

Cad 4 – ALU

EE 477

Yexuan Chen, Yuxiang Chen, Xinyi Chang

1. Introduction:

The purpose of this mini project is to design a 16 bits ALU that can perform the add, subtract, and, or, xor, invert and compare operation. The output of the ALU is selected by 3 three control. Table 1.1 shows the control inputs corresponding to different operation modes. And it also updates the carry flag, negative flag and the over flow.

Table 1.1 - input pattern

0 0 0	AND
1 0 0	OR
0 1 0	INVERT
1 1 0	XOR
0 0 1	ADD
0 1 1	SUBSTRACT
1 1 1	COMPARE

2. Analytical Approach

One big part of the ALU is the adder, and we choose the sklansky tree adder. The adder can perform both the add and subtract operation. The subtract is done by inverting one of the inputs and taking in the carry-in of '1', and then do the addition. Specifically, when the subtract signal into the adder is '1', the carry-in is also true. And there is also a array of logic gates to perform the logic operations. And the output selection of the ALU is done by an array of MUXs.

3. Schematic

Figure 3.1 shows the top level schematic design of ALU. Figure 3.2 shows the sklansky tree adder. Figure 3.3 shows and, or, invert and xor operation for ALU. Figure 3.4 shows one of the muxi array that used to select operations. All the simulation waveforms is placed in the "verification" section.

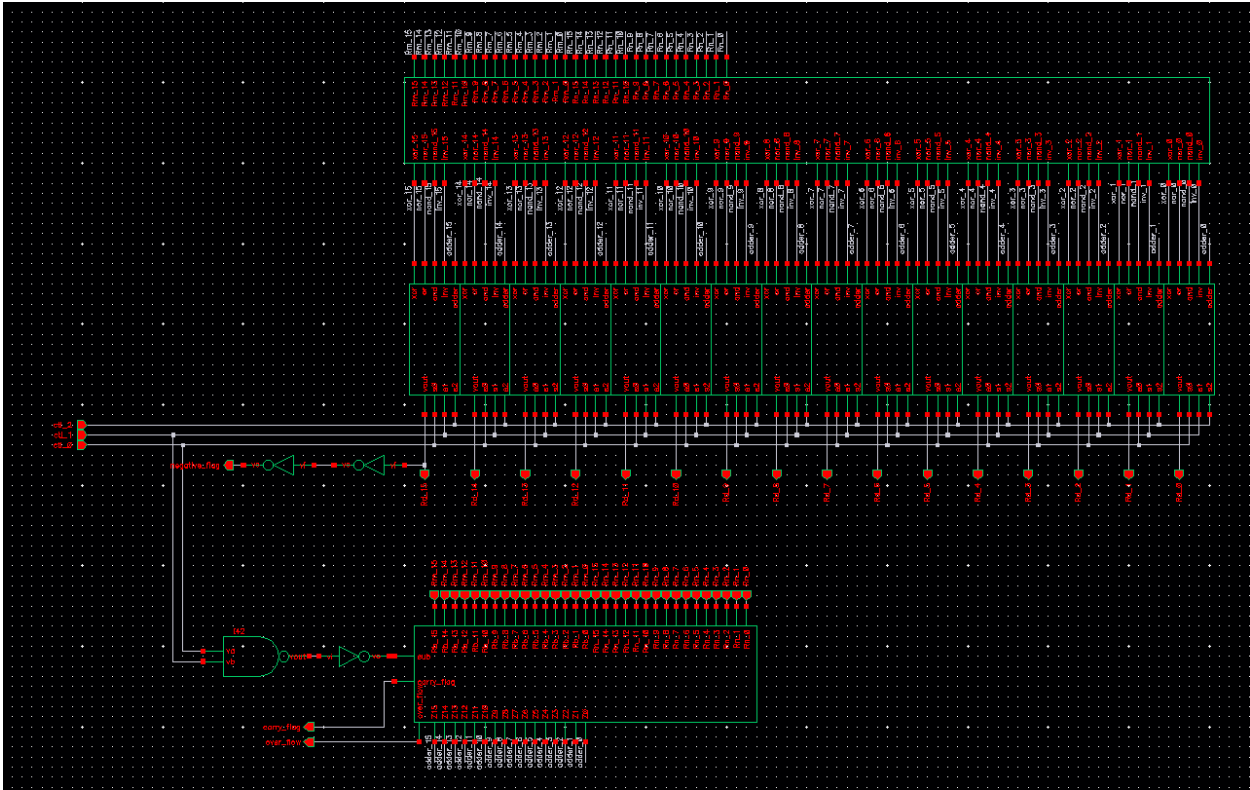


Figure 3.1 Top Level ALU Schematic

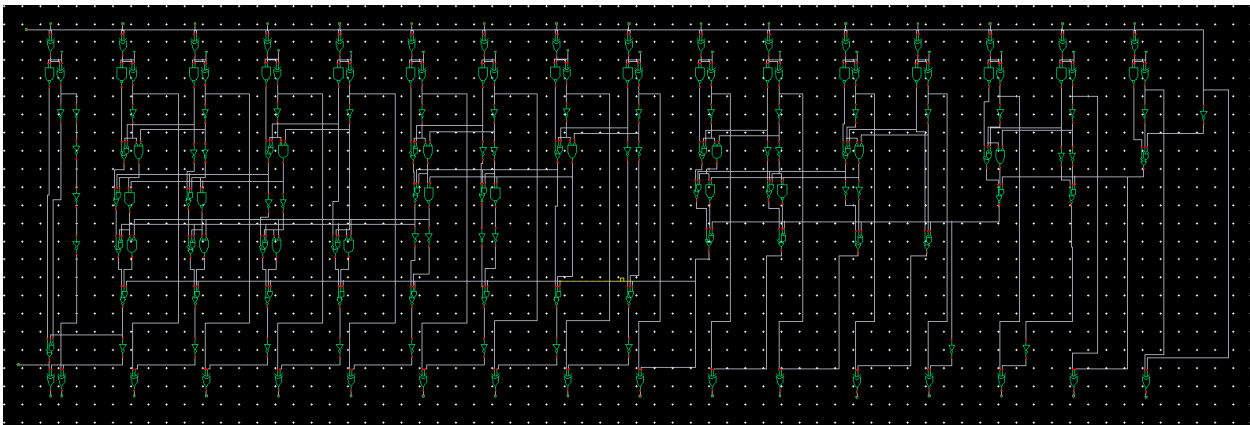


Figure 3.2 Adder Implementation

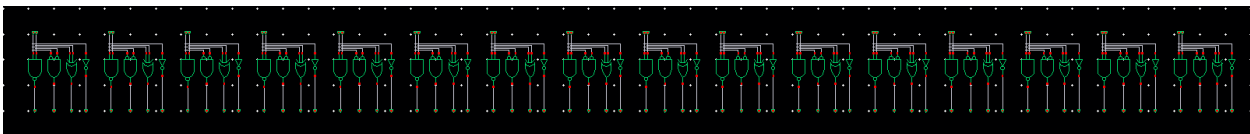


Figure 3.3 And, Or, Xor, Invert operation

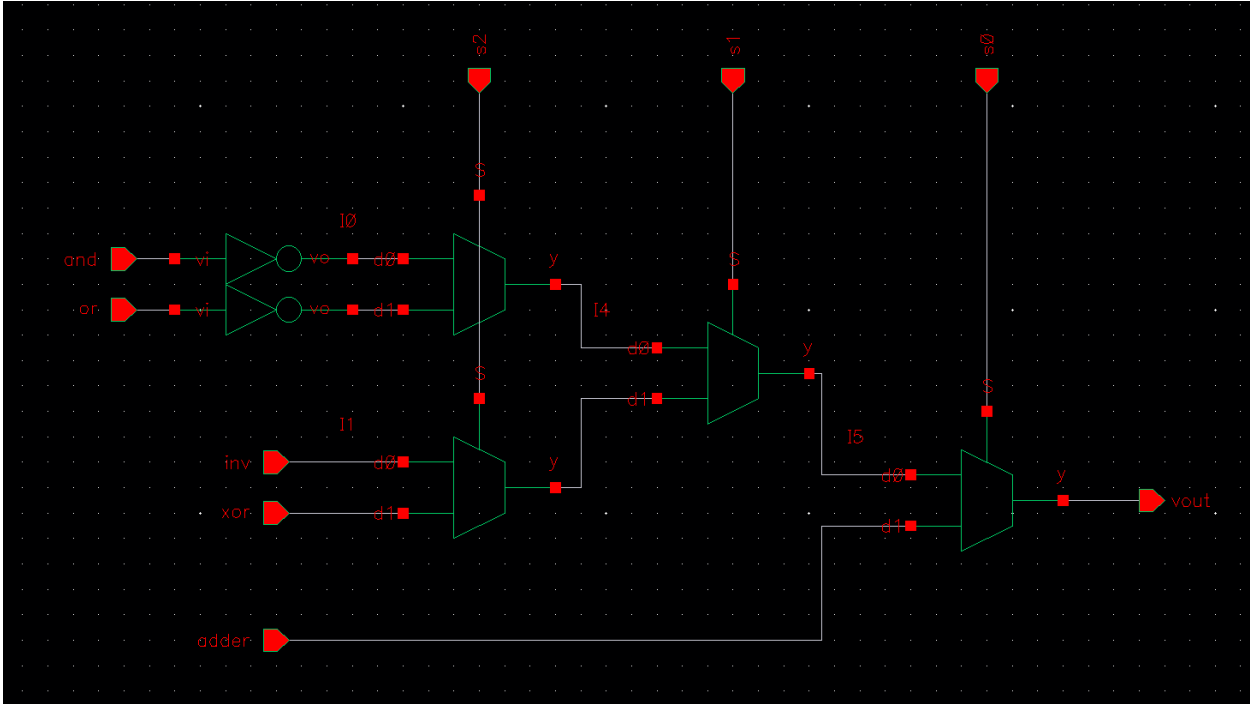


Figure 3.4 Muxi Array for Selecting Operations

4. Layout

Figure 4.1 shows the layout design of the ALU. And it is also DRC and LVS clean.

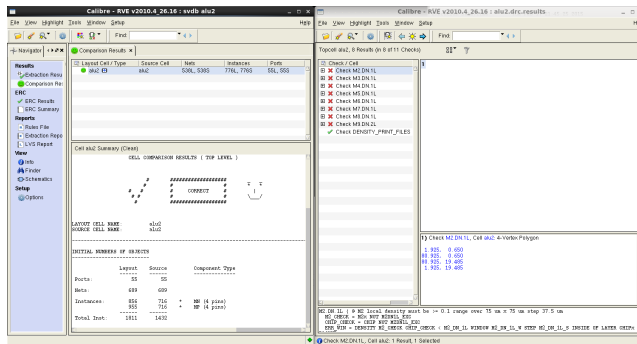
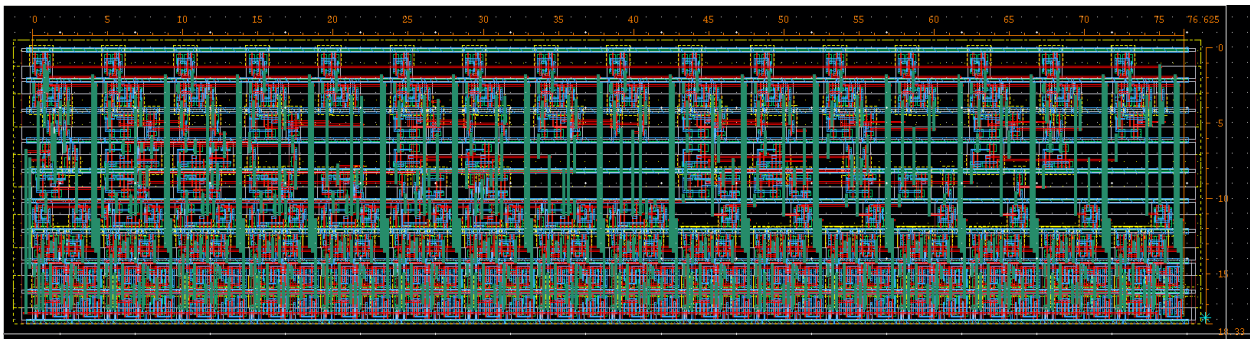


Figure 4.1 - Layout design of the ALU

5. Verification

In order to verify that all the operations are correct. We use a lot of random data as input and test the functionality of all operations. Figure 5.1 shows simulation of the **ADD** operation. Figure 5.2 shows the simulation of **SUBTRACT** operation. Figure 5.3 shows the simulation of **AND** operation. Figure 5.4 shows the simulation of the **OR** operation, Figure 5.5 shows the simulation of the **XOR** operation. Figure 5.6 shows the simulation of **INVERT** operation.

A2D.Rn A2D	16s	1	5276	27731	24432	14290	16229	22829	-1988	23860	5278	-22961	-1498	7857	1358	-32713	13233	12987
A2D.Rm A2D	16s	1	-21615	19540	32666	5502	17864	25401	7199	-7992	27935	2645	17626	27694	-14745	-18722	-19897	23442
A2D.Rd A2D	16s	2	-26891	-18265	-8438	19792	-3443	-17306	5211	15868	-32323	20316	16128	29985	-13387	14101	6664	29107
A2D.OV(over_flow)	1b																	

Figure 5.1 - ADD operation

A2D.Rn A2D	16s	1	5276	27731	24432	14290	16229	22829	-1988	23860	5278	-22961	-1498	7857	1358	-32713	13233	12987
A2D.Rm A2D	16s	1	-21615	19540	32666	5502	17864	25401	7199	-7992	27935	2645	17626	27694	-14745	-18722	-19897	23442
A2D.Rd A2D	16s	0	16339	8191	8234	8788	-1635	-2572	9187	31852	-22657	-25606	-19124	-19837	16103	-13991	32406	-10455

Figure 5.2 - SUBTRACT operation

A2D.Rn A2D	16s	0000000000000001	11101011001000	011011000101001	010111110110000	001101111101000	001111101100101	010110010010101	111110000011100	0101110100110100	000101001001110	101001100100111	11111010001010
A2D.Rm A2D	16s	0000000000000001	101010110010001	010011000101010	011111110011010	00010101111110	010001011001000	0110001100111001	000111000001111	1110000011001000	011011010001111	00001010010101	010001011011010
A2D.Rd A2D	16s	0000000000000001	101010110000000	010111000101010	010111110001000	000101010101010	000001010100000	0110000100101001	000110000001100	010000000000000	000001000001110	100000010100101	010000000000010

Figure 5.3 - AND operation

A2D.Rn A2D	16s	0000000000000001	11101011001000	011011000101001	010111110110000	001101111101000	001111101100101	010110010010101	111110000011100	0101110100110100	000101001001110	101001100100111	11111010001010
A2D.Rm A2D	16s	0000000000000001	101010110010001	010011000101010	011111110011010	00010101111110	010001011001000	0110001100111001	000111000001111	1110000011001000	011011010001111	00001010010101	010001011011010
A2D.Rd A2D	16s	0000000000000001	101010110000000	010111000101010	010111110001000	000101010101010	000001010100000	0110000100101001	000110000001100	010000000000000	000001000001110	100000010100101	010000000000010

Figure 5.4 - OR operation

A2D.Rn A2D	16s	0000000000000001	11101011001000	011011000101001	010111110110000	001101111101000	001111101100101	010110010010101	111110000011100	0101110100110100	000101001001110	101001100100111	11111010001010
A2D.Rm A2D	16s	0000000000000001	101010110010001	010011000101010	011111110011010	00010101111110	010001011001000	0110001100111001	000111000001111	1110000011001000	011011010001111	00001010010101	010001011011010
A2D.Rd A2D	16s	0000000000000001	111010111110101	011011000101011	01111111111010	000101111110101	01111111010101	01110110011101	1111100001111	1111101111100	0111101100111	1010110001111	111111011110

Figure 5.5 - XOR operation

A2D.Rn A2D	16s	0000000000000001	101010110010001	010011000101010	011111110011010	00010101111110	010001011001000	0110001100111001	000111000001111	1110000011001000	011011010001111	00001010010101	010001011011010
A2D.Rd A2D	16s	1111111111111110	010101000110110	101100111010101	1000000001100101	1110101000000101	101101000010111	10011001100010	111000111100000	000111110010111	100100011100000	111101010101010	101101100100101

Figure 5.6 - INVERT operation

6. Measurements and Analysis

The critical path of the ALU schematic circuit is indicated in Figure 6.1. And Figure 6.2 shows the critical path of the ALU layout design.

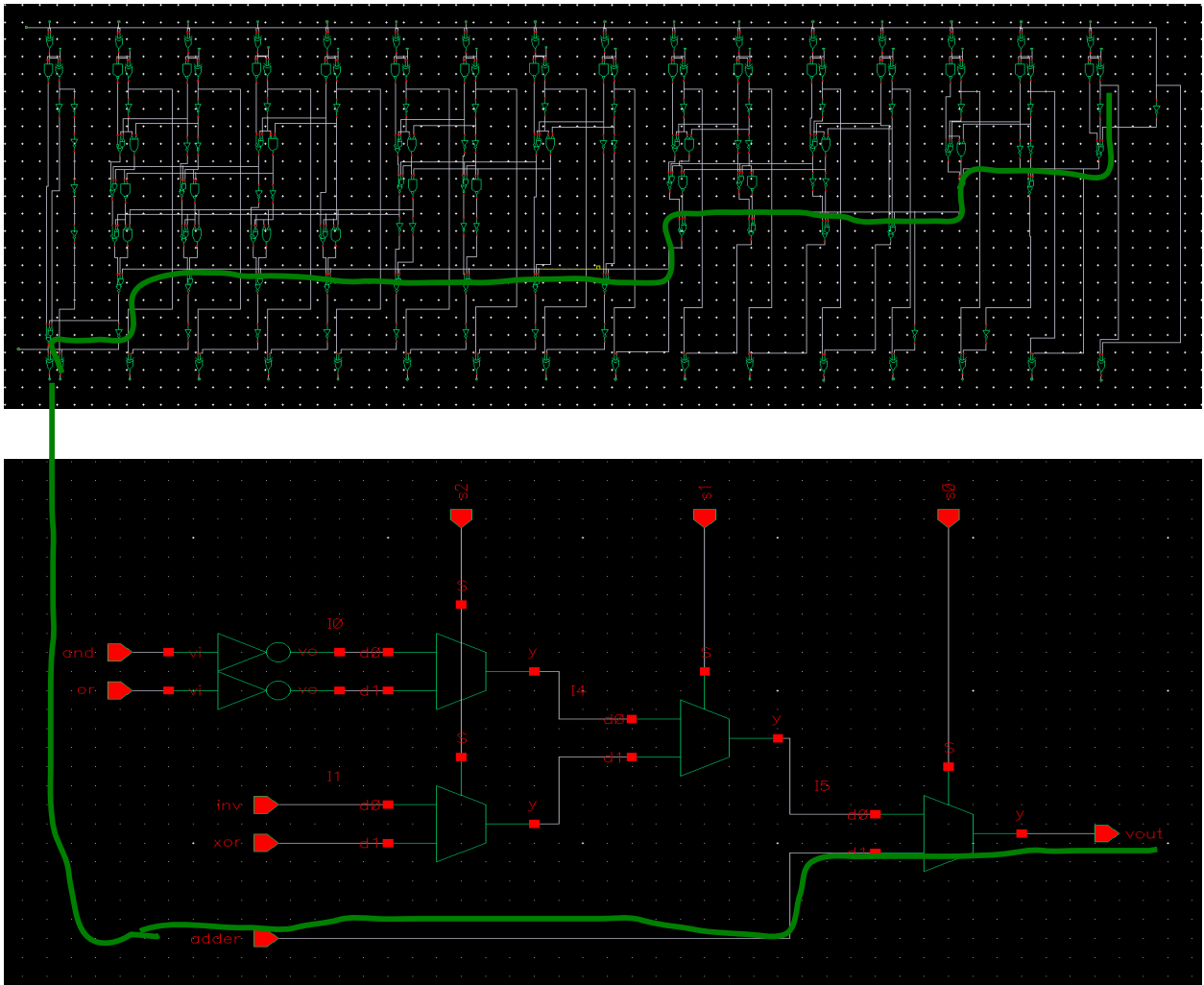


Figure 6.1 - Critical Path of Schematic

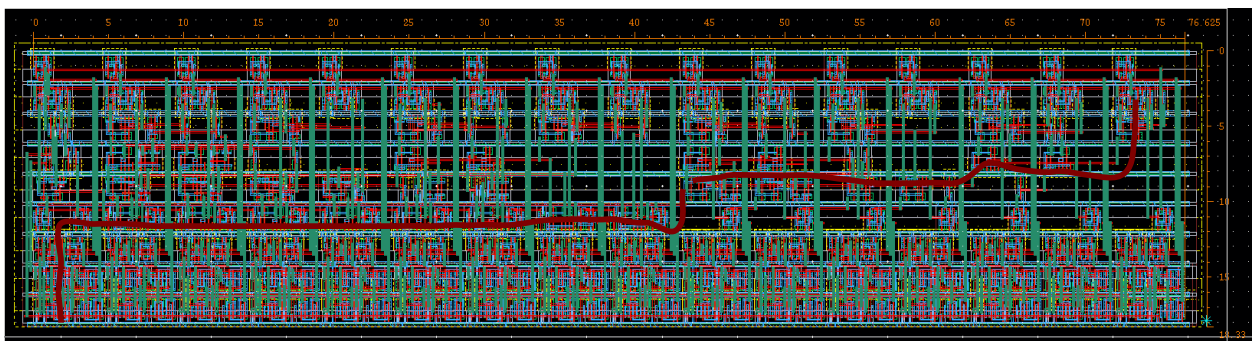


Figure 6.2 - Critical Path of Layout Design

Table 6.1 shows the maximum delay of the critical path.

Table 6.1 - maximum delay

	Delay
Schematic	366 ps
Layout	502 ps

The energy dissipation in performing (**LAYOUT**)

(i): Add 16'hffff with 16'h0001:

Energy: 0.51 pJ

Power: 0.52 mW

(ii): Subtract 16'h0001 from 16'hffff:

Energy: 0.23 fJ

Power: 0.23 mW

Layout Size:

Width: 76.6

Height: 18.3

Area: 1401.8

Aspect-Ratio: 4.2

7. Directory

“/home/projects/ee477/yuxiangc/cad4/results”

Schematic:

Control file: alu.ctl

Netlist file: alu.ckt

Waveform: alu.tr0

Measurements: alu.mt0

Layout:

Control file: alu_layout.ctl

Netlist file: alu_layout.ckt

Simulation: alu_layout.tr0

Pex: alu2.pex.netlist / alu2.pex.netlist.ALU.pxi

drc: /drc

lvs: /lvs