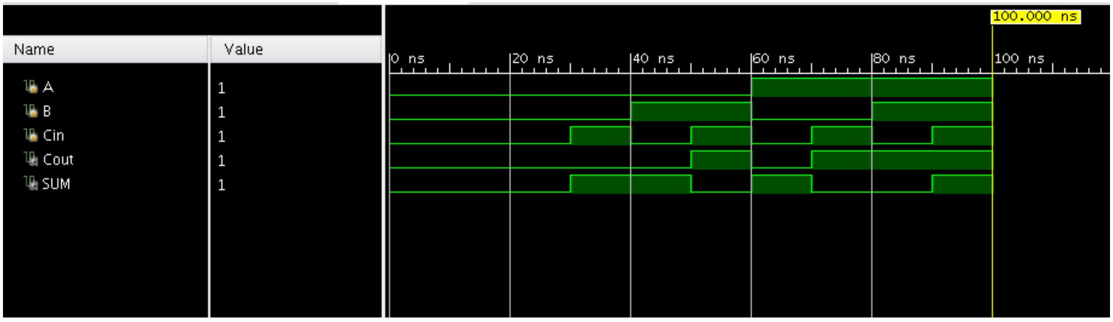
LAB2: ALU

Group Member: Rui Chen/Mengxi Wang

Complete Time: 09/24/2017

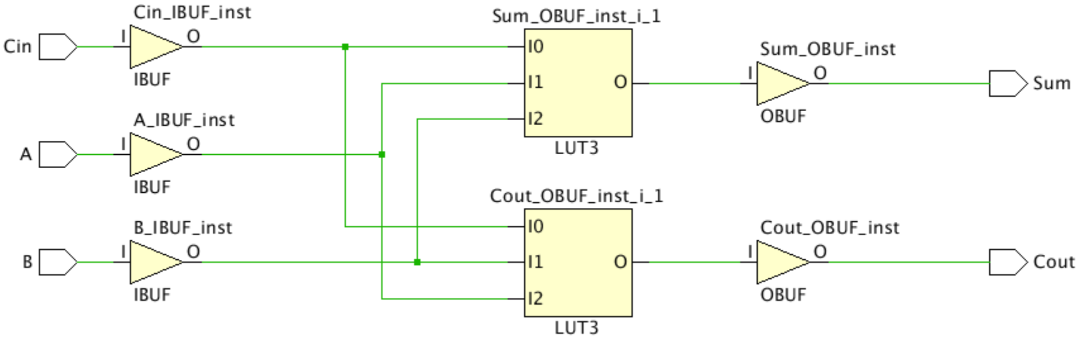
TASK1: Gate-Level Full Adder

1. Waveform:



Img1: Gate-Level Full Adder waveform

2. Vivado Schematic:



Img2: Gate-Level Full Adder schematic

3. Description:

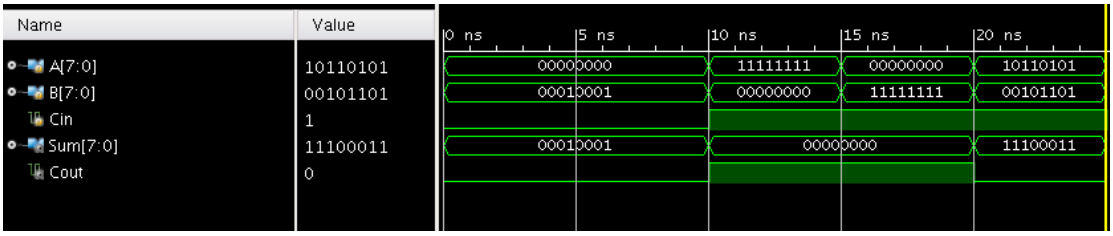
A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

1	1	1	1	1

Form1: Truth table of gate-level full adder

TASK2: Ripple Carry Adder

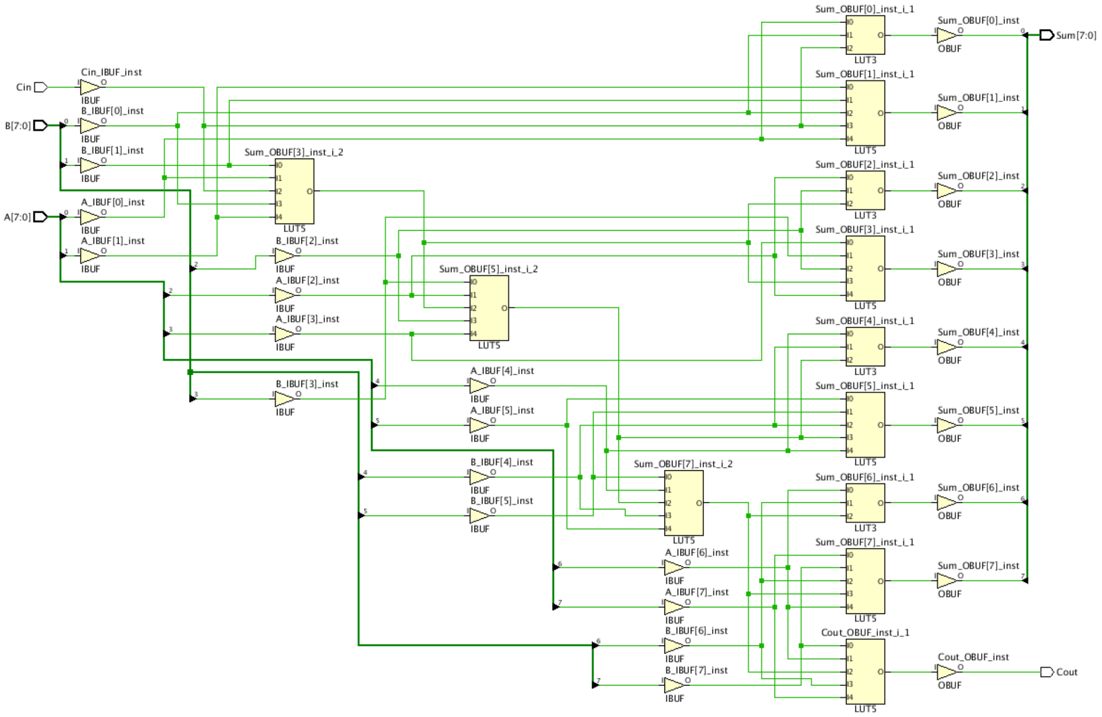
1. Waveform:



Name	Value
A[7:0]	10110101
B[7:0]	00101101
Cin	1
Sum[7:0]	11100011
Cout	0

Img3: Ripple carry adder waveform

2. Vivado Schematic:



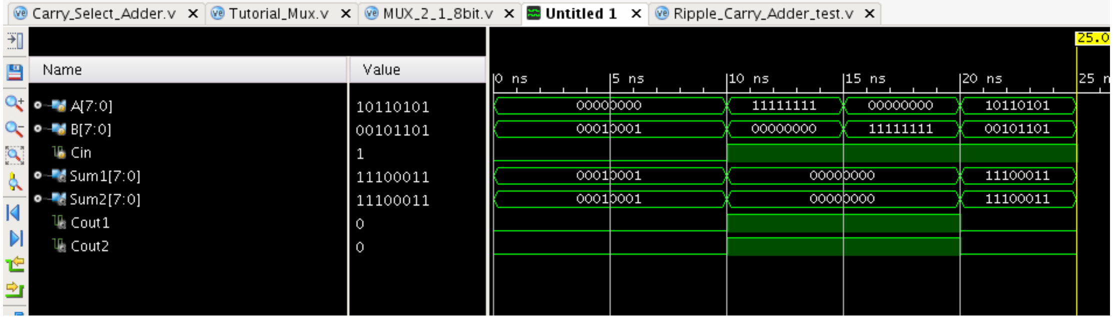
Img4: Ripple carry adder schematic

3. Description:

Use full adder on every bit of A and B, and the Cout is carried to next bit. For example, A=10110101, B=00101101, and Cin=1, so the SUM is 11100011, and Cout of the last bit is 0.

TASK3: Carry Select Adder

1. Waveform:

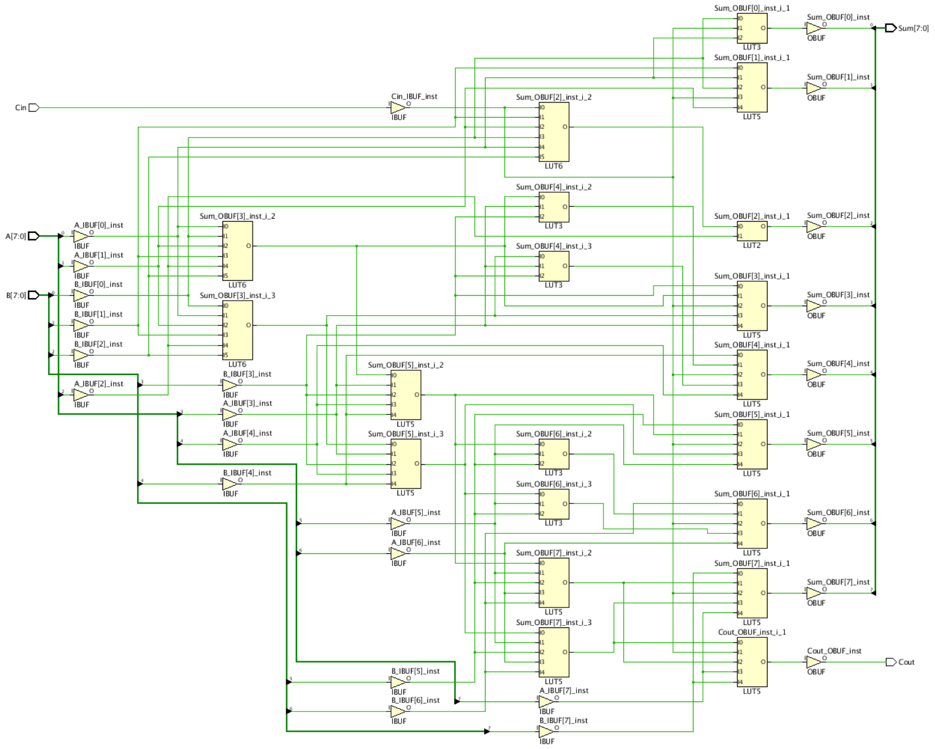


Carry_Select_Adder.v × Tutorial_Mux.v × MUX_2_1_8bit.v × Untitled 1 × Ripple_Carry_Adder_test.v ×

Name	Value
A[7:0]	10110101
B[7:0]	00101101
Cin	1
Sum1[7:0]	11100011
Sum2[7:0]	11100011
Cout1	0
Cout2	0

Img5: Carry select adder waveform

2. Vivado Schematic:



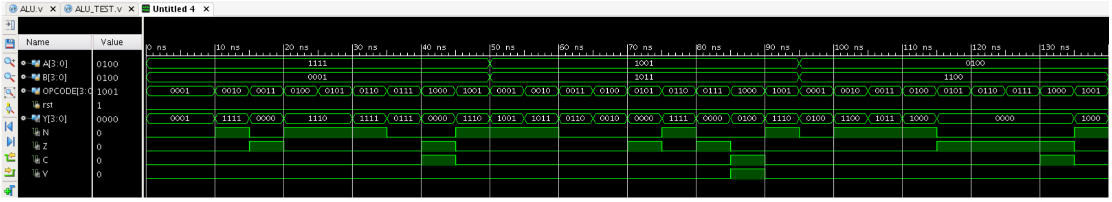
Img6: Carry select adder schematic

3. Description:

We assumed Cin were 1 and 0 separately in two parallel full adder and got two sum and Cout results. We also use another full adder to calculate the Cout as the select signal. Then using a 2:1 MUX to select one as the final result.

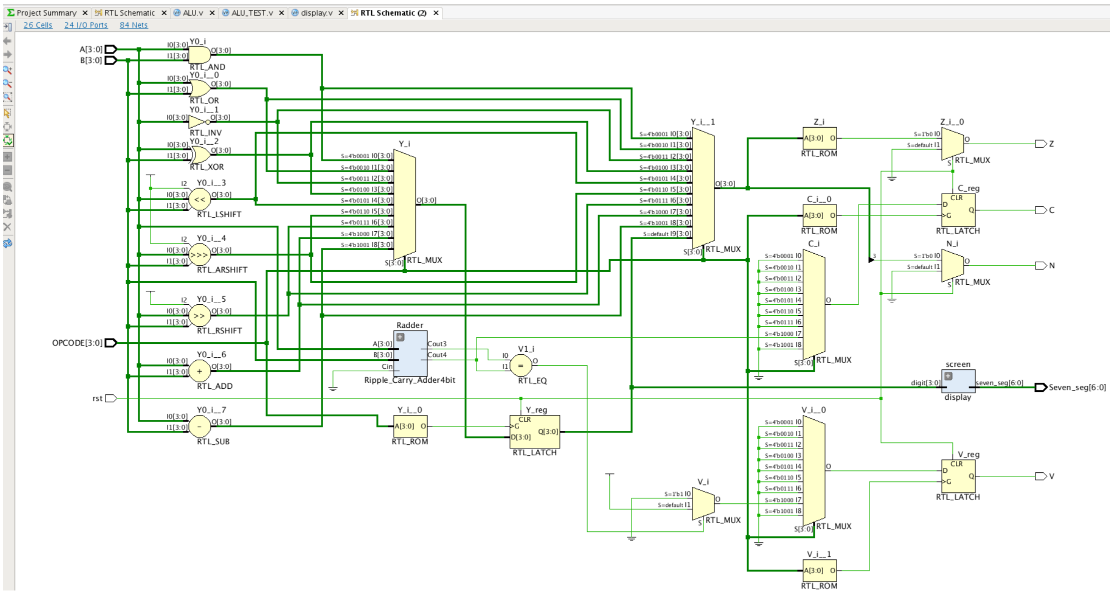
TASK4: ALU

1. Waveform:

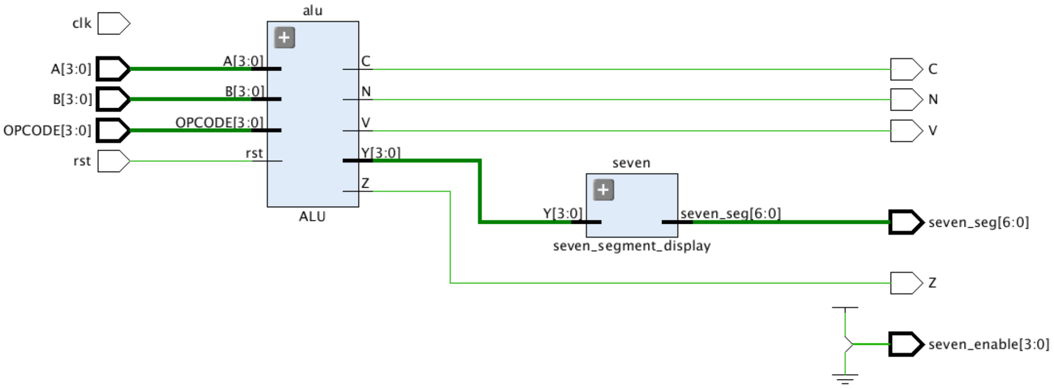


Img7: ALU waveform

2. Vivado Schematic:

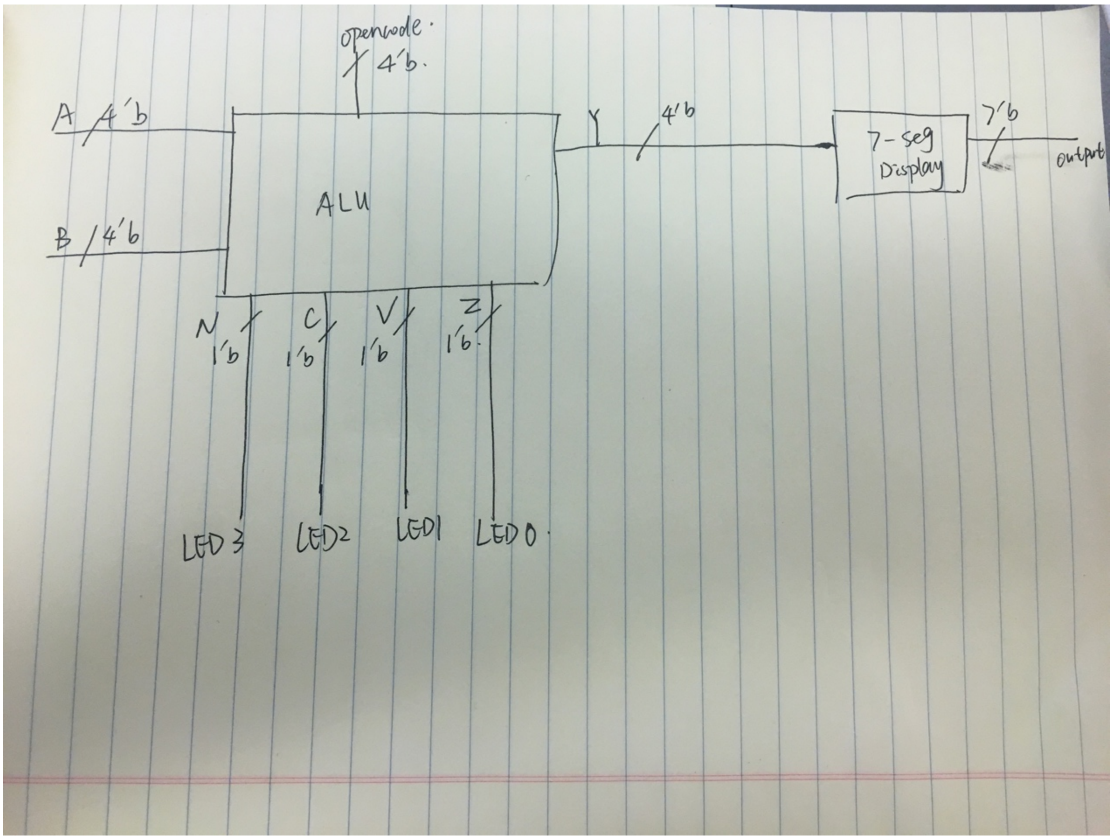


Img8: ALU schematic



Img9: ALU schematic (Top module)

3. Block Diagram:



Img10: ALU block diagram

4. Description:

We used behavioral language to do task4. As required, we use opencode as select signal to decide the operation.

We also add the ripple carry adder in task2 to do the 'add' operation, using the Cout of last bit as flag C, and compared the Cout and Cin of last bit to choose the flag V.