The project description: ALU

It is required to design the ALU shown in Fig.1 using circuit design tools (CAD). This ALUcan execute arithmetic and logical operations. The operation of the ALU is described by table1. The output (arithmetic or logical) is selected by the MSB of the selection line, while the required operation is selected by the other 3 bits. Input (a), input (b), and output (y) are all 4 bits, Control signal (sel) is a 4 bits word.

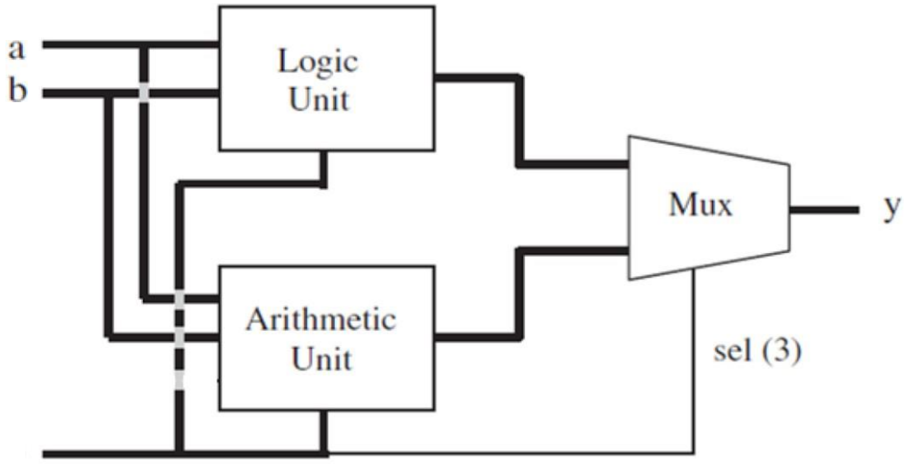


Fig.1

Table 1

Sel	Operation	Unit
0000	Increment a	Arithmetic
0001	Decrement a	
0010	Transfer a	
0011	Increment b	
0100	Decrement b	
0101	Transfer b	
0110	Add a and b	
0111	Subtract a and b (first compare which is smaller and subtract it from the other)	
1000	Complement a	Logic
1001	Complement b	
1010	AND	
1011	OR	
1100	XOR	
1101	XNOR	
1110	NAND	
1111	NOR	

Contents

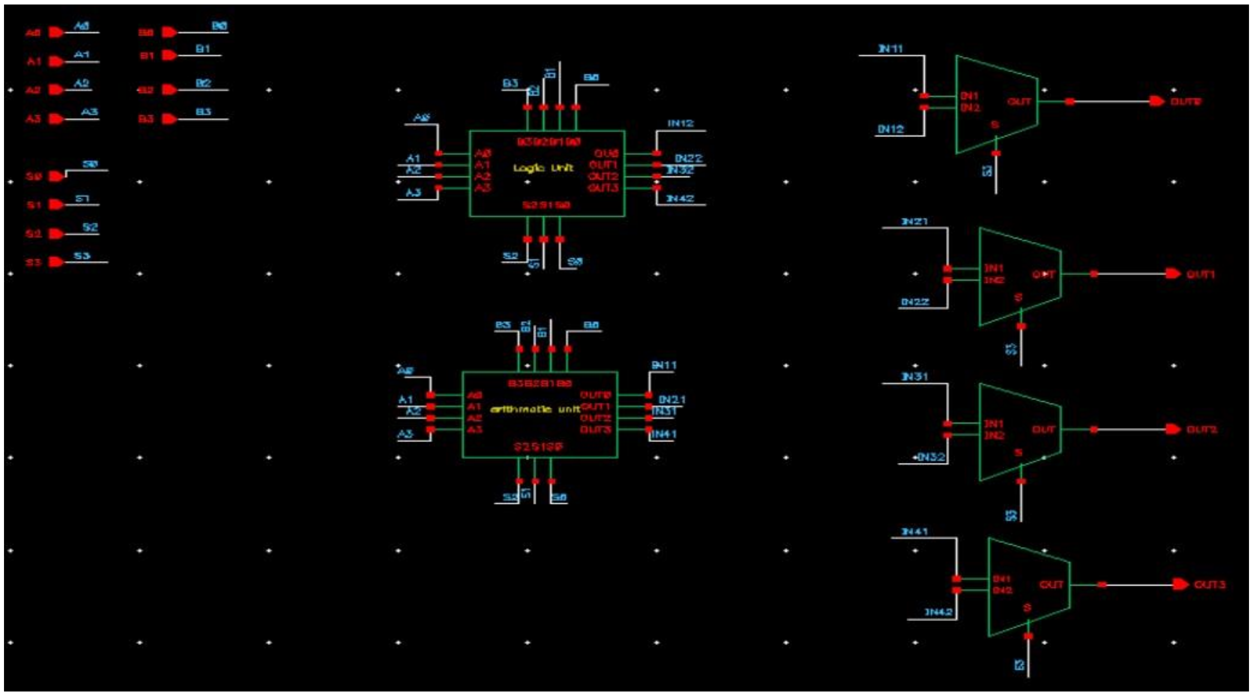
Introduction

An ALU is a combinational logic circuit, meaning that its outputs will change asynchronously in response to input changes. In normal operation, stable signals are applied to all of the ALU inputs and, when enough time (known as the "propagation delay") has passed for the signals to propagate through the ALU circuitry, the result of the ALU operation appears at the ALU outputs. The external circuitry connected to the ALU is responsible for ensuring the stability of ALU input signals throughout the operation, and for allowing sufficient time for the signals to propagate through the ALU before sampling the ALU result.

In general, external circuitry controls an ALU by applying signals to its inputs. Typically, the external circuitry employs sequential logic to control the ALU operation, which is paced by a clock signal of a sufficiently low frequency to ensure enough time for the ALU outputs to settle under worst-case conditions.

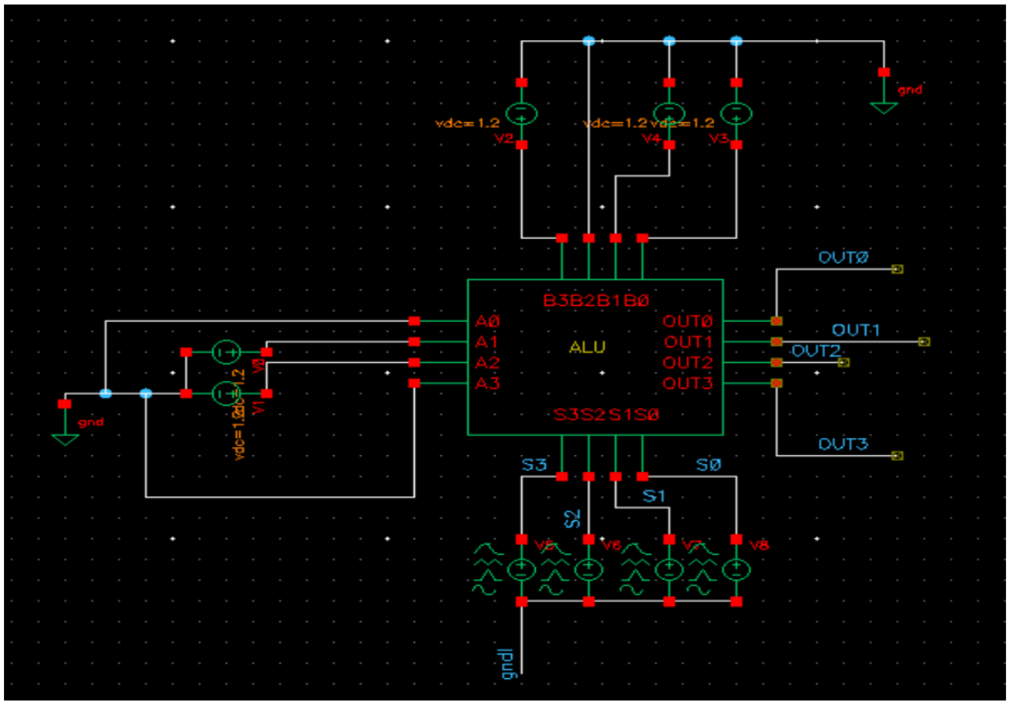
4Bit ALU Design

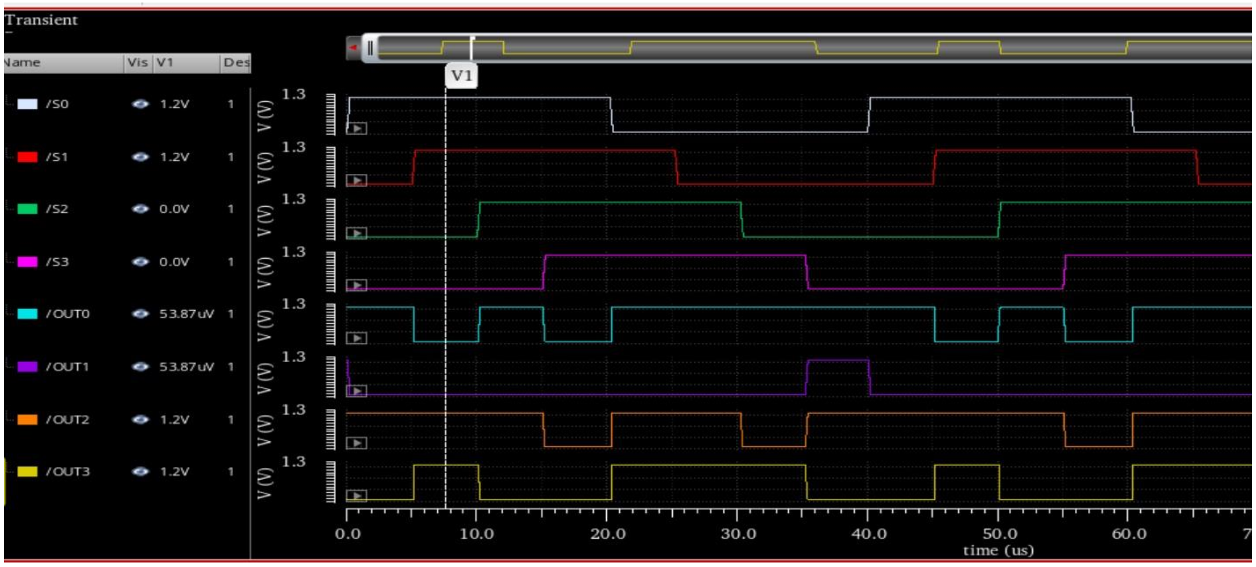
The ALU consists of 2 blocks the Logic unit and the Arithmetic unit the MUXs choose between the output of each block to be the output of the ALU there are 4 MUXs each responsible for one bit of the 4 bit output of the ALU



ALU Test bench

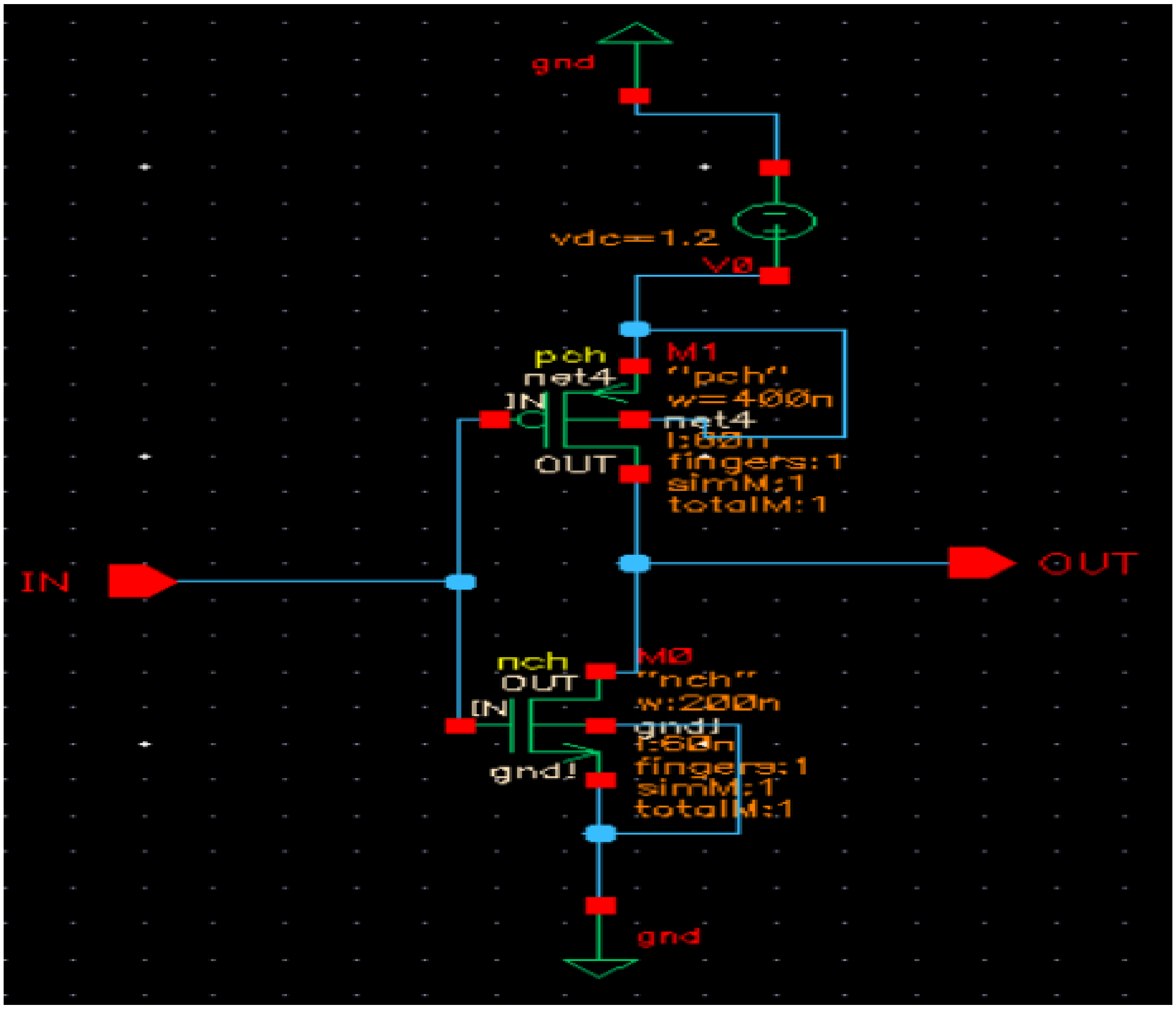
The below testbench tests different outputs of the ALU when the inputs are A= 0110 and B=1011 while sweeping the select pins inputs



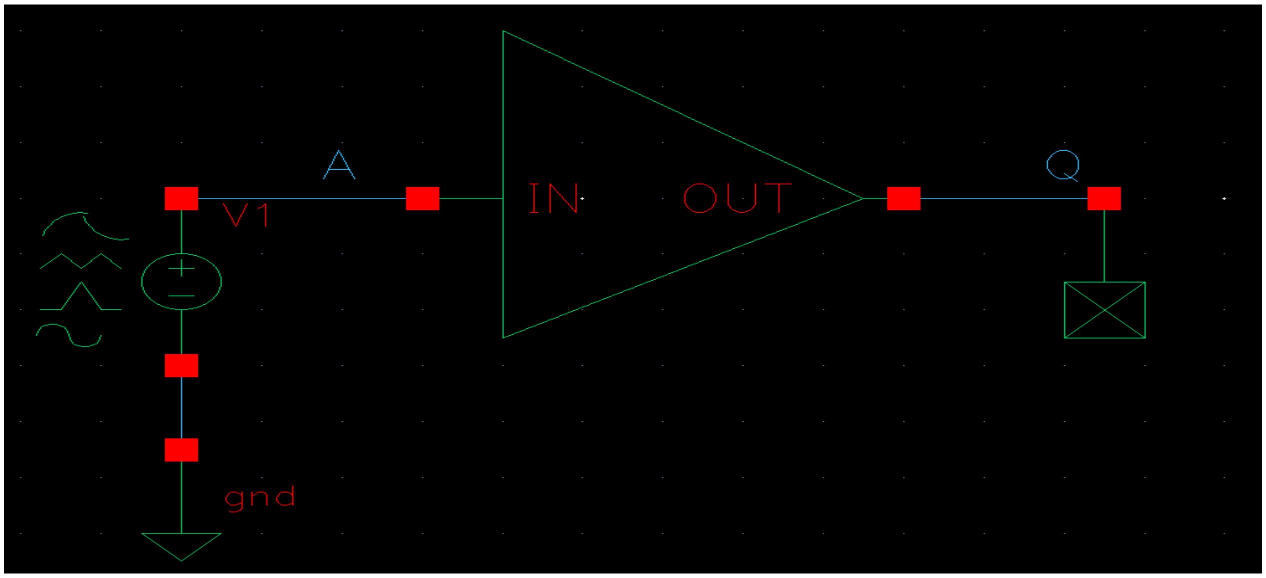


Implementation of gates and MUX

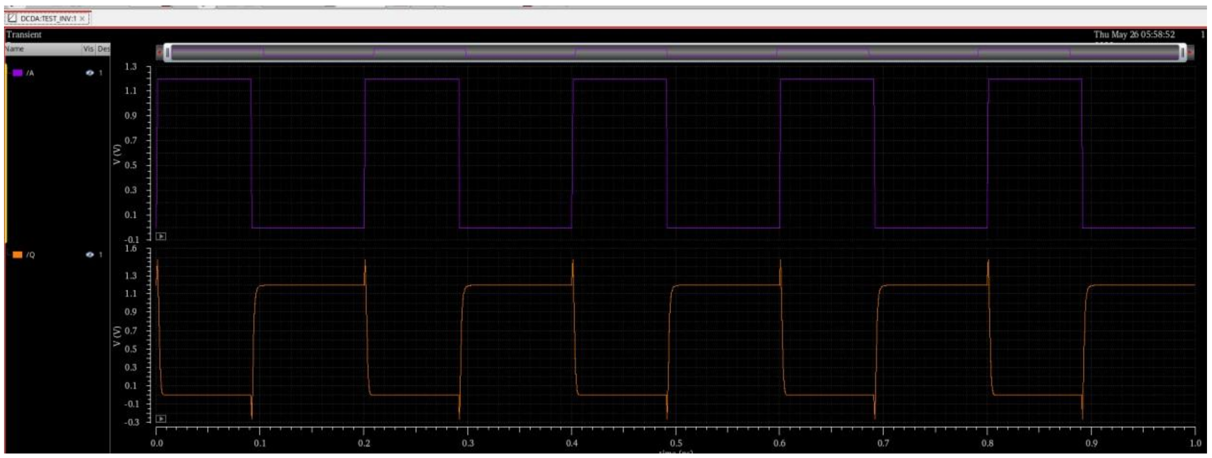
Complement a or b (inverter)



Inverter Design

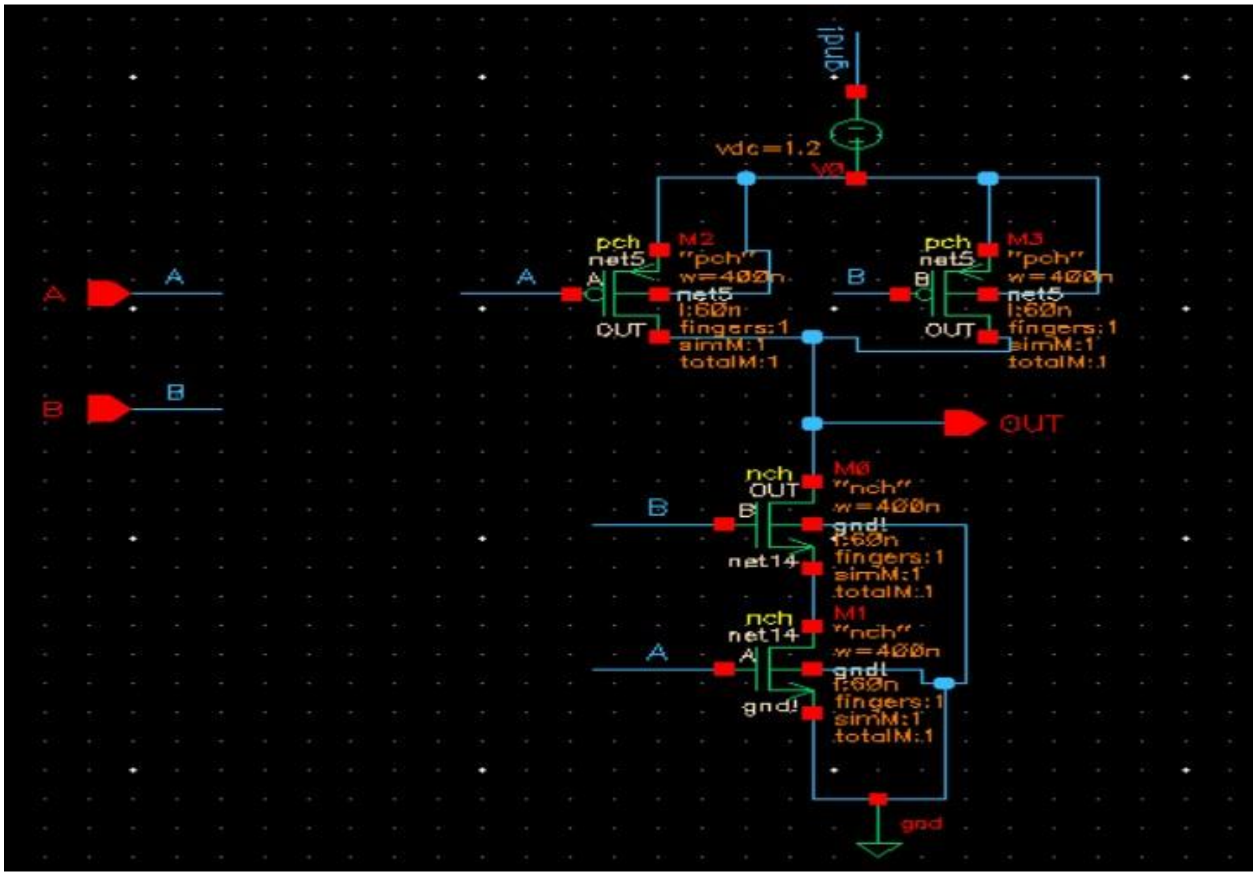


Inverter symbol

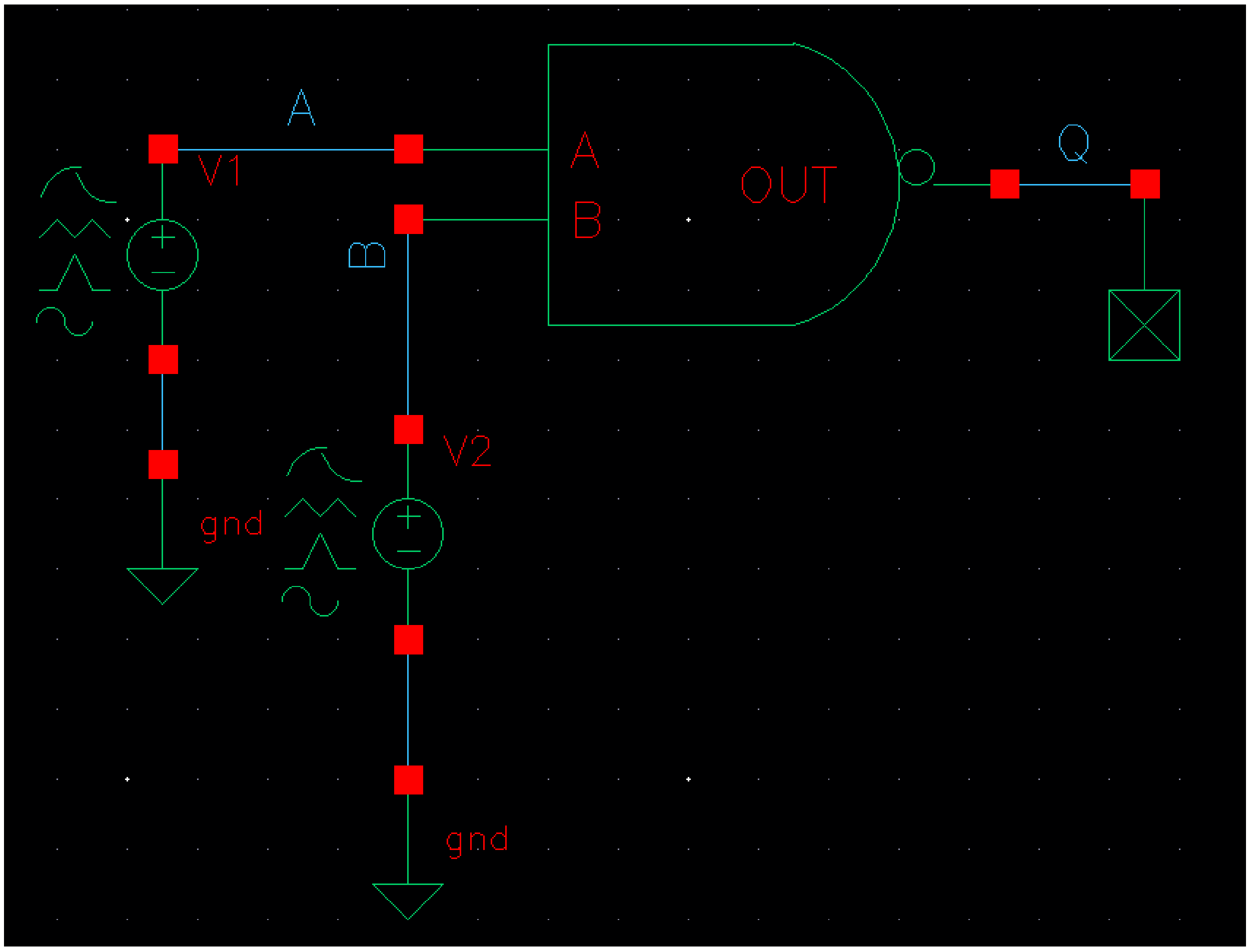


Inverter TestBench

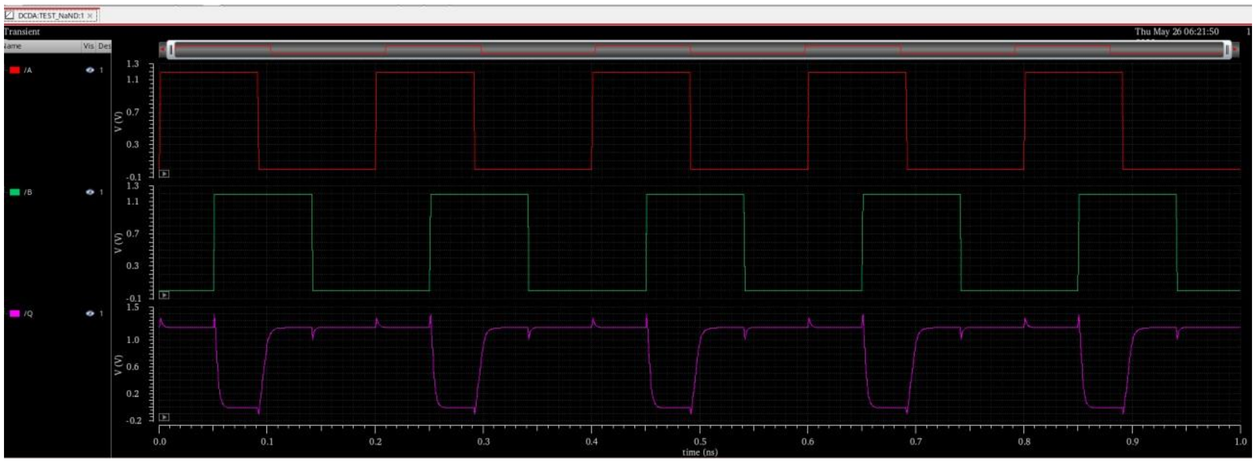
NAND



NAND Design

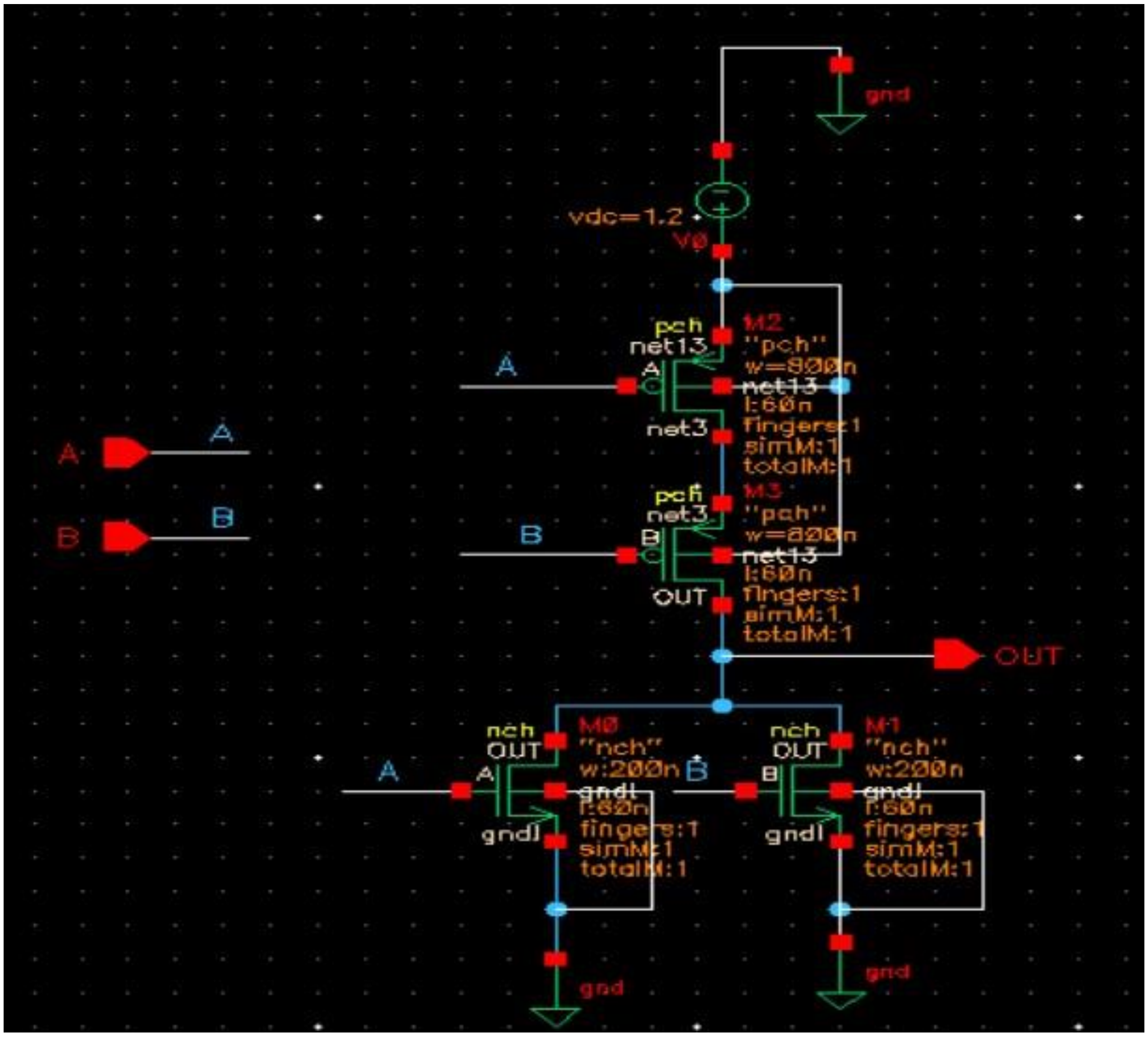


NAND Symbol

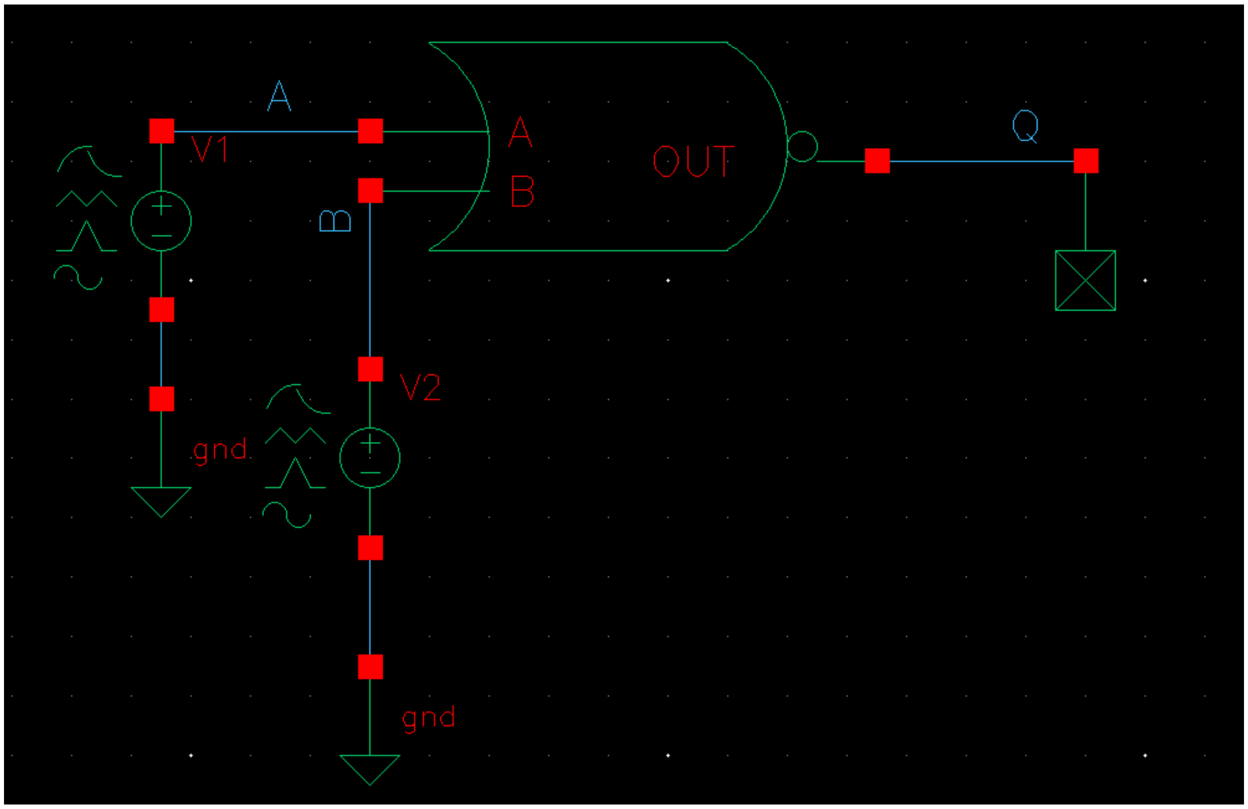


NAND TestBench

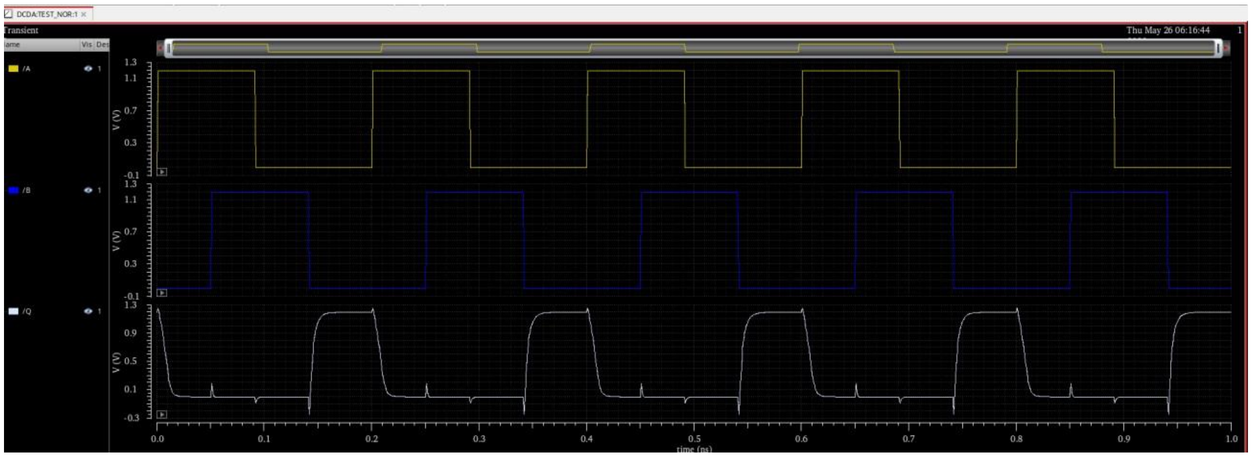
NOR



NOR design

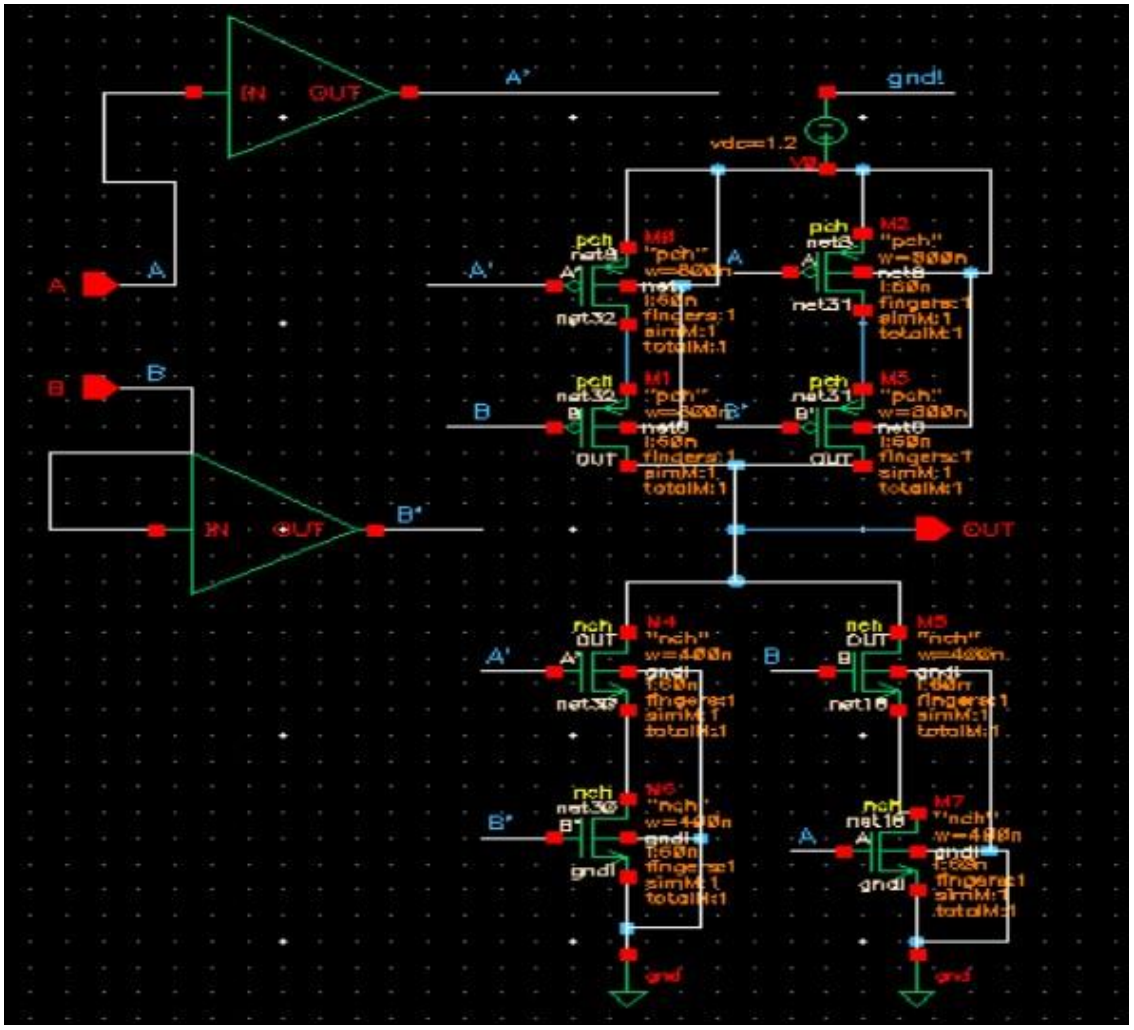


NOR Symbol

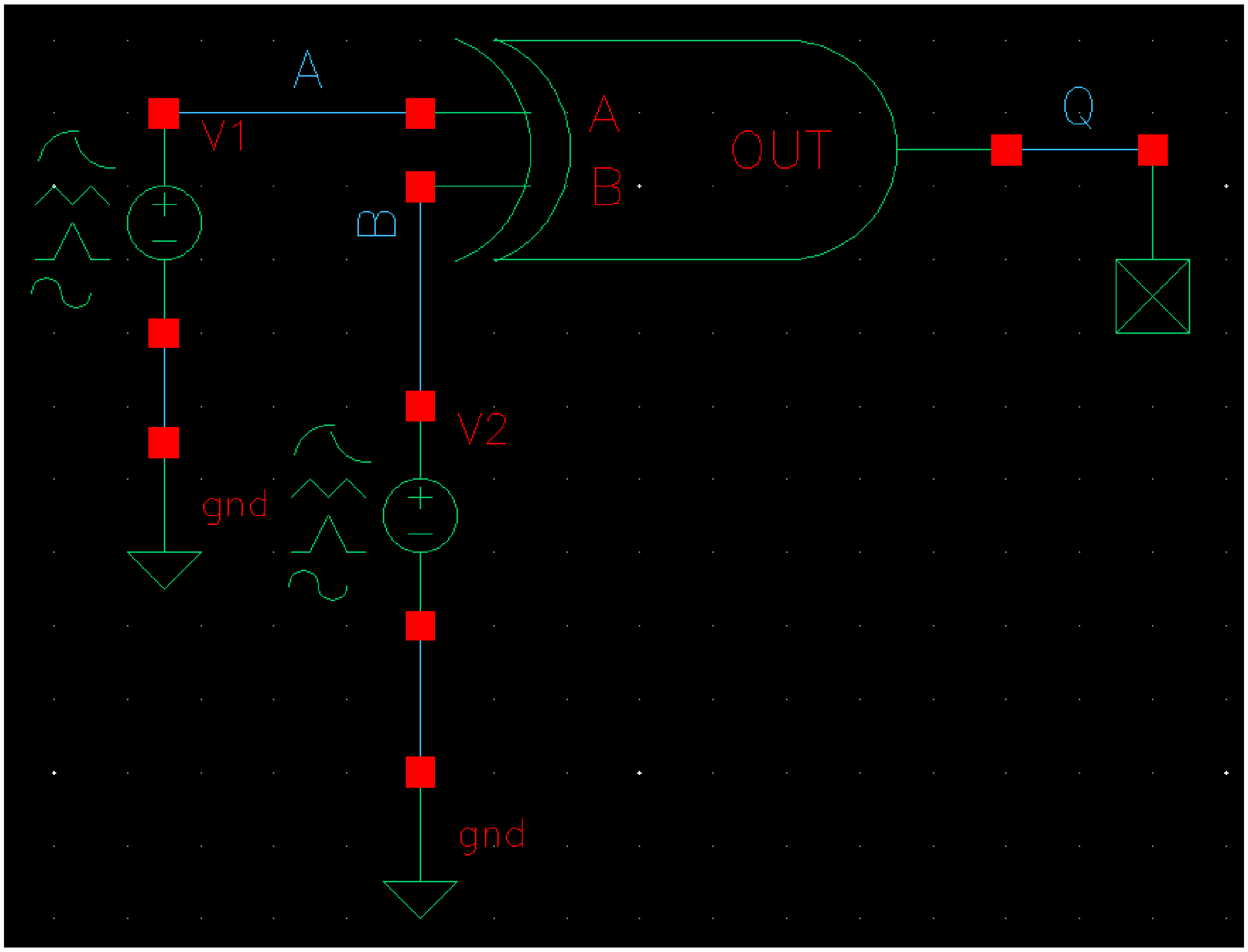


NOR TestBench

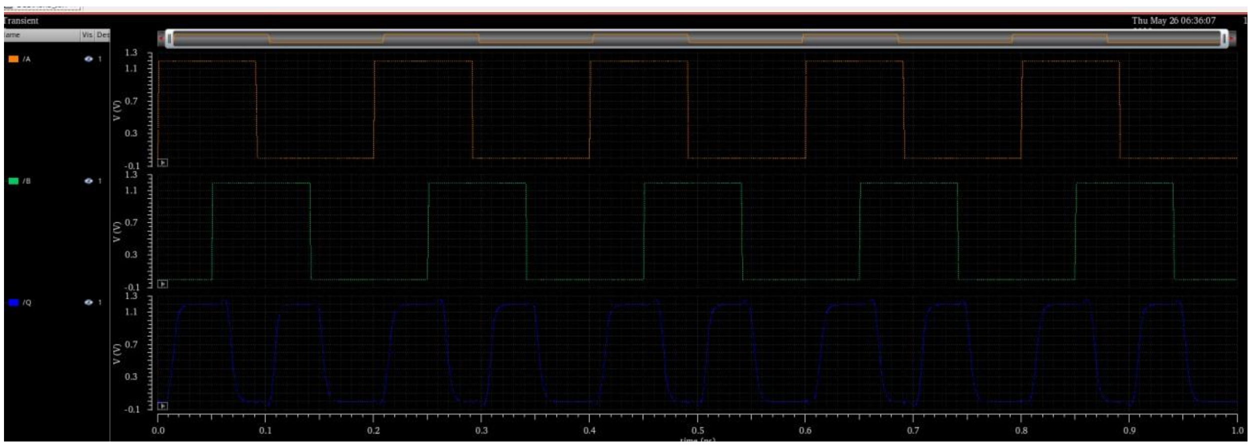
XOR



XOR Design

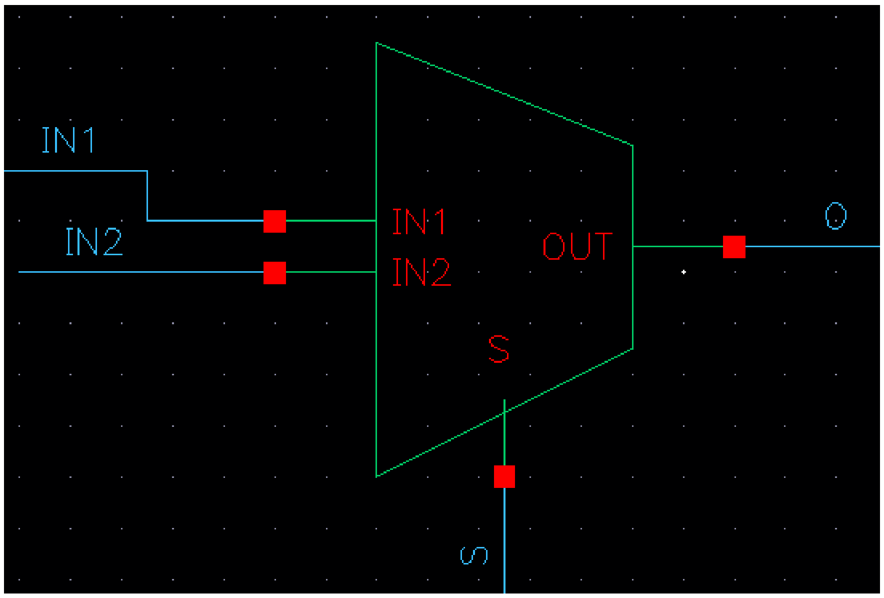


XOR Symbol

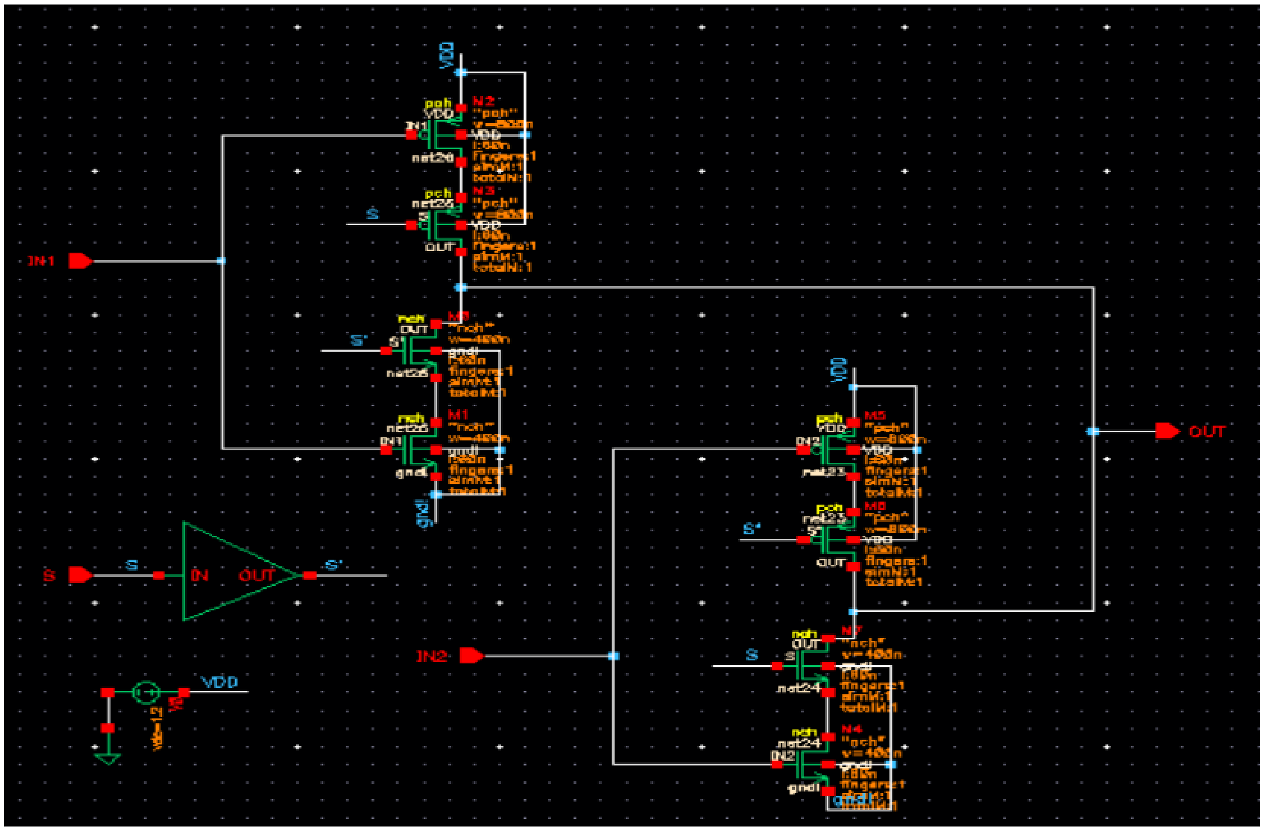


XOR TestBench

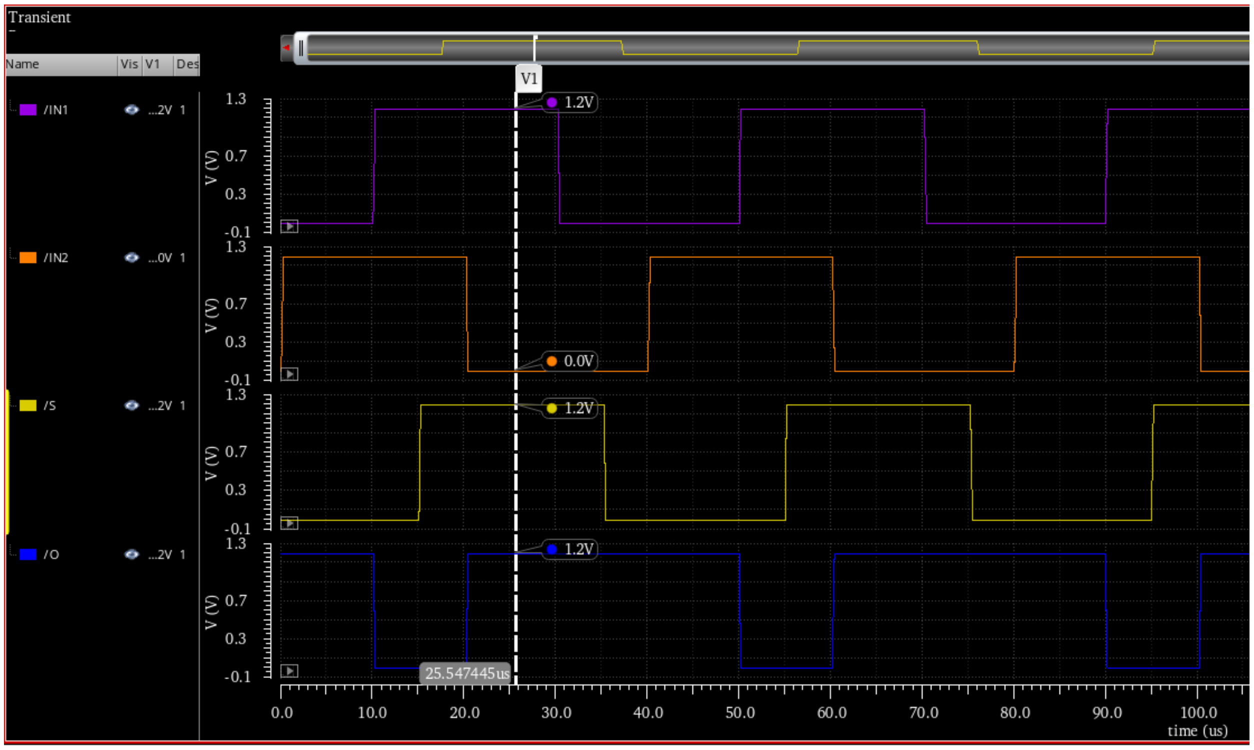
2 v 1 mux (INVERTED)



2x1 MUX Symbol

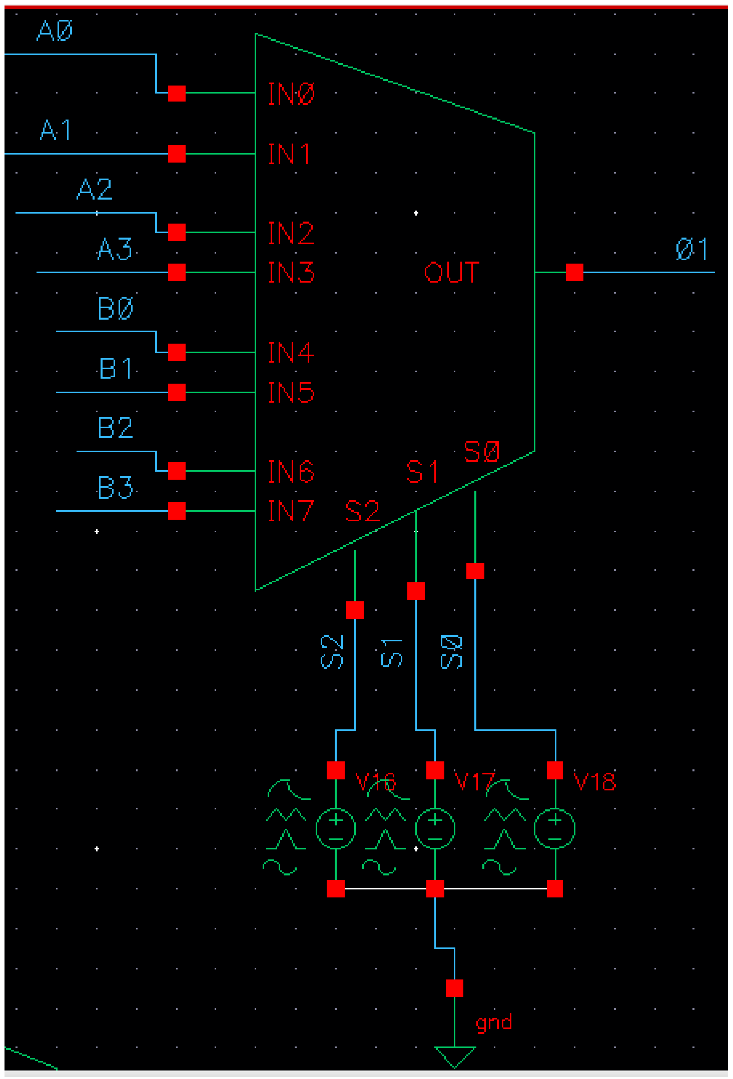


2x1 MUX design

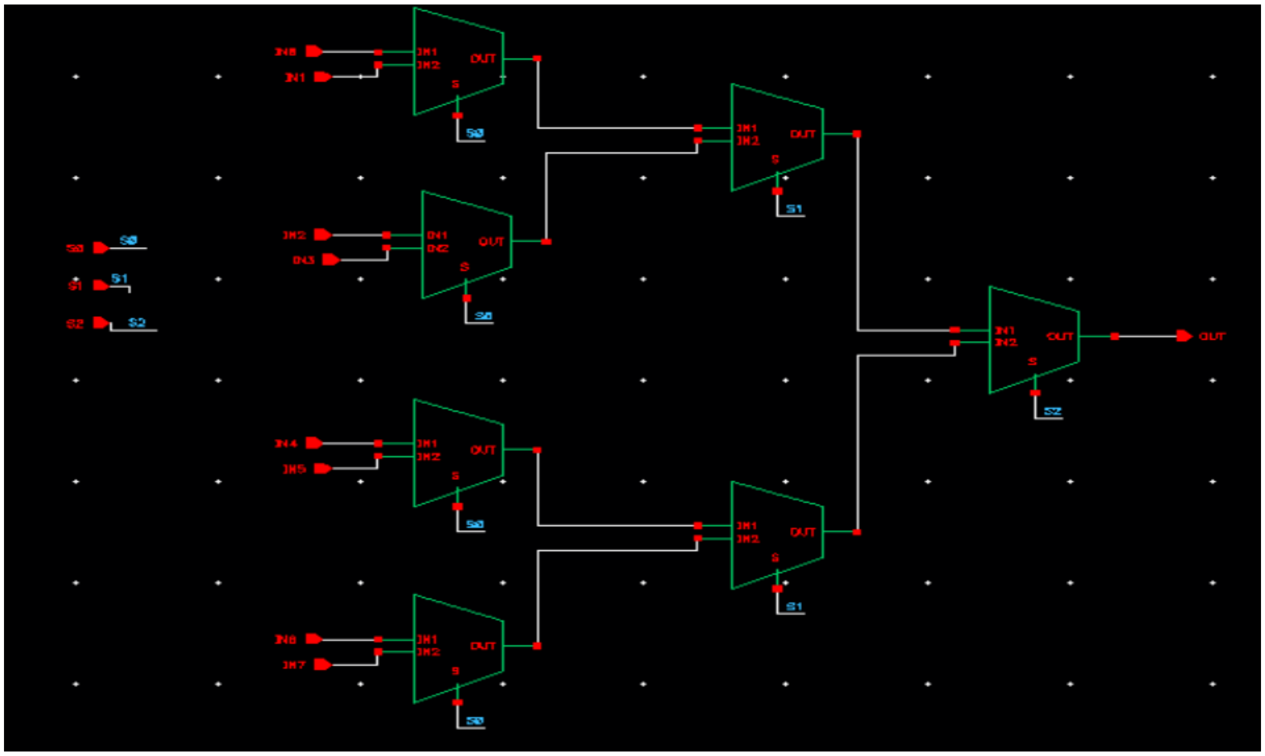


2x1 MUX TestBench

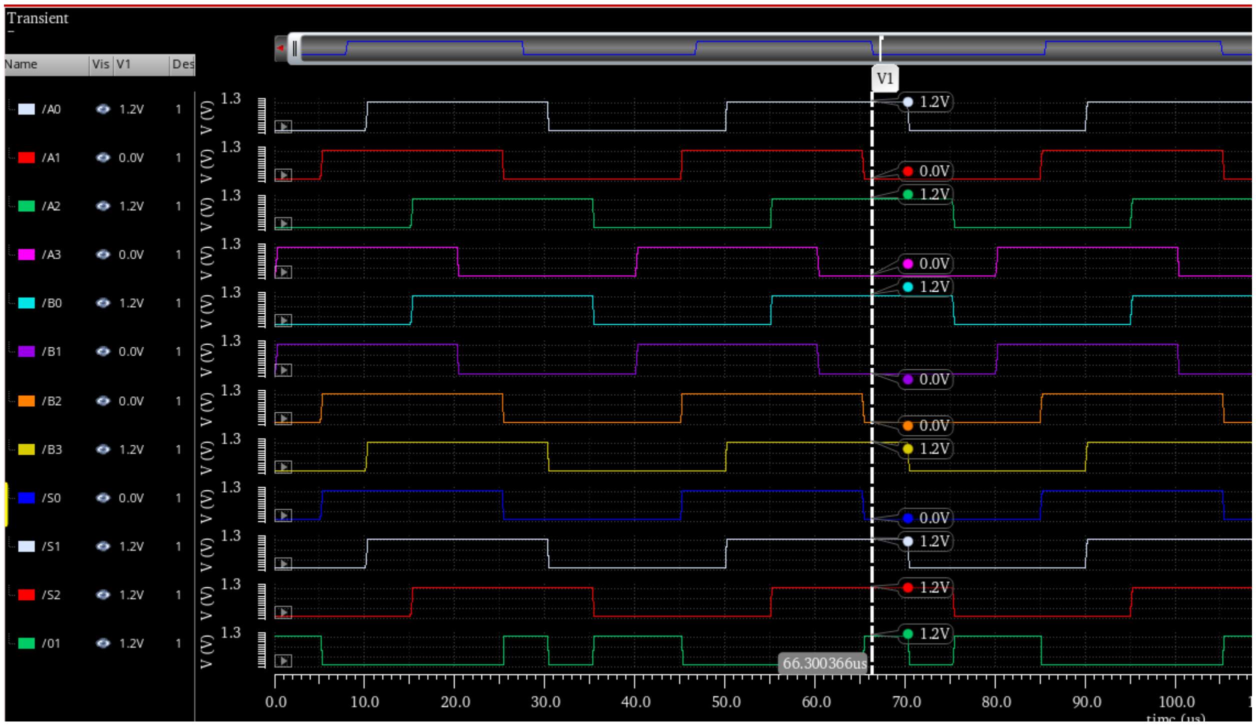
8 v 1 mux (INVERTED)



8X1 MUX Symbol



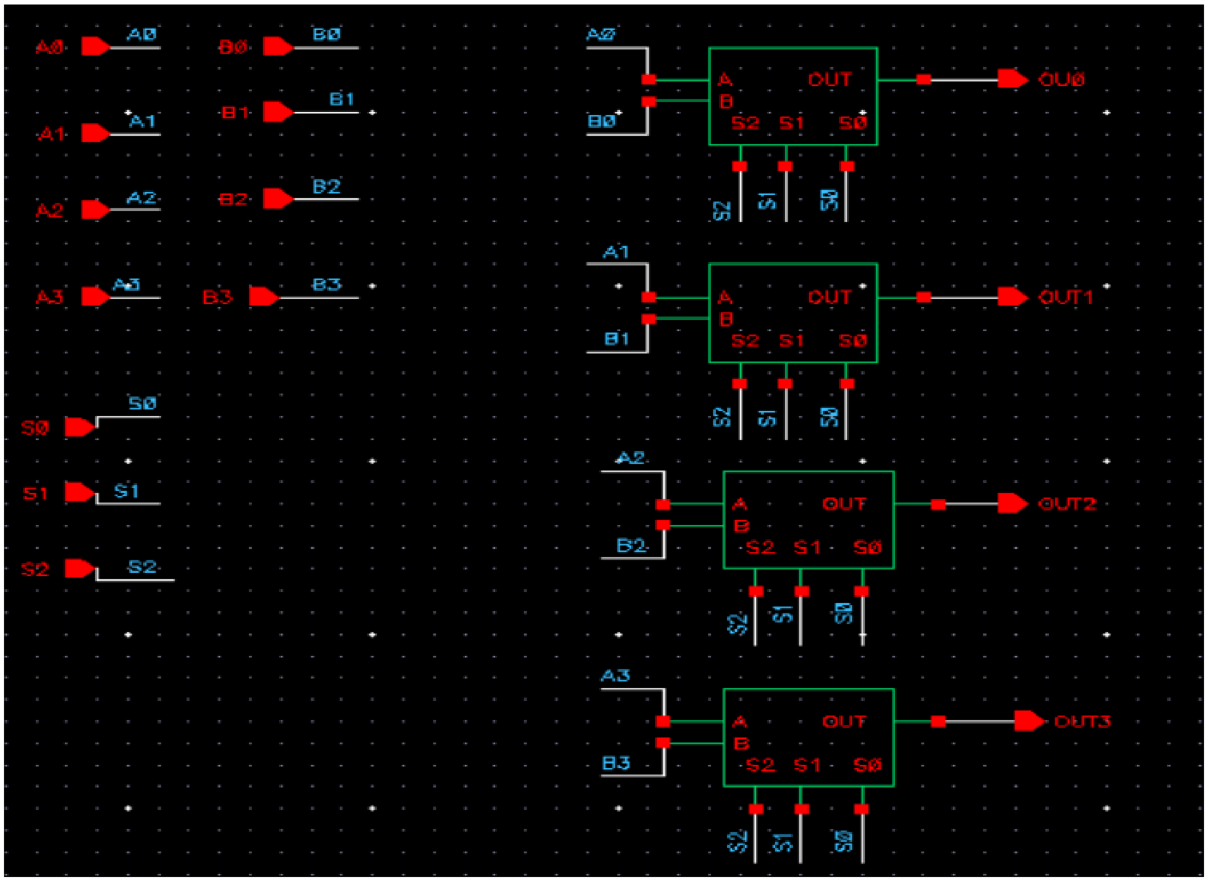
8X1 MUX Design



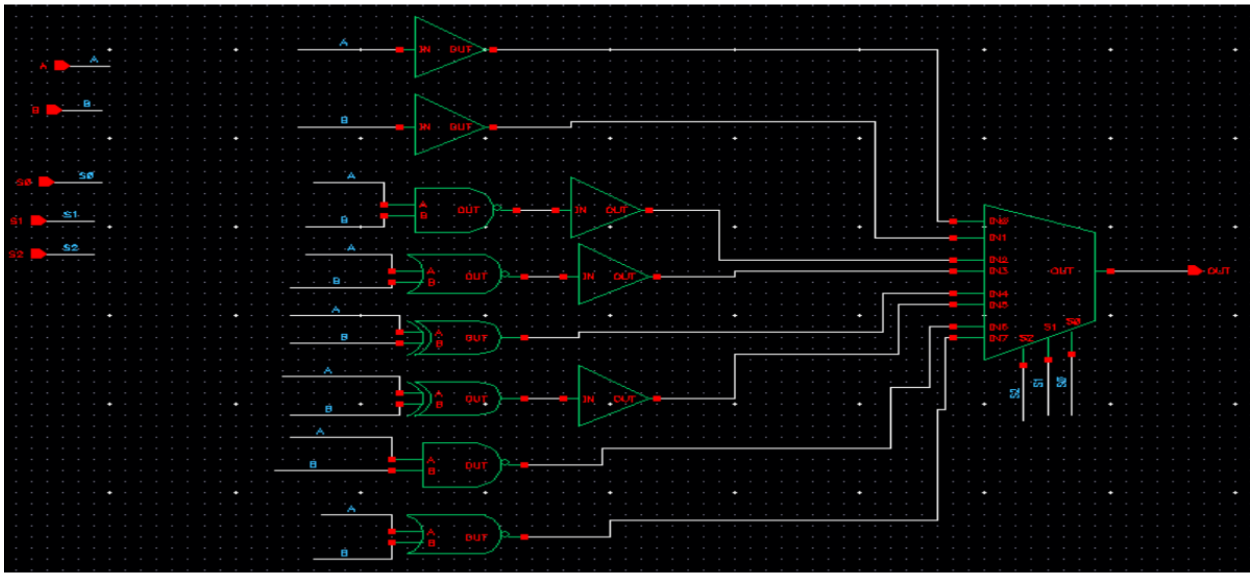
8X1 MUX TestBench

Logic unit (INVERTED)

The Logic Unit consists of four blocks each is responsible for outputting 1 bit of the 4 bits of the output, the output of the blocks is inverted due to the use of inverted muxes but when the Logic unit output is connected to the muxes of the ALU it is inverted again so the output of the ALU is the true value



Logic unit Blocks

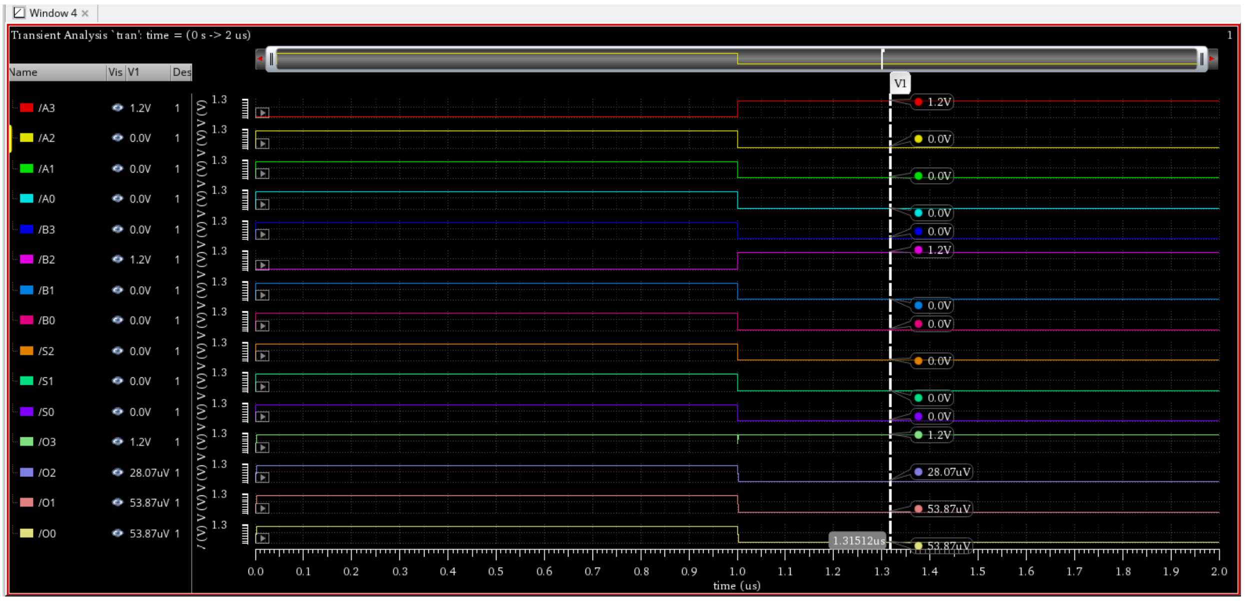


Logic unit Block components

TestBench of Logic Unit

All the outputs of the Logic unit are inverted due to the use of inverter logic MUXs this inversion is removed by the last layer of MUXs which is in the ALU

COMPLEMENT A

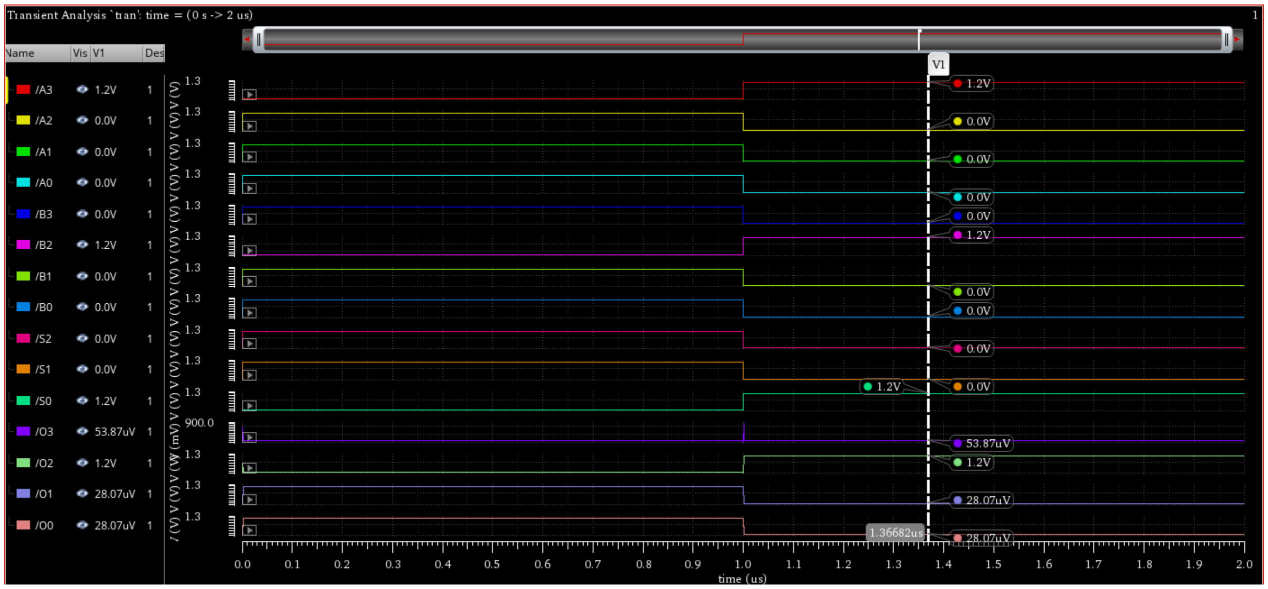


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		1.2V	1
/A2		0.0V	1
/A1		0.0V	1
/A0		0.0V	1
/B3		0.0V	1
/B2		1.2V	1
/B1		0.0V	1
/B0		0.0V	1
/S2		0.0V	1
/S1		0.0V	1
/S0		0.0V	1
/O3		1.2V	1
/O2		28.07uV	1
/O1		53.87uV	1
/O0		53.87uV	1

Inverter TestBench

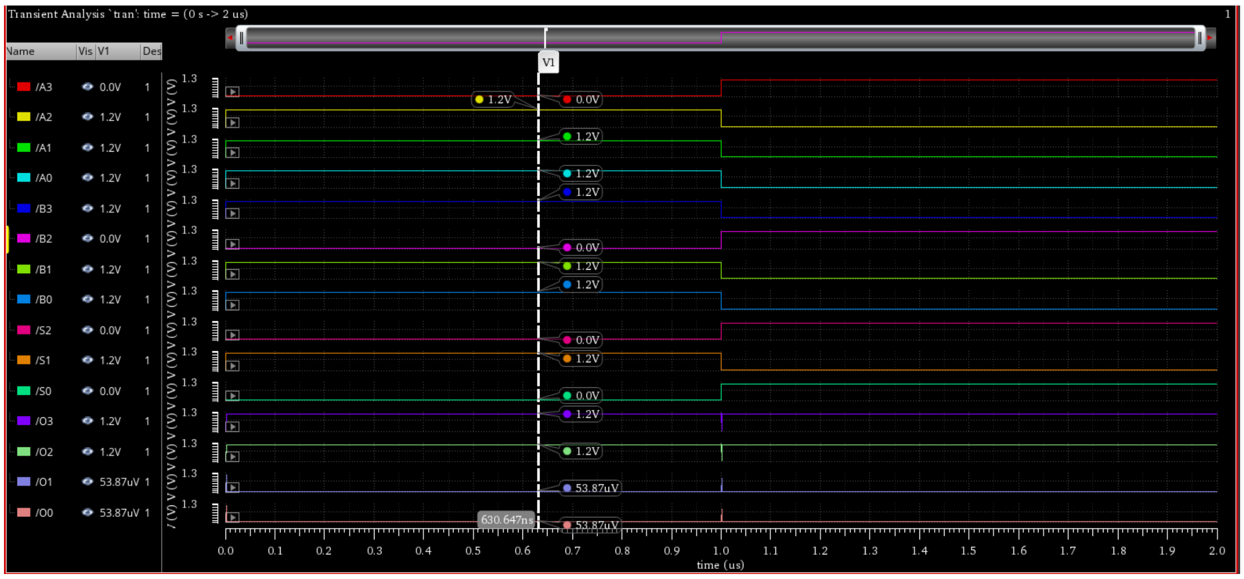
COMPLEMENT B


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		1.2V	1
/A2		0.0V	1
/A1		0.0V	1
/A0		0.0V	1
/B3		0.0V	1
/B2		1.2V	1
/B1		0.0V	1
/B0		0.0V	1
/S2		0.0V	1
/S1		0.0V	1
/S0		1.2V	1
/O3		53.87uV	1
/O2		1.2V	1
/O1		28.07uV	1
/O0		28.07uV	1

Inverter TestBench

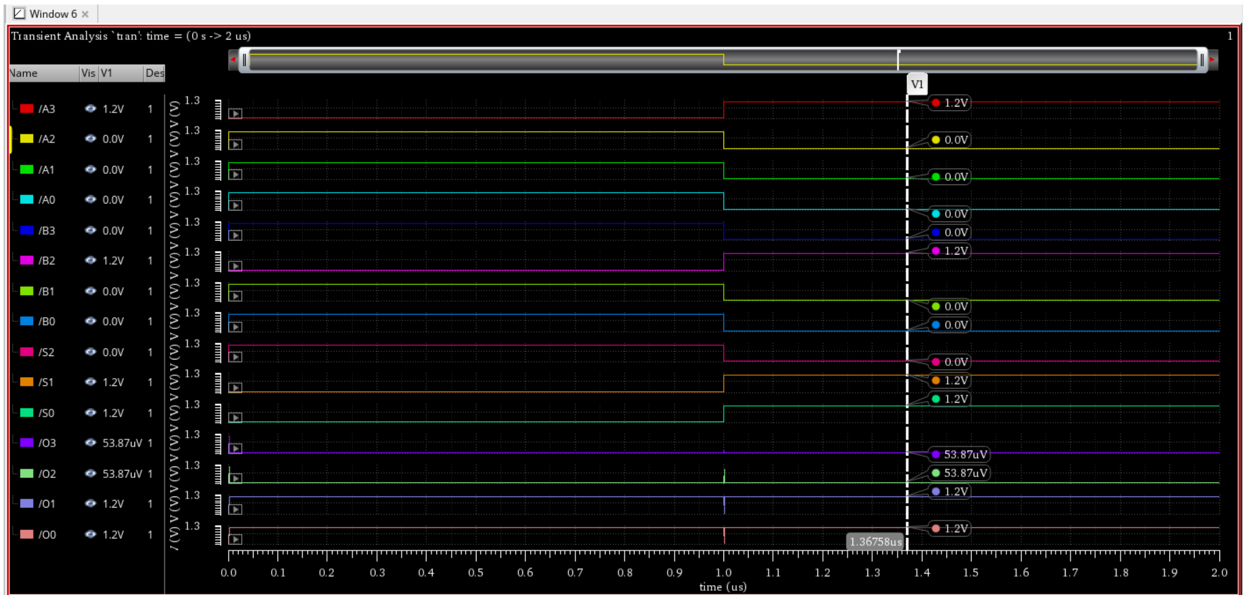
AND


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		0.0V	1
/A2		1.2V	1
/A1		1.2V	1
/A0		1.2V	1
/B3		1.2V	1
/B2		0.0V	1
/B1		1.2V	1
/B0		1.2V	1
/S2		0.0V	1
/S1		1.2V	1
/S0		0.0V	1
/O3		1.2V	1
/O2		1.2V	1
/O1		53.87uV	1
/O0		53.87uV	1

AND TestBench

OR

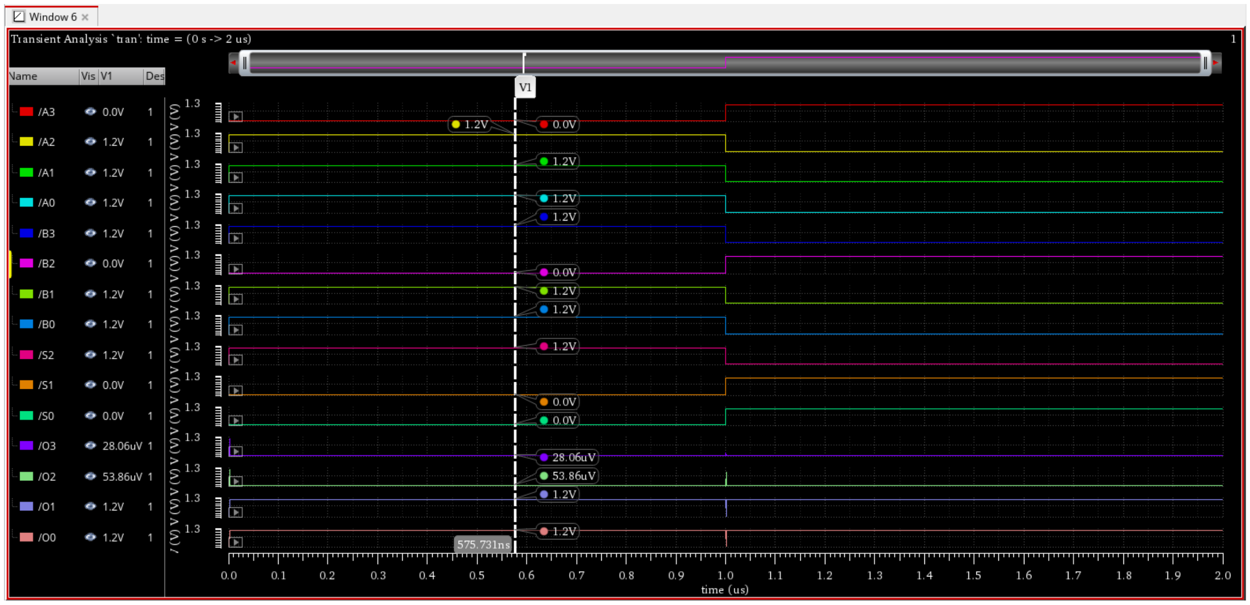


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		1.2V	1
/A2		0.0V	1
/A1		0.0V	1
/A0		0.0V	1
/B3		0.0V	1
/B2		1.2V	1
/B1		0.0V	1
/B0		0.0V	1
/S2		0.0V	1
/S1		1.2V	1
/S0		1.2V	1
/O3		53.87uV	1
/O2		53.87uV	1
/O1		1.2V	1
/O0		1.2V	1

OR TestBench

XOR

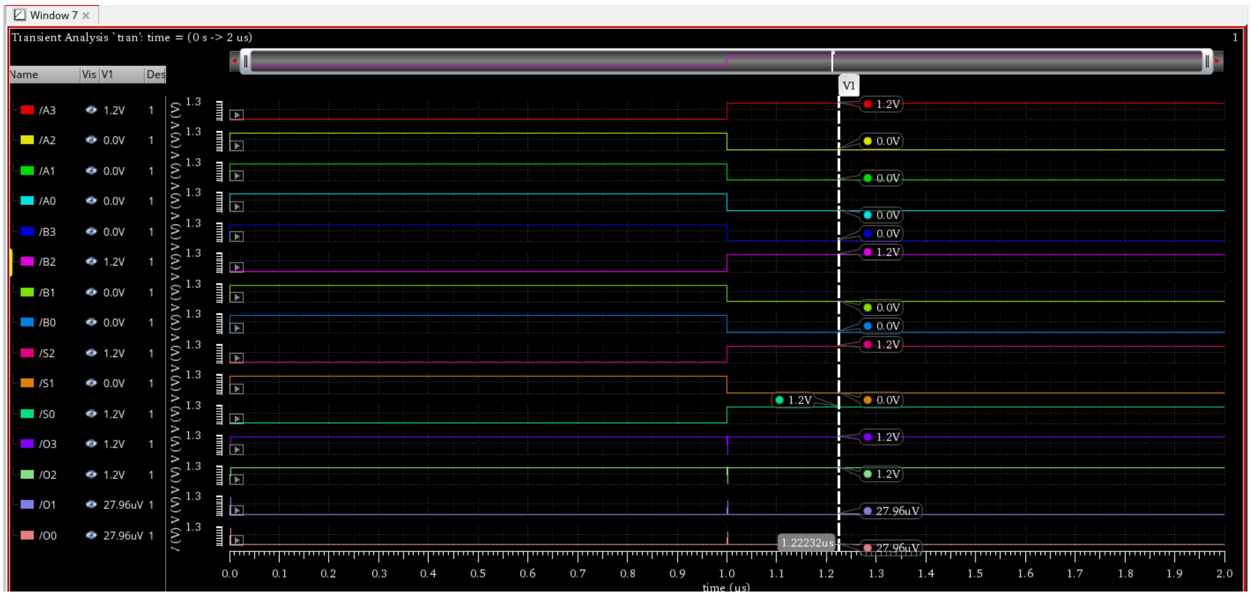


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		0.0V	1
/A2		1.2V	1
/A1		1.2V	1
/A0		1.2V	1
/B3		1.2V	1
/B2		0.0V	1
/B1		1.2V	1
/B0		1.2V	1
/S2		1.2V	1
/S1		0.0V	1
/S0		0.0V	1
/O3		28.06uV	1
/O2		53.86uV	1
/O1		1.2V	1
/O0		1.2V	1

XOR TestBench

XNOR

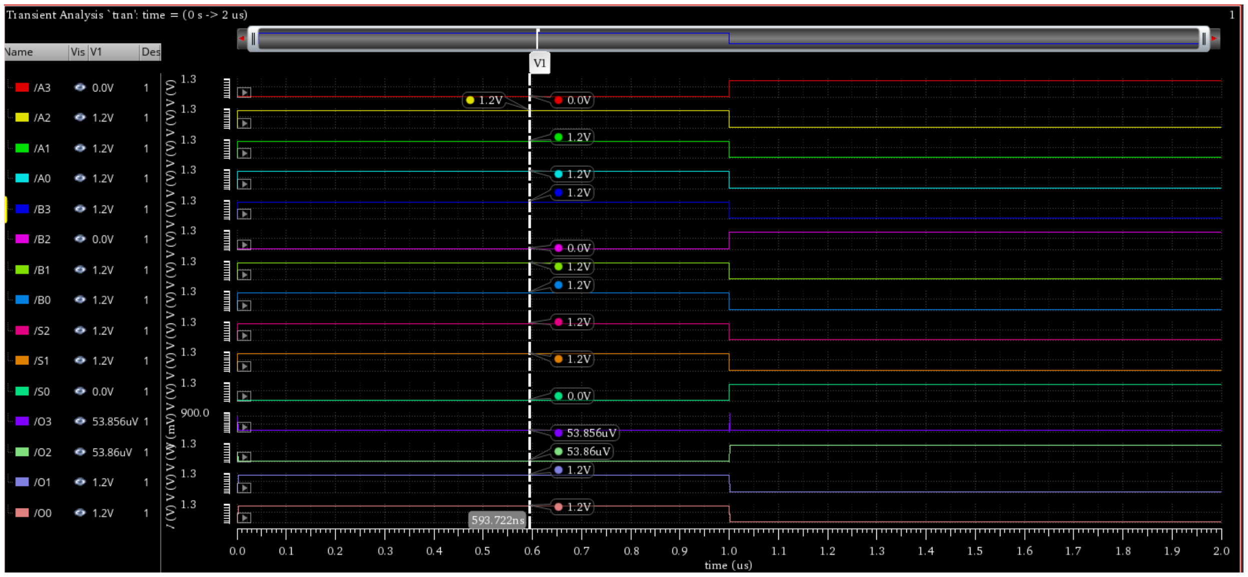

Window 7

Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3	👁	1.2V	1
/A2	👁	0.0V	1
/A1	👁	0.0V	1
/A0	👁	0.0V	1
/B3	👁	0.0V	1
/B2	👁	1.2V	1
/B1	👁	0.0V	1
/B0	👁	0.0V	1
/S2	👁	1.2V	1
/S1	👁	0.0V	1
/S0	👁	1.2V	1
/O3	👁	1.2V	1
/O2	👁	1.2V	1
/O1	👁	27.96uV	1
/O0	👁	27.96uV	1

XNOR TestBench

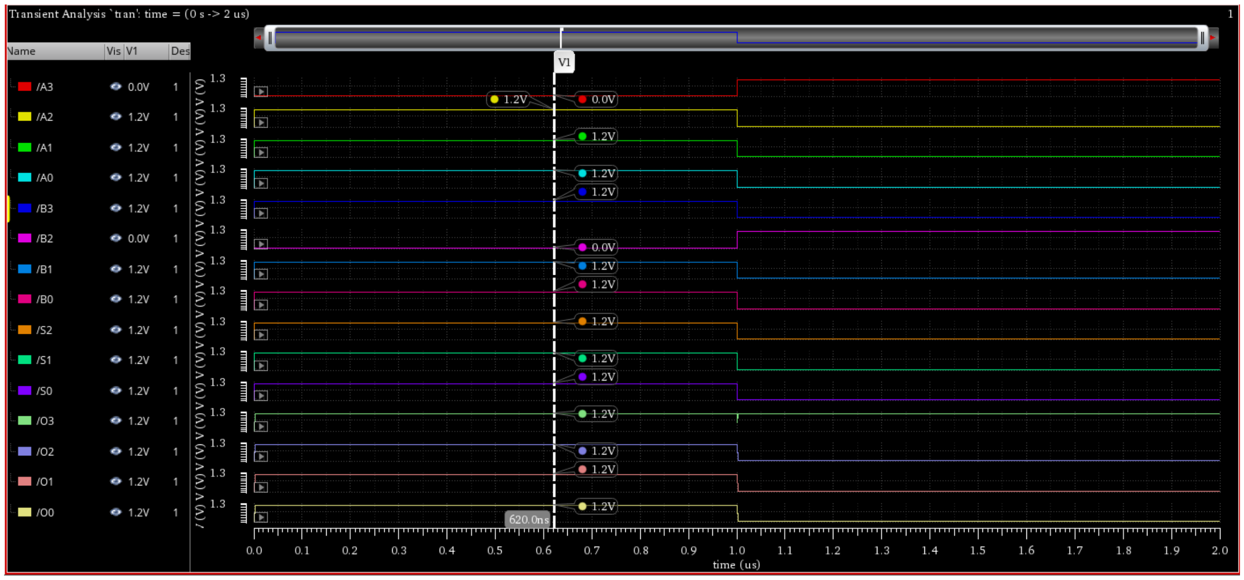
NAND


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3	👁	0.0V	1
/A2	👁	1.2V	1
/A1	👁	1.2V	1
/A0	👁	1.2V	1
/B3	👁	1.2V	1
/B2	👁	0.0V	1
/B1	👁	1.2V	1
/B0	👁	1.2V	1
/S2	👁	1.2V	1
/S1	👁	1.2V	1
/S0	👁	0.0V	1
/O3	👁	53.856uV	1
/O2	👁	53.86uV	1
/O1	👁	1.2V	1
/O0	👁	1.2V	1

NAND TestBench

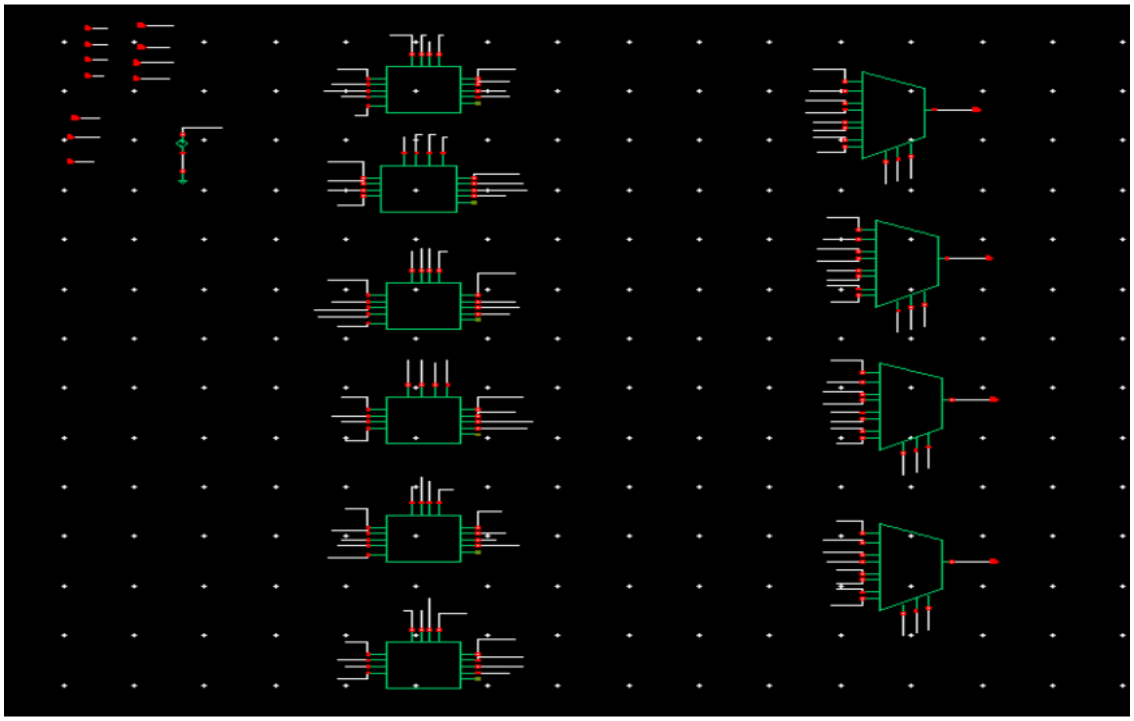
NOR

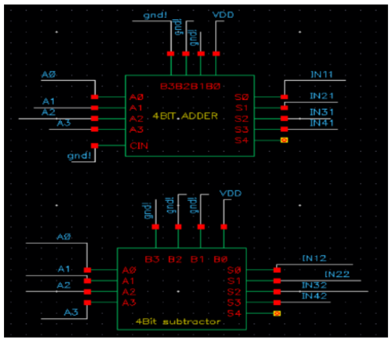


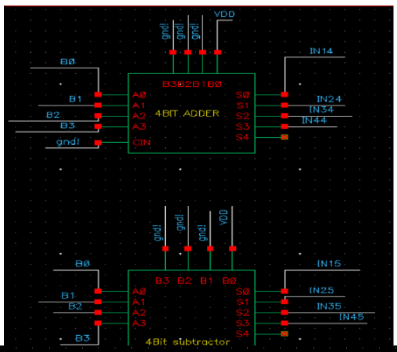
NOR TestBench

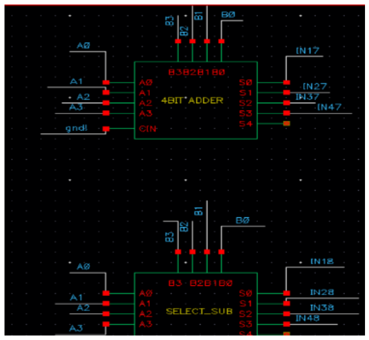
Arithmetic unit (INVERTED)

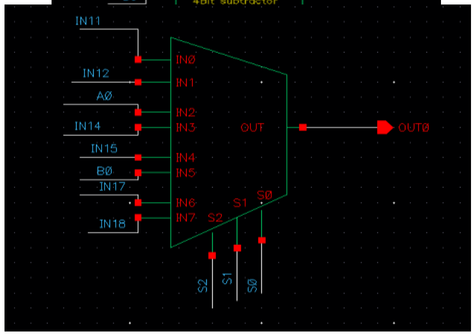
The arithmetic unit consists of 6 blocks each block outputs the output of one of the 8 functions specific for the arithmetic unit the other two outputs which are transfer A and transfer B are taken directly from the input, then the outputs are connected to 4 MUXs responsible for outputting the 4 bit output, the output of the blocks is inverted due to the use of inverted MUXs but when the Arithmetic unit output is connected to the MUXs of the ALU it is inverted again so the output of the ALU is the true value







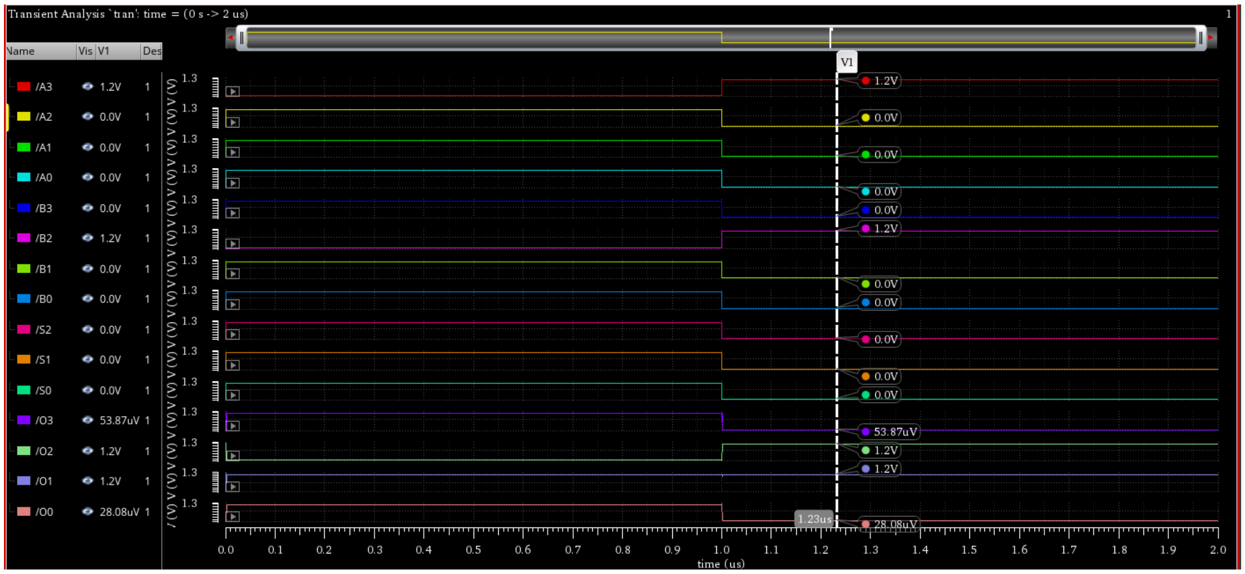




TestBench of Arithmetic Unit

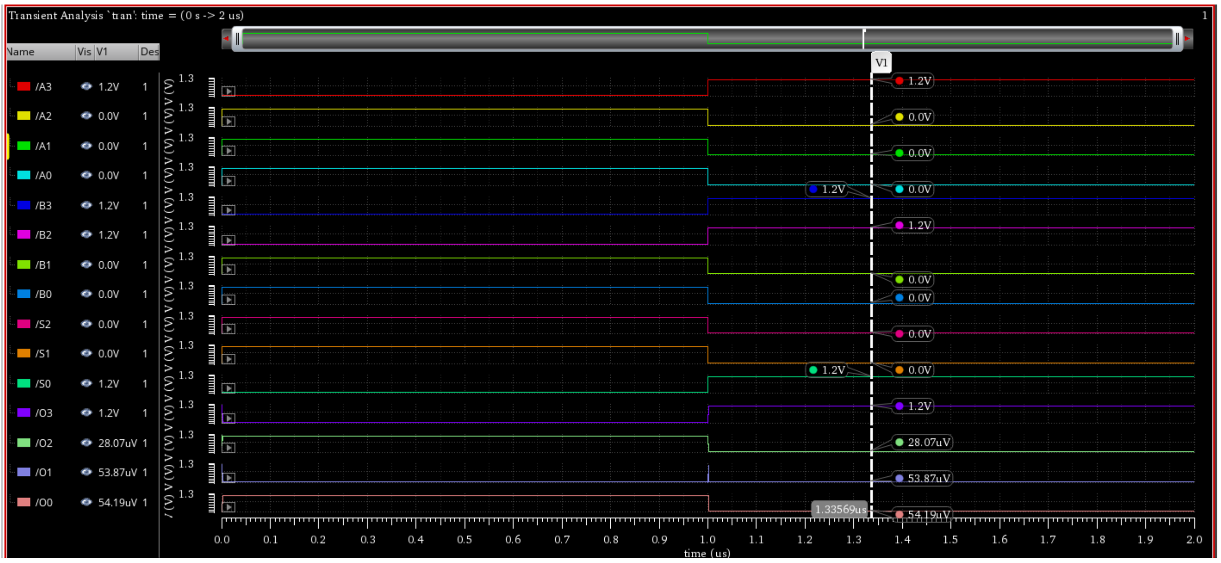
All the outputs of the arithmetic unit are inverted due to the use of inverter logic MUXs this inversion is removed by the last layer of MUXs which is in the ALU

INCREMENT A


Transient Analysis `tran': time = (0 s -> 2 us)

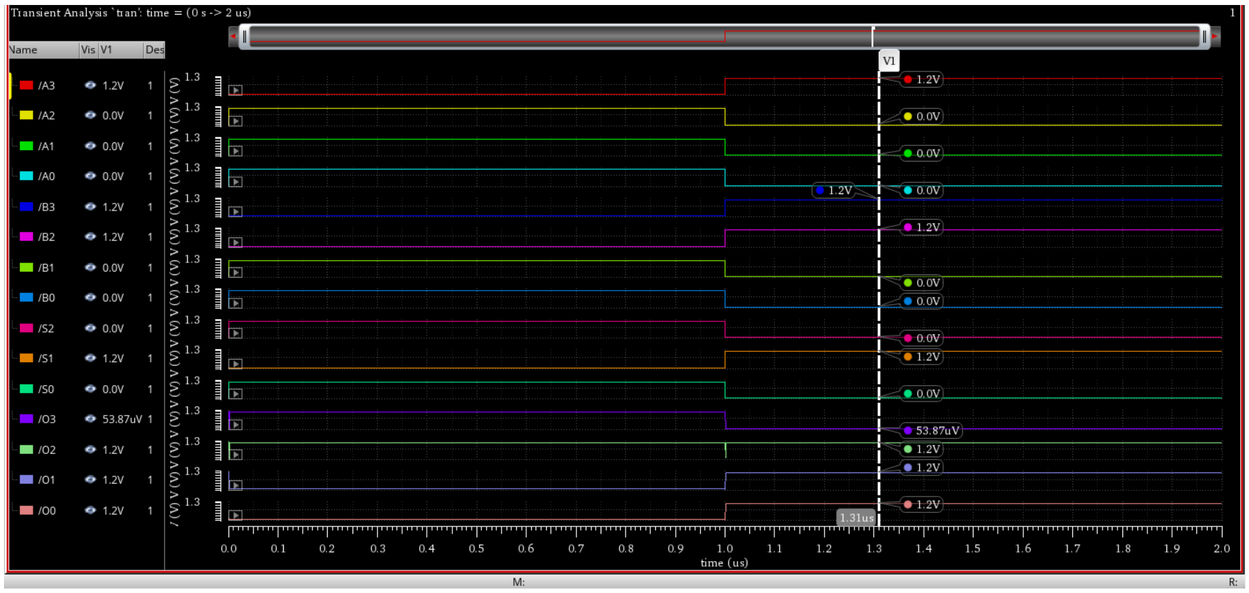
Name	Vis	V1	Des
/A3	👁	1.2V	1
/A2	👁	0.0V	1
/A1	👁	0.0V	1
/A0	👁	0.0V	1
/B3	👁	0.0V	1
/B2	👁	1.2V	1
/B1	👁	0.0V	1
/B0	👁	0.0V	1
/S2	👁	0.0V	1
/S1	👁	0.0V	1
/S0	👁	0.0V	1
/O3	👁	53.87uV	1
/O2	👁	1.2V	1
/O1	👁	1.2V	1
/O0	👁	28.08uV	1

DECREMENT A


Transient Analysis `tran': time = (0 s -> 2 us)

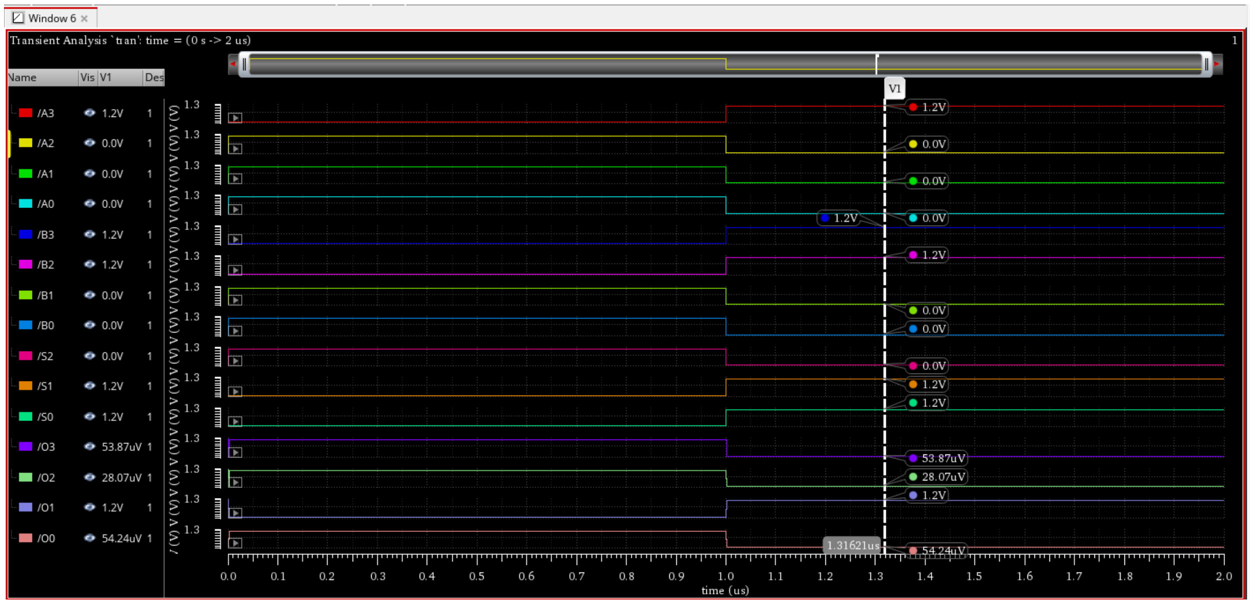
Name	Vis	V1	Des
/A3	👁	1.2V	1
/A2	👁	0.0V	1
/A1	👁	0.0V	1
/A0	👁	0.0V	1
/B3	👁	1.2V	1
/B2	👁	1.2V	1
/B1	👁	0.0V	1
/B0	👁	0.0V	1
/S2	👁	0.0V	1
/S1	👁	0.0V	1
/S0	👁	1.2V	1
/O3	👁	1.2V	1
/O2	👁	28.07uV	1
/O1	👁	53.87uV	1
/O0	👁	54.19uV	1

TRANSFER A



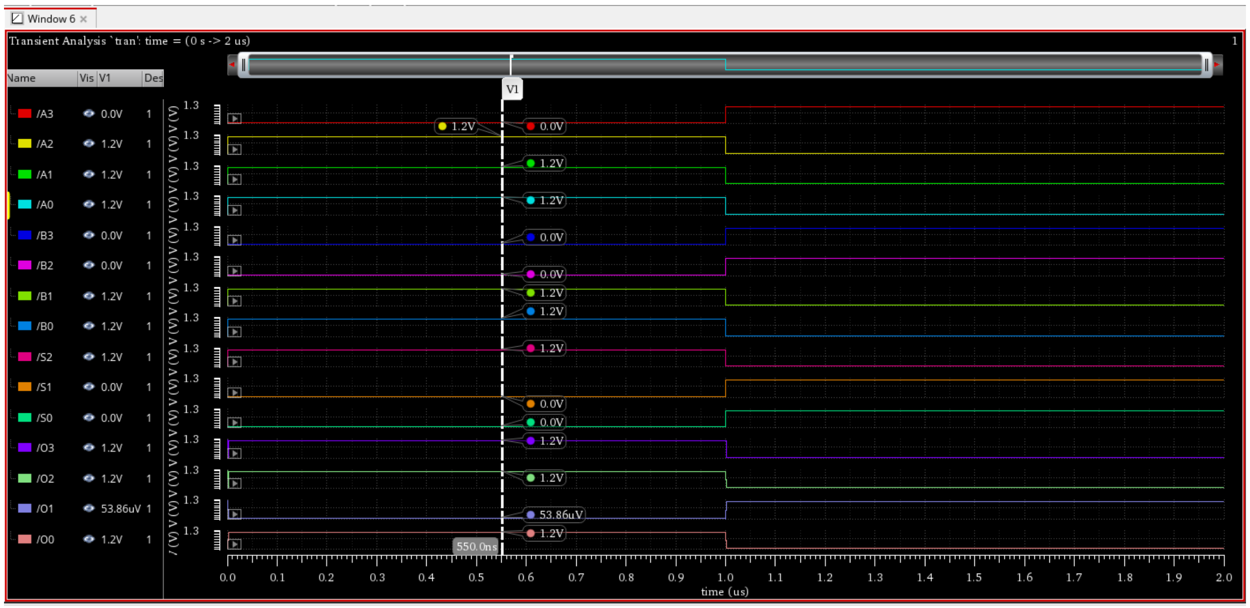
Name	Vis	V1	Des
/A3		1.2V	1
/A2		0.0V	1
/A1		0.0V	1
/A0		0.0V	1
/B3		1.2V	1
/B2		1.2V	1
/B1		0.0V	1
/B0		0.0V	1
/S2		0.0V	1
/S1		1.2V	1
/S0		0.0V	1
/O3		53.87uV	1
/O2		1.2V	1
/O1		1.2V	1
/O0		1.2V	1

INCREMENT B



Name	Vis	V1	Des
/A3		1.2V	1
/A2		0.0V	1
/A1		0.0V	1
/A0		0.0V	1
/B3		1.2V	1
/B2		1.2V	1
/B1		0.0V	1
/B0		0.0V	1
/S2		0.0V	1
/S1		1.2V	1
/S0		1.2V	1
/O3		53.87uV	1
/O2		28.07uV	1
/O1		1.2V	1
/O0		54.24uV	1

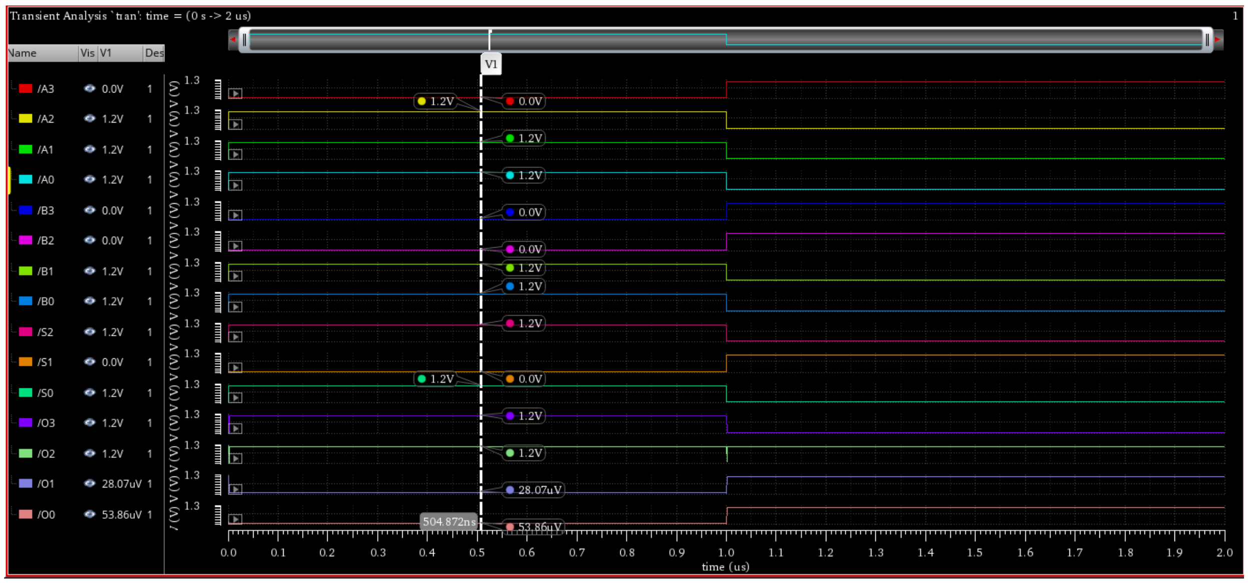
DECREMENT B



Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		0.0V	1
/A2		1.2V	1
/A1		1.2V	1
/A0		1.2V	1
/B3		0.0V	1
/B2		0.0V	1
/B1		1.2V	1
/B0		1.2V	1
/S2		1.2V	1
/S1		0.0V	1
/S0		0.0V	1
/O3		1.2V	1
/O2		1.2V	1
/O1		53.86uV	1
/O0		1.2V	1

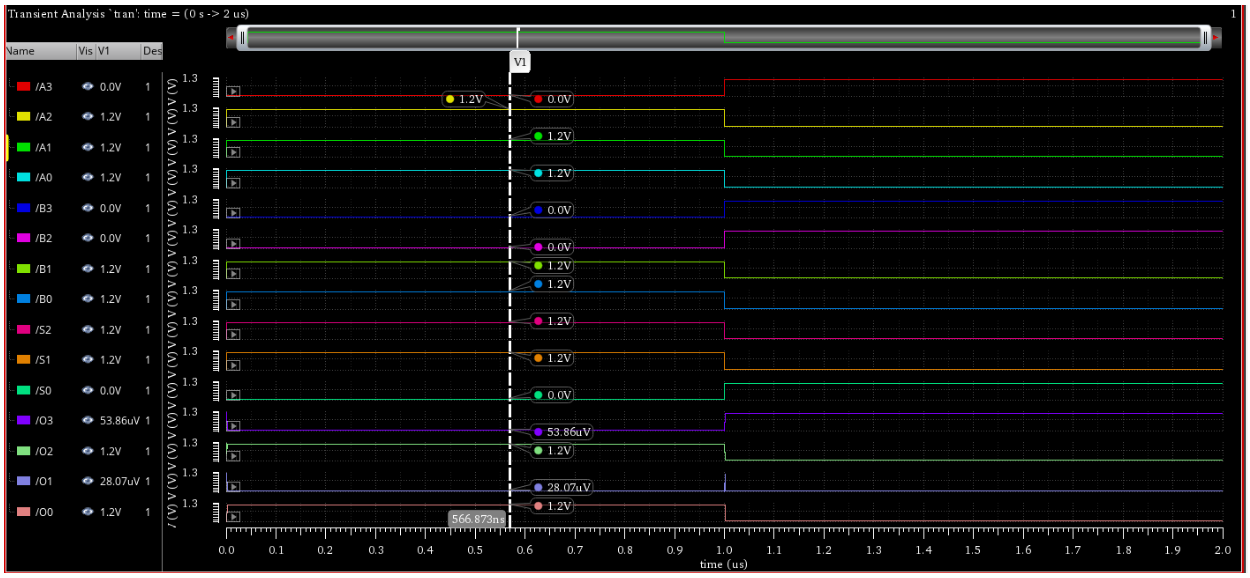
TRANSFER B



Transient Analysis `tran': time = (0 s -> 2 us)

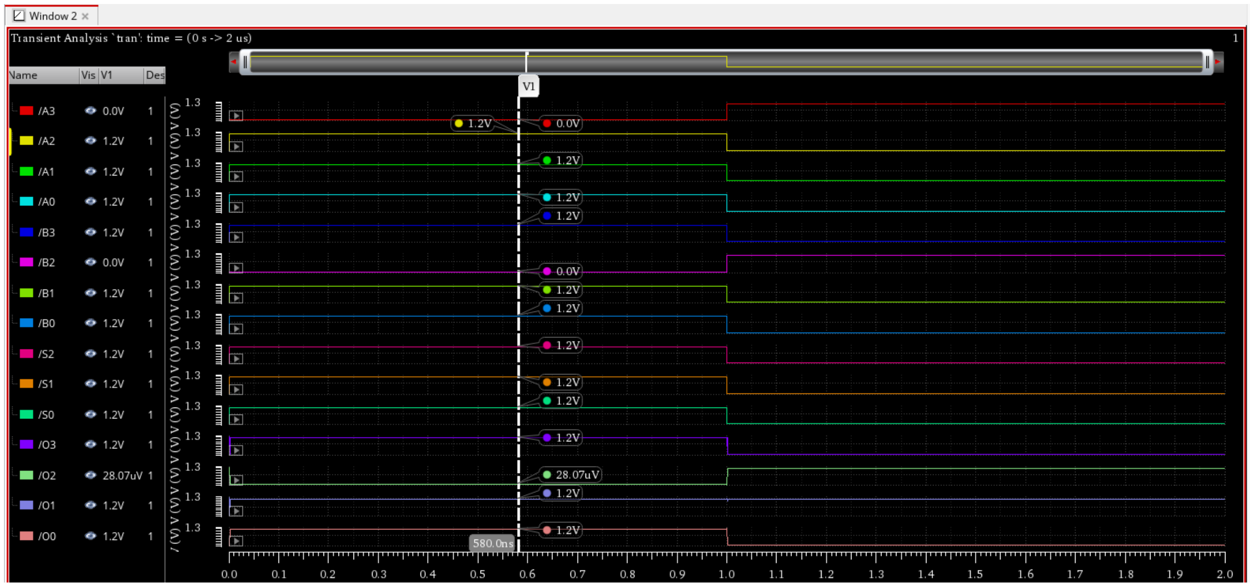
Name	Vis	V1	Des
/A3		0.0V	1
/A2		1.2V	1
/A1		1.2V	1
/A0		1.2V	1
/B3		0.0V	1
/B2		0.0V	1
/B1		1.2V	1
/B0		1.2V	1
/S2		1.2V	1
/S1		0.0V	1
/S0		1.2V	1
/O3		1.2V	1
/O2		1.2V	1
/O1		28.07uV	1
/O0		53.86uV	1

ADD A+B


Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		0.0V	1
/A2		1.2V	1
/A1		1.2V	1
/A0		1.2V	1
/B3		0.0V	1
/B2		0.0V	1
/B1		1.2V	1
/B0		1.2V	1
/S2		1.2V	1
/S1		1.2V	1
/S0		0.0V	1
/O3		53.86uV	1
/O2		1.2V	1
/O1		28.07uV	1
/O0		1.2V	1

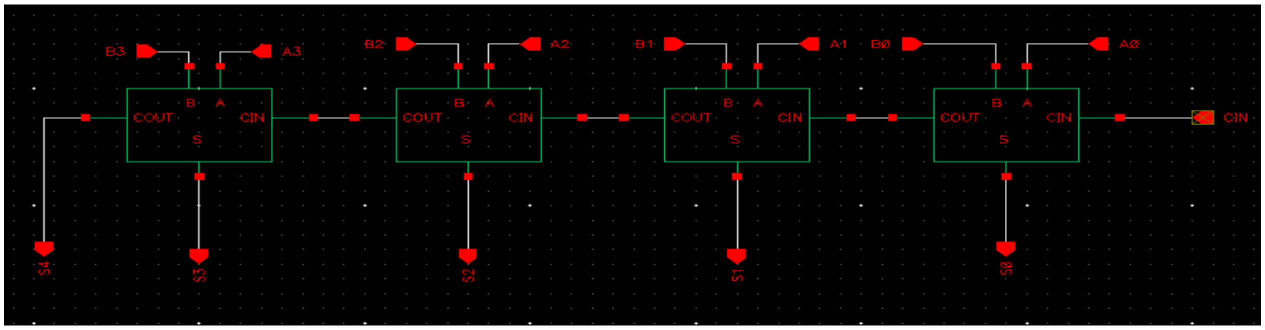
SUBTRACT A-B


Window 2

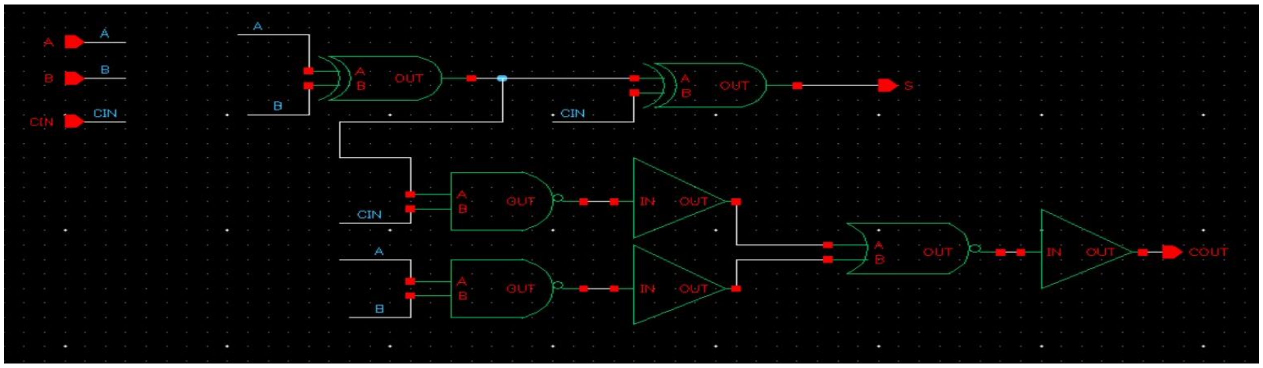
Transient Analysis `tran': time = (0 s -> 2 us)

Name	Vis	V1	Des
/A3		0.0V	1
/A2		1.2V	1
/A1		1.2V	1
/A0		1.2V	1
/B3		1.2V	1
/B2		0.0V	1
/B1		1.2V	1
/B0		1.2V	1
/S2		1.2V	1
/S1		1.2V	1
/S0		1.2V	1
/O3		1.2V	1
/O2		28.07uV	1
/O1		1.2V	1
/O0		1.2V	1

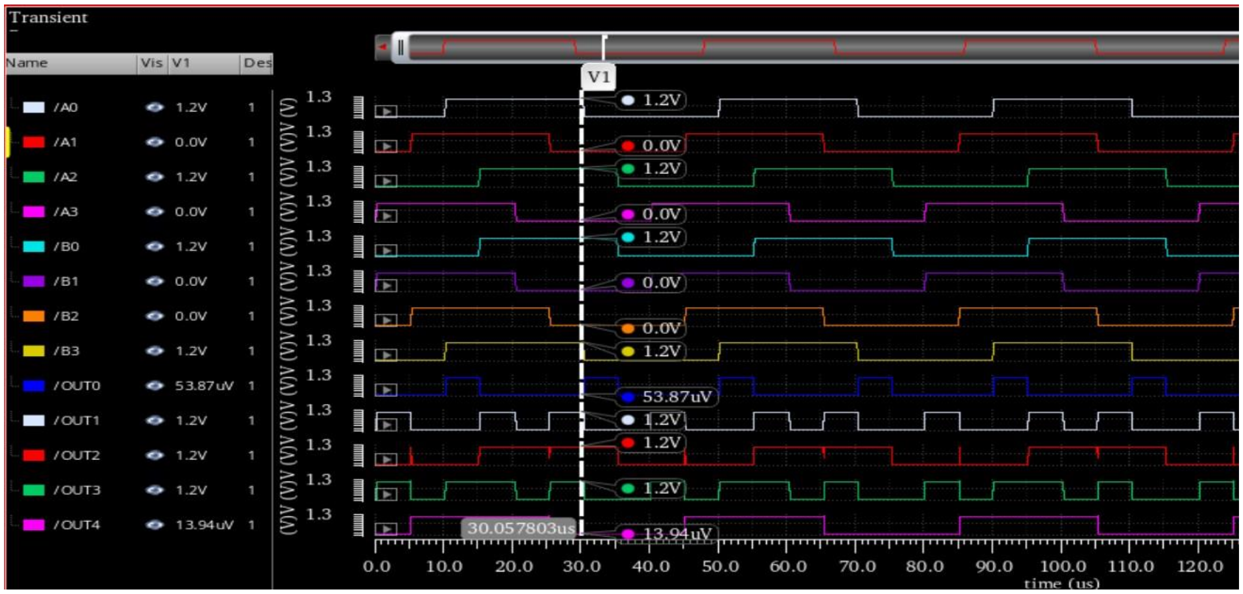
Adder



Full Adder block



Full Adder design



Transient

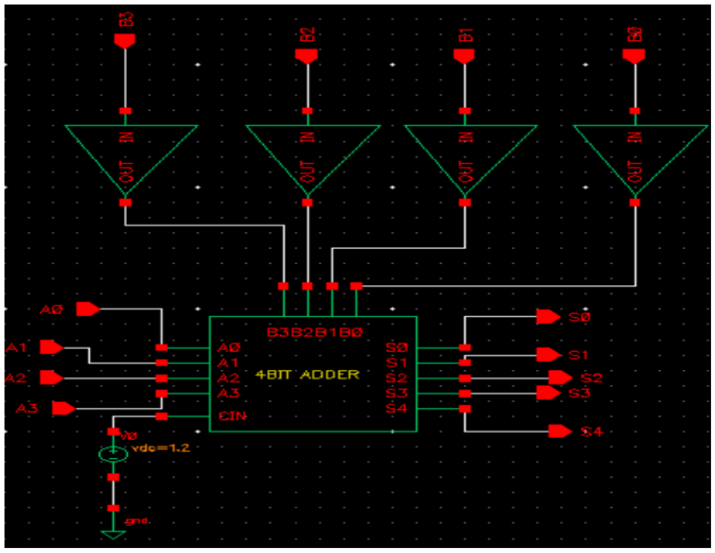
Name	Vis	V1	Des
/A0		1.2V	1
/A1		0.0V	1
/A2		1.2V	1
/A3		0.0V	1
/B0		1.2V	1
/B1		0.0V	1
/B2		0.0V	1
/B3		1.2V	1
/OUT0		53.87uV	1
/OUT1		1.2V	1
/OUT2		1.2V	1
/OUT3		1.2V	1
/OUT4		13.94uV	1

Full Adder TestBench

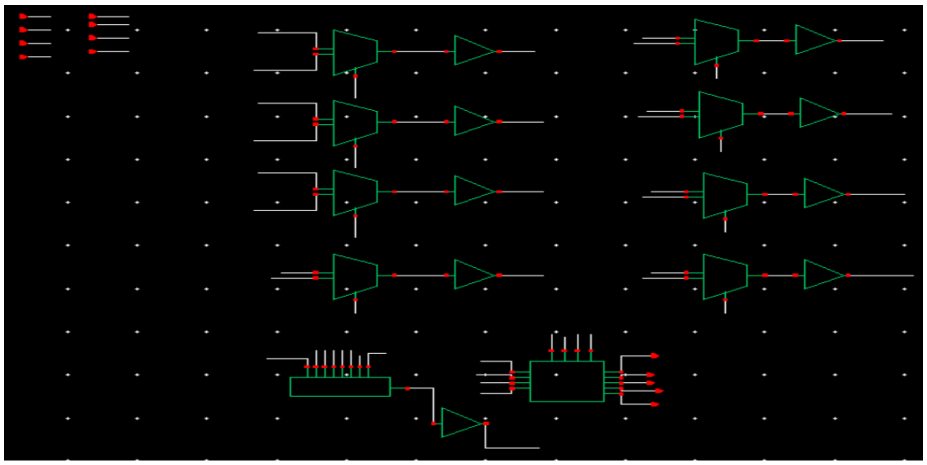
Subtractor

The 4bit subtractor consists of a 4bit adder but the carry in bit should be high and the input B should be inverted as A-B=A+B’+1

The subselect block uses the 4bit subtractor we designed which subtracts A(4bits)-B(4bits) and 8 MUXs that control which input goes to A and B of the subtractor the MUXs are controlled by the AGTB(A greater than B) block which outputs high if input A is greater than B if this is the case then Input A is connected to input A of the subtractor and input B is connected to input B of the subtractor if the output of AGTB is zero the muxes flip and input B is connected to input A of the subtractor and input A is connected to input B of the subtractor so the subtractor operation becomes (B-A) if B>A

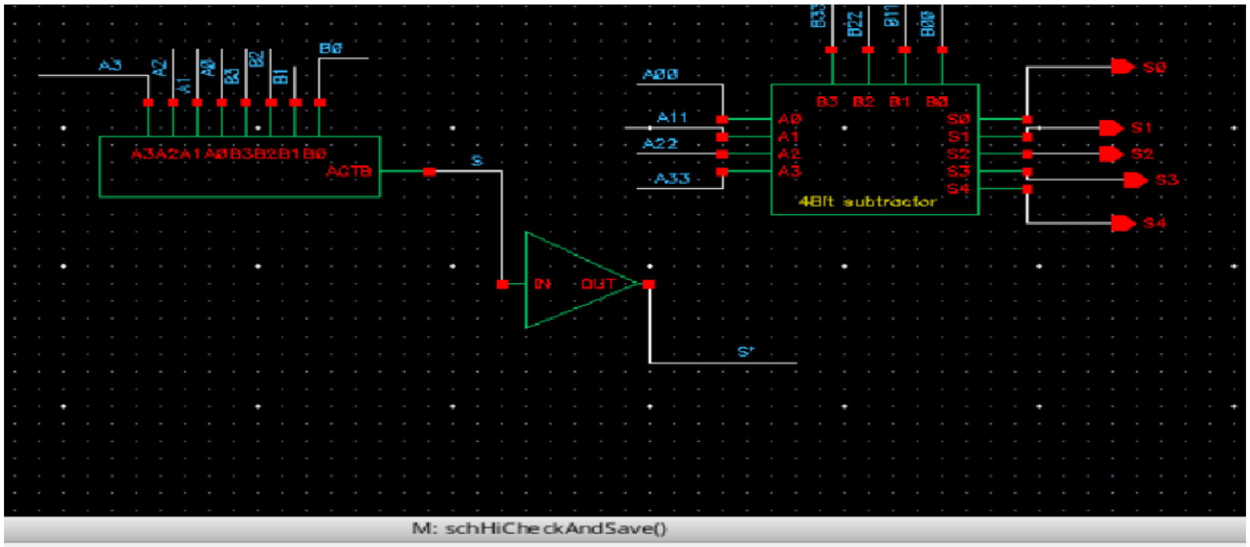


4 Bit Subtractor

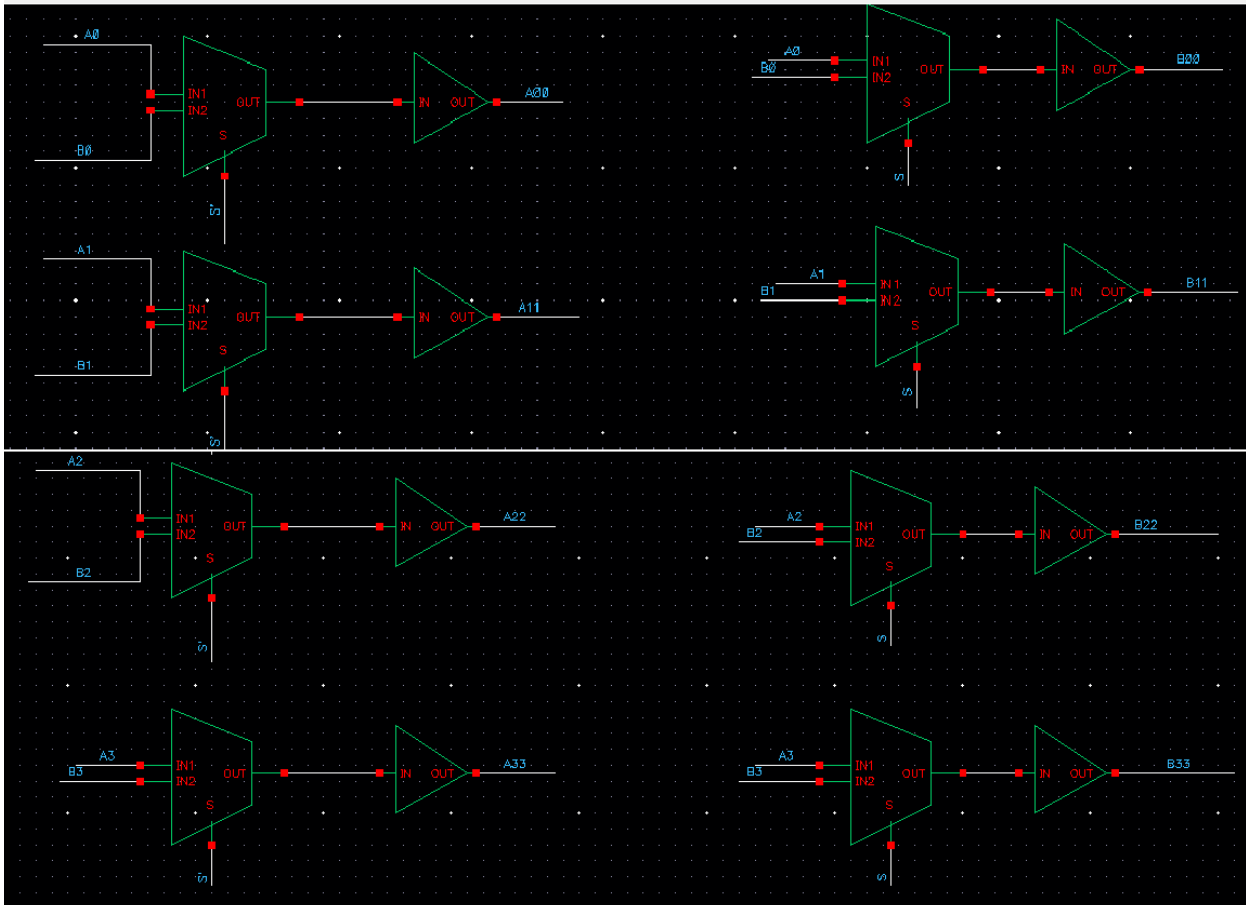


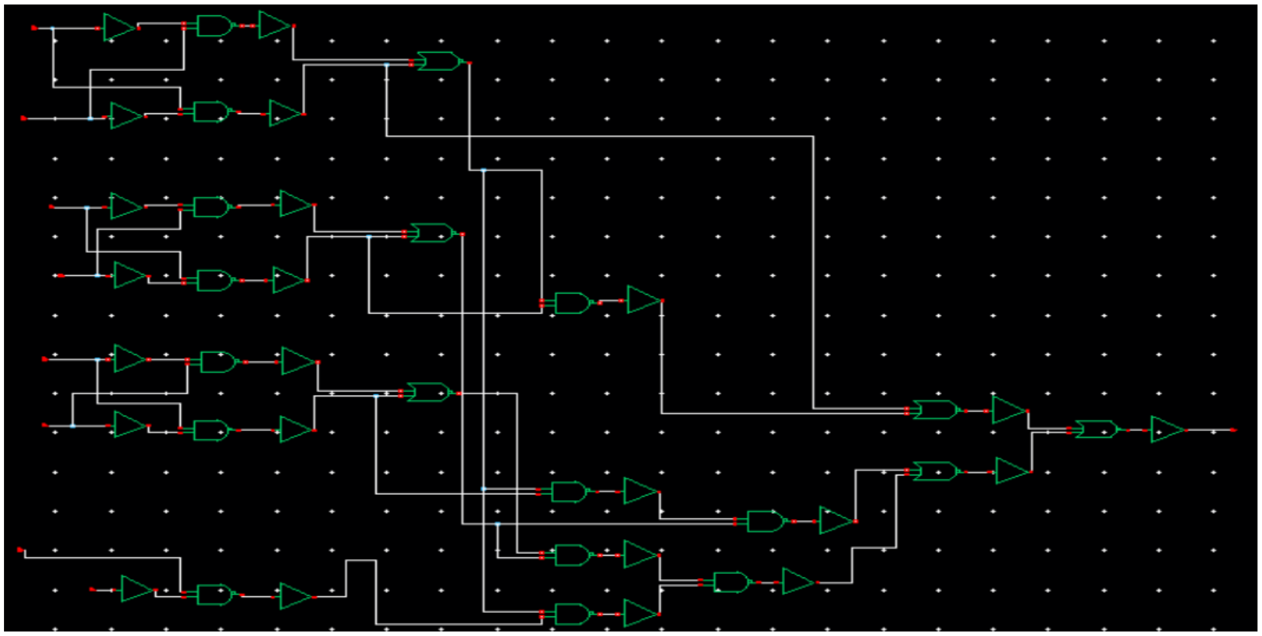
Subselect block

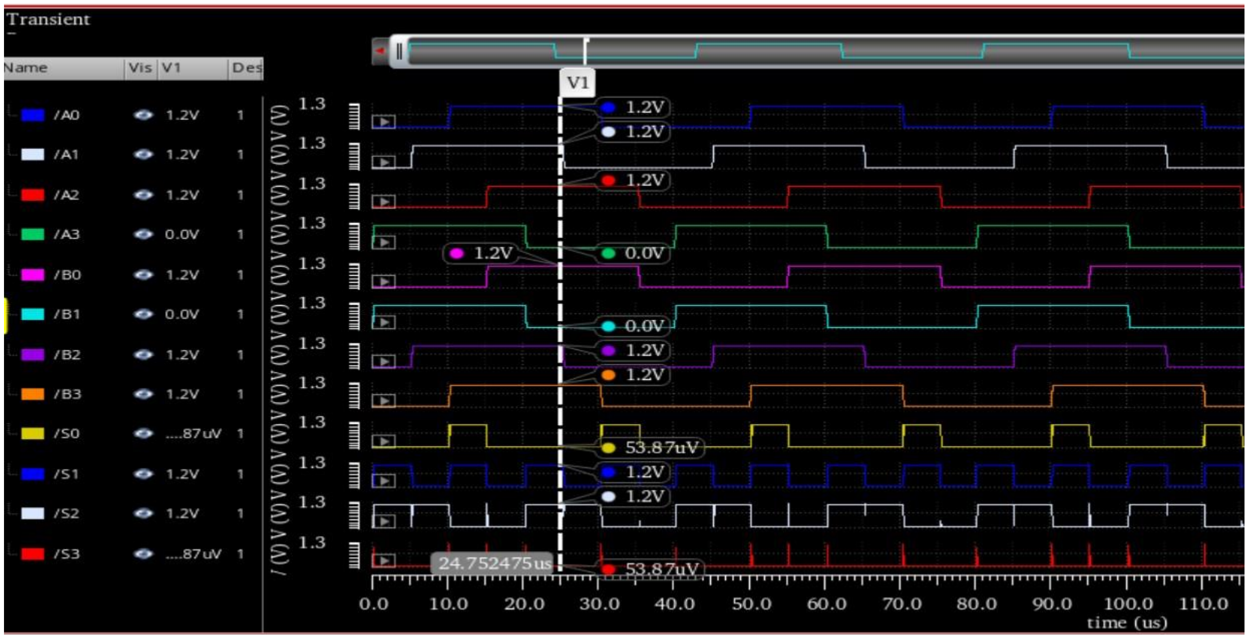
Zoomed in



M: schHiCheckAndSave()




Gate level of A greater than B block



Name	Vis	V1	Des
/A0		1.2V	1
/A1		1.2V	1
/A2		1.2V	1
/A3		0.0V	1
/B0		1.2V	1
/B1		0.0V	1
/B2		1.2V	1
/B3		1.2V	1
/S0		....87uV	1
/S1		1.2V	1
/S2		1.2V	1
/S3		....87uV	1

A greater than B TestBench