Spring 2019

California State University, Northridge

Department of Electrical & Computer Engineering

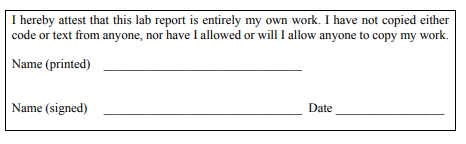
Lab Experiment 8

Arithmetic-Logic Unit Modeling

April 11, 2019

ECE 526L

Written By: Juan Silva

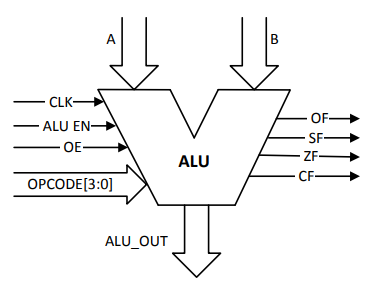


**Introduction:**

In this experiment, an Arithmetic-Logic Unit will be designed using behavioral logic. An ALU is a digital circuit that can perform arithmetic and logic operations. This circuit, when enabled, will receive two data inputs and a unique opcode that will determine the type of operation to be executed. This ALU will support six instructions consisting of two arithmetic and four logical operations. Additionally, the carry flag, overflow flag, zero flag, and sign flag will be defined based on the operations that set them. The output will be the result of the given operation as well as the flags that were set during that operation. Any undefined operands should not modify the output. If the output enable is low, the circuit must produce a high Z output and preserve the flags. For simplicity, the four flags are set regardless if the data inputs are signed or unsigned.

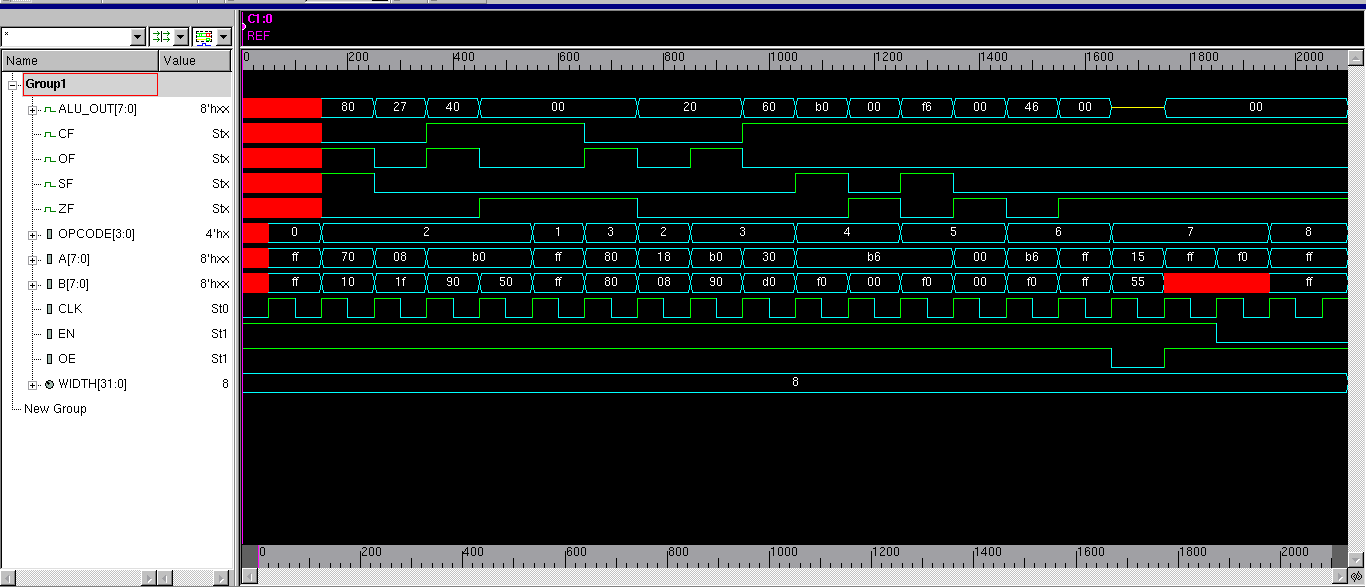
**Procedure:**

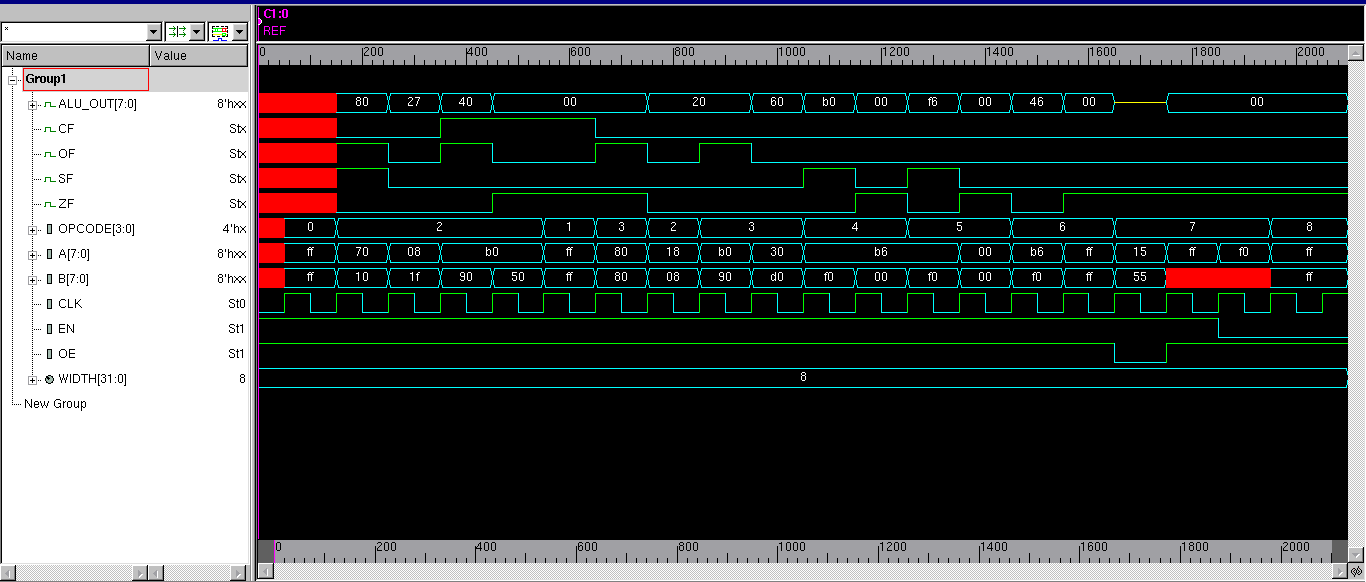
1. Local parameters are used to assign an opcode for all six operands. A switch case is used to describe the operation of each operand when a match occurs. Any undefined operand will be caught with a default case where the output is not modified.
2. Arithmetic and logic operators are used to define each operand. Additional if statements are included to set the appropriate flags based on the operand that was executed.
3. The carry flag (CF) is set when the result of an unsigned arithmetic operation results in a carry out of the MSB. For addition, the MSB of inputs A and B are OR’d together and compared with the MSB of the output. If the MSB of A or B are one and the MSB of the output is zero, then a carry has occurred. For subtraction, the algorithm remains the same only if input A is less than input B and if subtraction is done using Two’s Complement.
4. The overflow flag (OF) is set when the result of a signed arithmetic operation was too large for the ALU. This is checked by comparing the MSBs of inputs A and B as well as the output. If they do not match, then an overflow has occurred. In other words, the sum or difference of two positive numbers resulted in a negative number and vice versa.
5. The sign flag (SF) is set when the result of any operation results in a logic one in the MSB of the output. This is checked by simply checking if the MSB of the output holds has a one.
6. The zero flag (ZF) is set when the result of any operation results in all zeros. This is checked by using a reduction AND and checking if all bits are zero.
7. The output enable (OE) input allows the ALU to produce an output. If OE is low, the flags must be preserved and the ALU will output a high Z value.



***Figure 1.1*** *- Arithmetic-Logic Unit*

**Results:**

**  
*Figure 1.2*** *- Waveform output of each operand case.*

**

***Figure 1.3*** *-Waveform output with signed integer inputs / outputs.*

**Discussion:**

An ALU can be modeled using behavioral code and switch case statements. The design of this ALU is rather simple given a 4-bit opcode and only 6 operations are assigned but the module itself is still complex. Switch statements allow for multiple operands to be defined given a unique opcode. Therefore, this module can support additional operations for opcodes that were not used. Behavioral modeling has the advantage of modeling an ALU sequentially while checking for changes for its inputs. The use of local parameters help avoid opcodes to be rewritten. These opcodes can be enumerated but declaring the inputs as local parameters is better practice to prevent issues like overwriting.

The purpose of the flags are to provide additional information to the user when operands are executed. As a hardware designer, the design for each flag disregards the idea of having signed or unsigned inputs for arithmetic operations. The carry flag is only concerned with unsigned inputs whereas the overflow flag is concerned with unsigned inputs. For the logical operations, the only flags that will be set will be the zero or negative flag.

If the data inputs and data output are read as signed inputs, this will affect the arithmetic flags that are being set. The carry flag will not be set because of the algorithm used to set it. A < B will not be true in one of the test bench cases. Now that the inputs are signed, A will be greater than B and the carry flag is not set. Therefore, the inputs should not be signed for this ALU design.

**Conclusion:**

The ALU is an important circuit in computer processing. All of the instruction executions will be performed with this module. Two arithmetic operations and four logical operations are assigned for this ALU. These operands are sufficient for normal operation of a computer. Flags are important to the user to notify the user for any behaviour based on an operation. As stated previously, the arithmetic flags are based on the interpretation of the inputs. Therefore, the inputs should remain unsigned and use Two’s Complement to determine signage for the input and output bits.