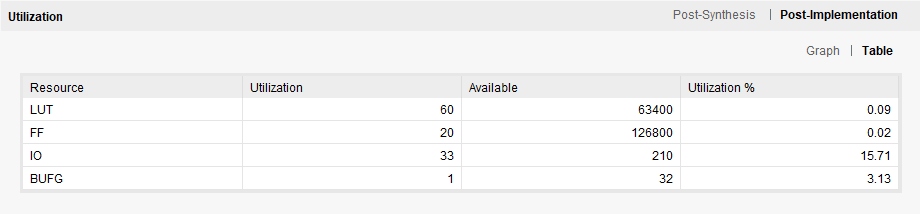
1. **How We Coded It:**

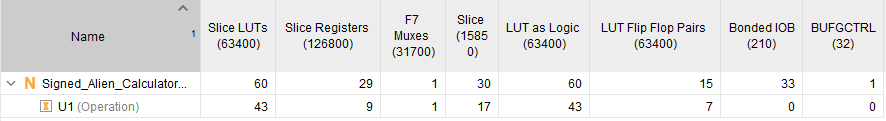
To code the Signed Alien Calculator, we used a mux to select the digit being displayed while cycling from one display to the next at a rate of 1/16 ms and keeping display AN[4] always OFF. The first 2 displays from the right, AN[0] and AN[1], displayed the signed input digit called portA, the following 2 displays, AN[2] and AN[3], displayed the signed input digit called portB, and the last 3 displays, AN[5], AN[6], and AN[7], displayed the signed output digit called value. portA and portB are controlled by switches 0-4 and switches 5-9 respectively, with switches 4 and 9 being the sign bits of portA and portB. Next, to convert the binary input to BCD, we used a truth table for the 7 segment display, driving specific segments to high or low depending on the digit we want to display. To create our alien language, we made use of an if else statement, first checking whether we are displaying the signs of portA, portB, or value, then checking whether the signs are negative or positive and outputting the corresponding a\_to\_g binary inputs to the 7 segment display. If we were not on any of the sign displays, then we followed the 7 segment display truth table. To code the addition and subtraction operations, we compared the sign bits of the two inputs, portA and portB, and either added or subtracted the magnitudes of portA and portB depending on which had a greater magnitude and the combination of 2 positive numbers, one positive and one negative, one negative and one positive, or 2 negative numbers and changing the sign bit of value accordingly. For the multiplication operation, we simply multiplied the magnitudes of portA and portB, and checked whether they had the same signs, meaning the output is positive, or different signs, meaning the output is negative.

1. **Vivado Data Collection:**

In Vivado, after running the synthesis and report utilization, we found that there was 60 look-up tables (LUTs), 29 slice registers, 1 F7 Mux,30 slices, 60 LUTs as logic, 15 LUT Flip Flop Pairs, 33 bonded input/output bits (IOB), and 1 BUFGCTRL. The total power of the full adder with the demultiplexer was reported to be 0.085 W.

Figure 1: Resource Utilization Graph

Figure 2: Resource Utilization Table 1

Figure 3: Resource Utilization Table 2

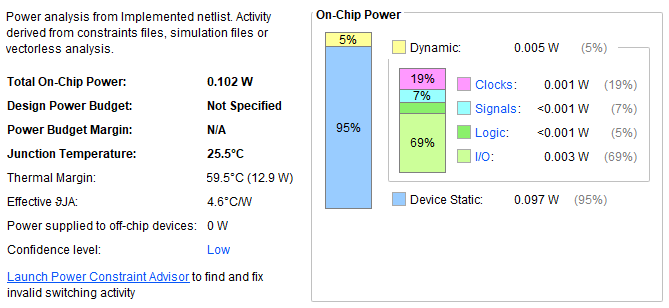


Figure 4: Power Usage