Name: Xinyu Hadrian Hu, and Duan Wei Zhang

Student Number: 500194233, and 500824903

Date: June 26, 2020

COE 608: Computer Architecture and Design

COE 608: Lab 3, Part 2 Report

Purpose

The purpose of this lab is to generate the components of an 8-bit ALU, with six different operations. The addition and subtraction units must be done in structural form of VHDL. We also have to include the seven-segment display driver into the build for this ALU. This is a simplified version of the 32-bit ALU for testing before implementing the 32-bit ALU.

Design and Implementation

The following: Figure 1, Figure 2, Figure 3, and Figure 4 are the 8-bit ALU, operational inverter, BCD to seven-segment display driver, and the 8-bit register codes needed for the function of the 8-bit ALU. The truth table for the 8-bit ALU is from the waveforms generated in Quartus II. The values of the 8-bit register are placed as HEX values for the input and output of d and Q, respectively, otherwise the table will be unreasonably large and unhelpful to the reader. There is no need to show the 8-bit adder, since the operations of the 8-bit adder is the same as those of the 32-bit and 1-bit adders from the previous lab 3a.

8-bit ALU			
OP Name	**Neg/Tsel**	**ALU-Select**	**Operation Performed**
AND (logical)	0	0 0	Result <= a AND b
OR (logical)	0	0 1	Result <= a OR b
ADD	0	1 0	Result <= a + b
SUB	1	1 0	Result <= a - b
ROL	1	0 0	Result <= a << 1
ROR	1	0 1	Result <= a >> 1

Figure 1: 8-bit ALU Truth Table

Op Inverter	
OP	**not OP**
0	1
1	0

Figure 2: Op Inverter Truth Table

BCD to 7 segment display		
BCD Code	**7seg Code**	**Symbol**
0 0 0 0	0 0 0 0 0 0 1	0
0 0 0 1	1 0 0 1 1 1 1	1
0 0 1 0	0 0 1 0 0 1 0	2
0 0 1 1	0 0 0 0 1 1 0	3
0 1 0 0	1 0 0 1 1 0 0	4
0 1 0 1	0 1 0 0 1 0 0	5
0 1 1 0	0 1 0 0 0 0 0	6
0 1 1 1	0 0 0 1 1 1 1	7
1 0 0 0	0 0 0 0 0 0 0	8
1 0 0 1	0 0 0 0 1 0 0	9
1 0 1 0	0 0 0 1 0 0 0	A
1 0 1 1	1 1 0 0 0 0 0	B
1 1 0 0	0 1 1 0 0 0 1	C
1 1 0 1	1 0 0 0 0 1 0	D
1 1 1 0	0 1 1 0 0 0 0	E
1 1 1 1	0 1 1 1 0 0 0	F

Figure 3: BCD to 7-segment display converter Truth Table

8 bit register				
clk	**clr**	**ld**	**d (hex)**	**Q (hex)**
0	0	1	0 0	0 0
1	0	1	0 1	0 1
0	0	1	0 2	0 1
1	0	1	0 3	0 3
0	0	1	0 4	0 3
1	0	1	0 5	0 5
0	0	1	0 6	0 5
1	0	1	0 7	0 7
0	0	1	0 8	0 7
1	0	1	0 9	0 9
0	0	1	0A	0 9
1	0	1	0B	0 B
0	0	1	0C	0 B
1	0	1	0D	0 D
0	0	1	0E	0 D
1	0	1	0F	0 F
0	0	1	10	0 F
1	0	1	1 1	1 1
0	0	1	1 2	1 1
1	0	1	1 3	1 3
0	0	1	1 4	1 3
1	0	1	1 5	1 5
0	0	1	1 6	1 5
1	0	1	1 7	1 7
0	0	1	1 8	1 7
1	1	1	1 9	0
0	1	1	2A	0
1	1	1	2B	0
0	1	1	2C	0
1	1	1	2D	0
0	1	1	2E	0
1	1	1	2F	0
0	1	1	3 0	0
1	1	1	3 1	0

Figure 4: 8-bit register Truth Table

The final implementation of the 8-bit ALU in block diagram, schematic format is shown in the next page:

See Figure 5 below.

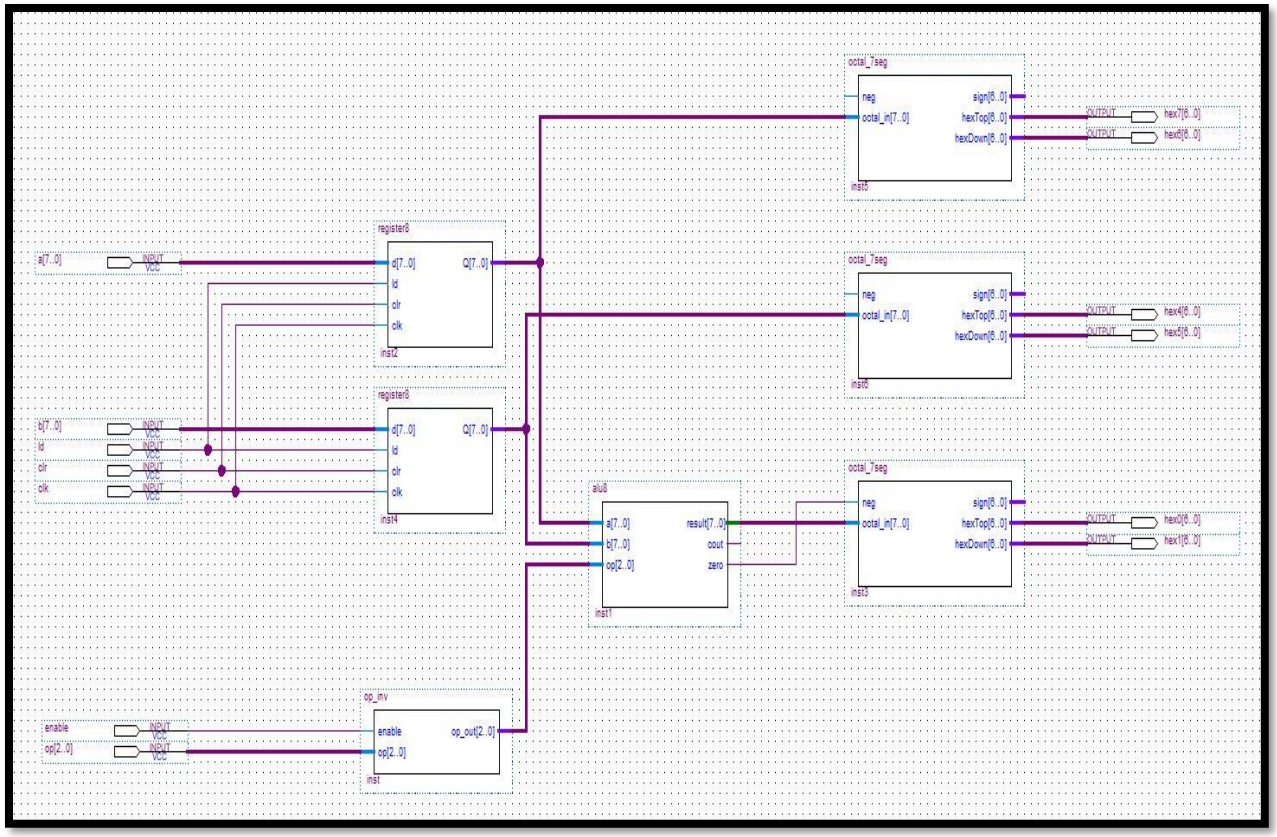

Figure 5: 8-bit ALU Schematic

Observations and Results

Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, and Figure 13 are the waveform results of both functional and timing for the 8-bit ALU.

8-bit Register: Functional and Timing Waveforms

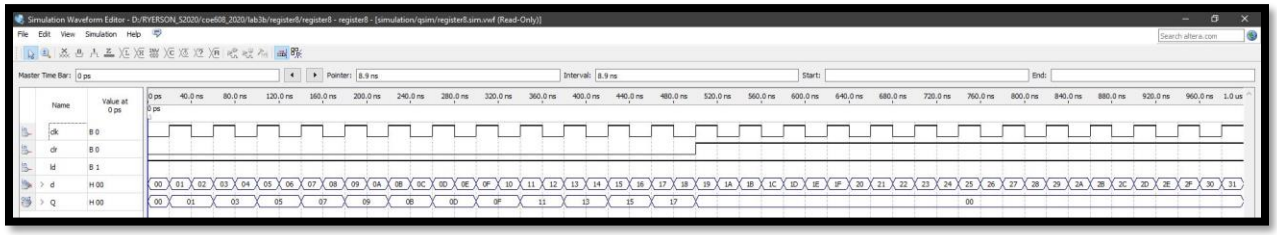

Figure 6: -bit register functional waveform

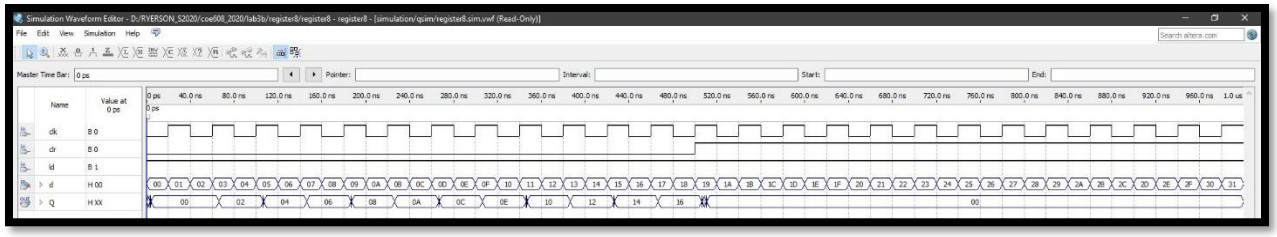

Figure 7: -bit register timing waveform

BCD to 7-segment converter: Functional and Timing Waveforms

Figure 8:: BCD to 7-segment display functional waveform

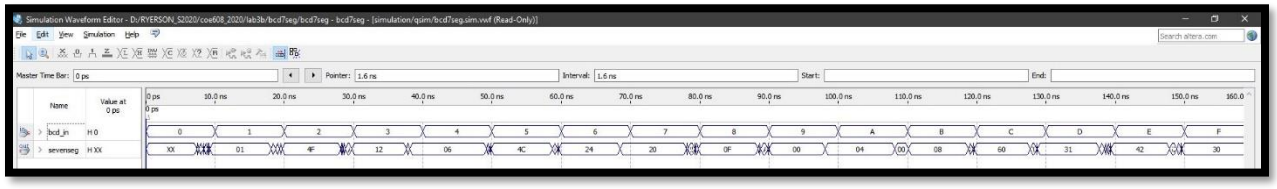

Figure 9: BCD to 7-segment display timing waveform

Operational Inverter: Functional and Timing Waveforms

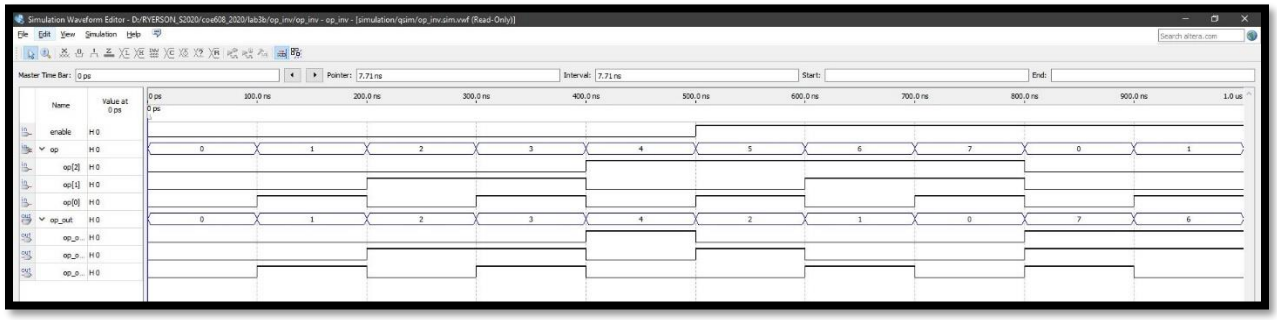

Figure 10: Op-inverter functional waveform

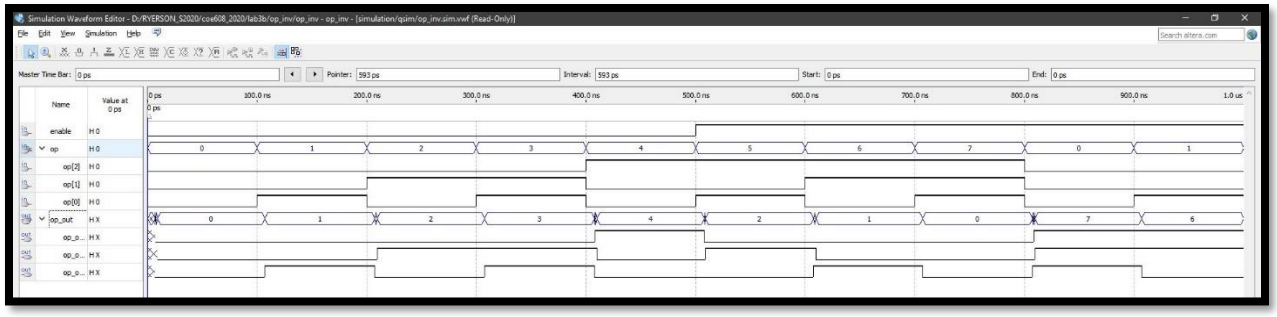

Figure 11: Op-inverter timing waveform

8-bit ALU: Functional and Timing Waveforms

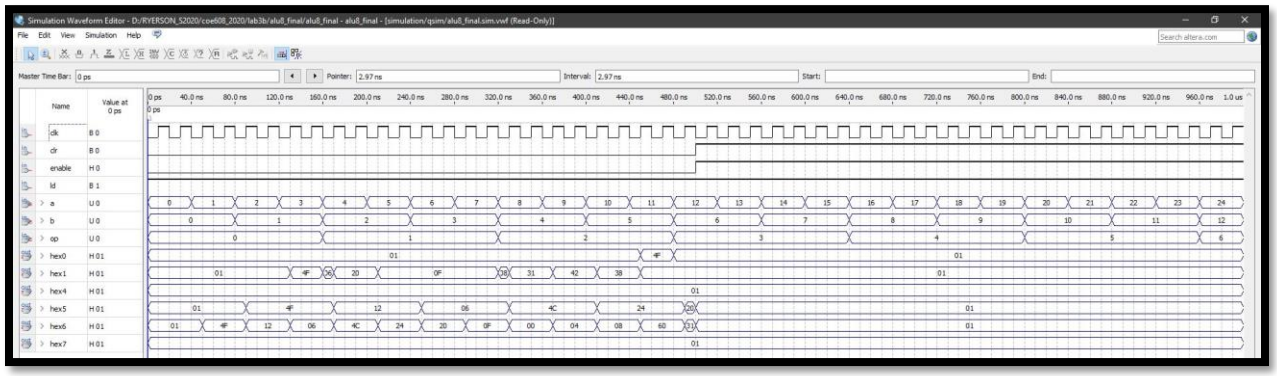

Figure 12: 8-bit ALU functional waveform

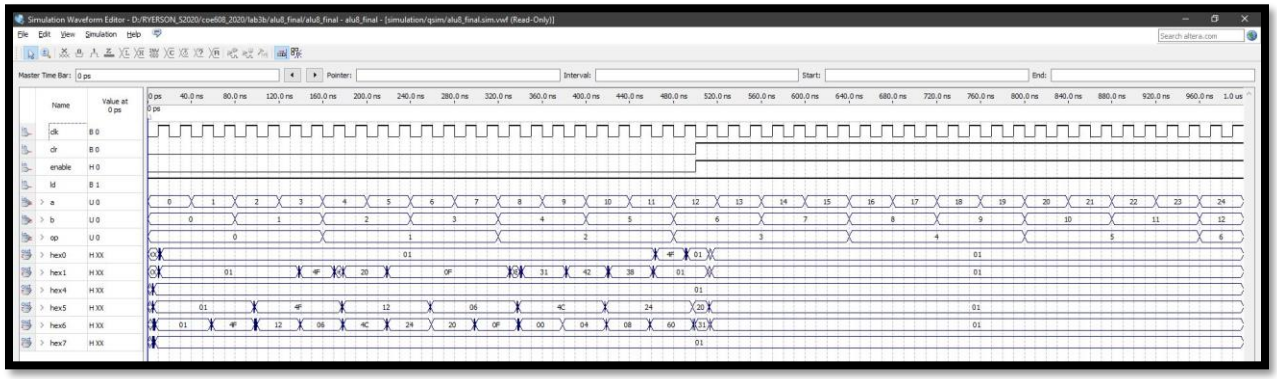

Figure 13: 8-bit ALU timing waveform

Appendices: VHDL Codes for the ALU, 7-segment displays, 8-bit register, and op inverter

The files are attached as PDF to make this document easier to read.