Department of Electrical and Computer Engineering

CPE 462/562 VHDL: Simulation & Synthesis

Midterm Project: 4-bit ALU Design and Implementation

Katelyn Charbonneau
Email: kc325844@wne.edu
Date: 10/27/2017

Tasks	**Grades**
Task 1. Design, test, and verify 4-bit ALU (50')	
Task 2. Program DE2-115 and demonstrate 4-bit ALU (10')	
Task 3. Display INPUT and OUTPUT decimal equivalent values on SSDs (20')	
Report (20')	
Bonus (10')	
Total (100/110):	

1 Objective

The objectives of the project are to be familiar with behavioral architecture; be able to write a behavioral VHDL code to design a 4-bit arithmetic logic unit (ALU), write a testbench to verify the design, and implement the ALU on FPGA.

2 Design Procedure

The top level design block is shown in Figure 1.

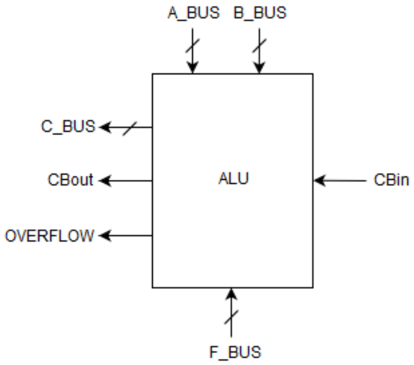

Figure 1: ALU Top Level Block

Where A_BUS, B_BUS, and C_BUS are SIGNED(N-1 DOWNTO 0), F_BUS is STD_LOGIC_VECTOR(2 DOWNTO 0), and CBin, CBout, and OVERFLOW are STD_LOGIC.

The operations that the ALU can perform, as listed in the project requirements, are listed below:

F_BUS	**C_BUS**	**CBout**	OVERFLOW
000	A_BUS	PASS	0
001	A_BUS + B_BUS + CBin	CBout	Overflow
010	A_BUS - B_BUS - CBin	CBout	Overflow
011	A_BUS *OR* B_BUS	PASS	0
100	A_BUS *XOR* B_BUS	PASS	0
101	A_*SRA* 3	PASS	0
110	A_BUS *SLL* 2	A_BUS(N-1)	Overflow
111	A_BUS *ROR* -3	PASS	0

Table 1: ALU Operations

While testing of this design was done using a 4-bit bus width, their sizes are really defined as generics. Thus, there is a variable bus size. As part of this project, a ripple carry adder (RCA) is needed. A fixed 4-bit width RCA design was available and was edited to also allow for a variable bus width. This design was tested using a 6-bit width; sample results are in Section 3. A block diagram for this is shown in Figure 2.

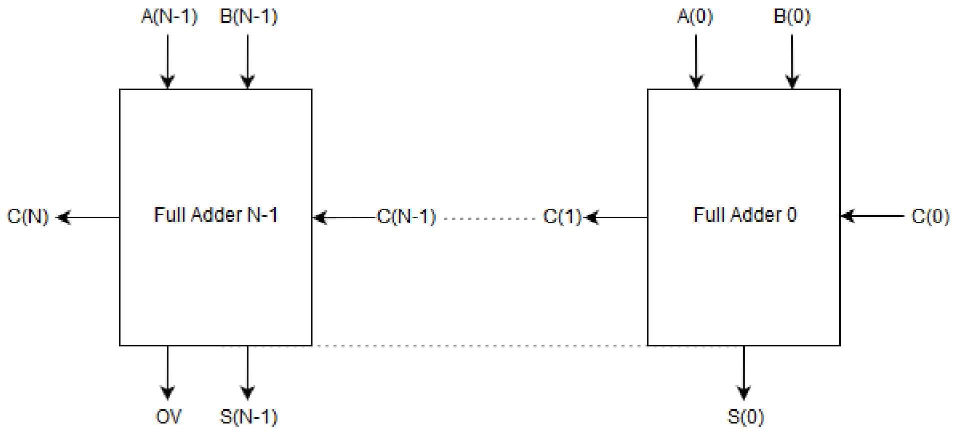

Figure 2: The RCA used to perform arithmetic calculations

Where N is the bus width, A, B, and S (sum) are signed vectors of length N-1, C (carry) is a standard logic vector of length N, and OV is defined as C(N) ⊕ C(N-1).

The RCA is used in cases 1 and 2 - addition and subtraction. A CBout of ”PASS” simply passes the value of CBin through. For subtraction, the complement of B_BUS is taken. All other cases are handled by simple pre-defined operations within VHDL. Note that there is overflow for case 6 if the value of A_Bus(N-1) changes after shifting.

In addition to this basic functionality, additional output ports had to be defined to accommodate implementation onto the FPGA. Inputs and outputs are displayed on both LEDs and seven-segment displays. A previously used design for these displays was re-used as a component. The third component used in the project described behavior for the negative signs that lit up only when the associated bus decimal value was negative.

3 Simulation Results

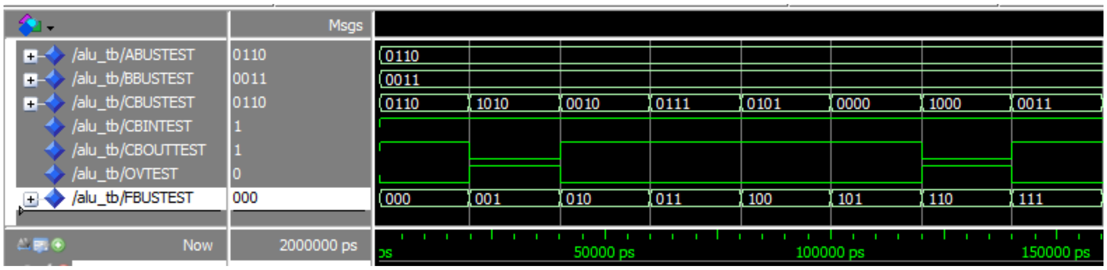

Figure 3: Sample simulation results for the ALU

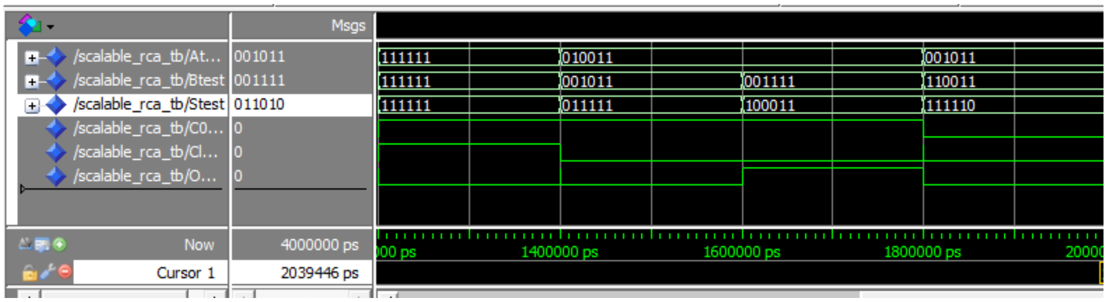

Figure 4: Operations on the RCA design showing variable bus width functionality

4 Demonstration on FPGA

The design was synthesized onto the Altera Cyclone IV E DE2-115F29C7 FPGA.

- User Controls:
  - Input A: Switches 3 down to 0
  - Input B: Switches 8 down to 5
  - Operation Select: Switches 15 down to 13
  - Carry/Borrow in: Switch 17
- Input and Output Displays:
  - Inputs
    - A: Hex displays 7 and 6
    - B: Hex displays 5 and 4
    - Carry/Borrow in: Green LED 8
  - Outputs
    - C: Green LEDs 3 down to 0, as well as hex displays 1 and 0
    - Carry/Borrow out: Green LED 5, as well as hex display 2
    - Overflow: Green LED 7, as well as hex display 3

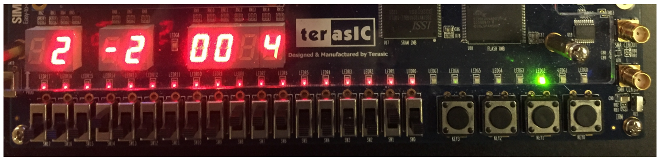

Figure 5: Subtraction of decimal 2 - (-2) = 4

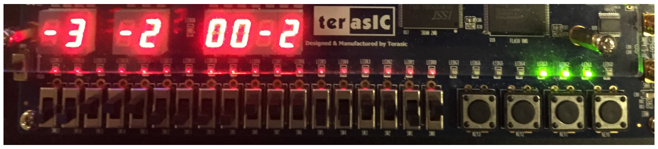

Figure 6: Rotate operation 1101 -> 1110

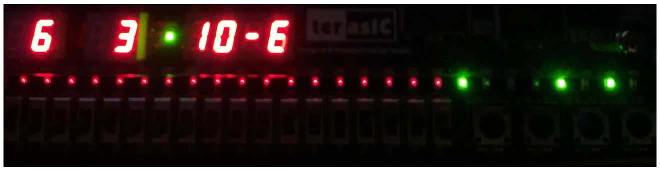

Figure 7: Example showing decimal 3 + 6 with overflow; in dark to show visibility

5 Conlusion

The project simulated and synthesized correctly. The code required sufficient knowledge of structural and concurrent VHDL. Multiple lower-level components were used and many concurrent statements were utilized. Generics were used on multiple levels to allow the size of the ALU to be changed easily, however, the implementation onto the FPGA relied upon a bus width of 4 bits. As such, it could be quite difficult to really change the size as-is. While the size is changeable from just a few constants, more work would need to be done to make it truly scalable.