

# **VLSI Final Project**

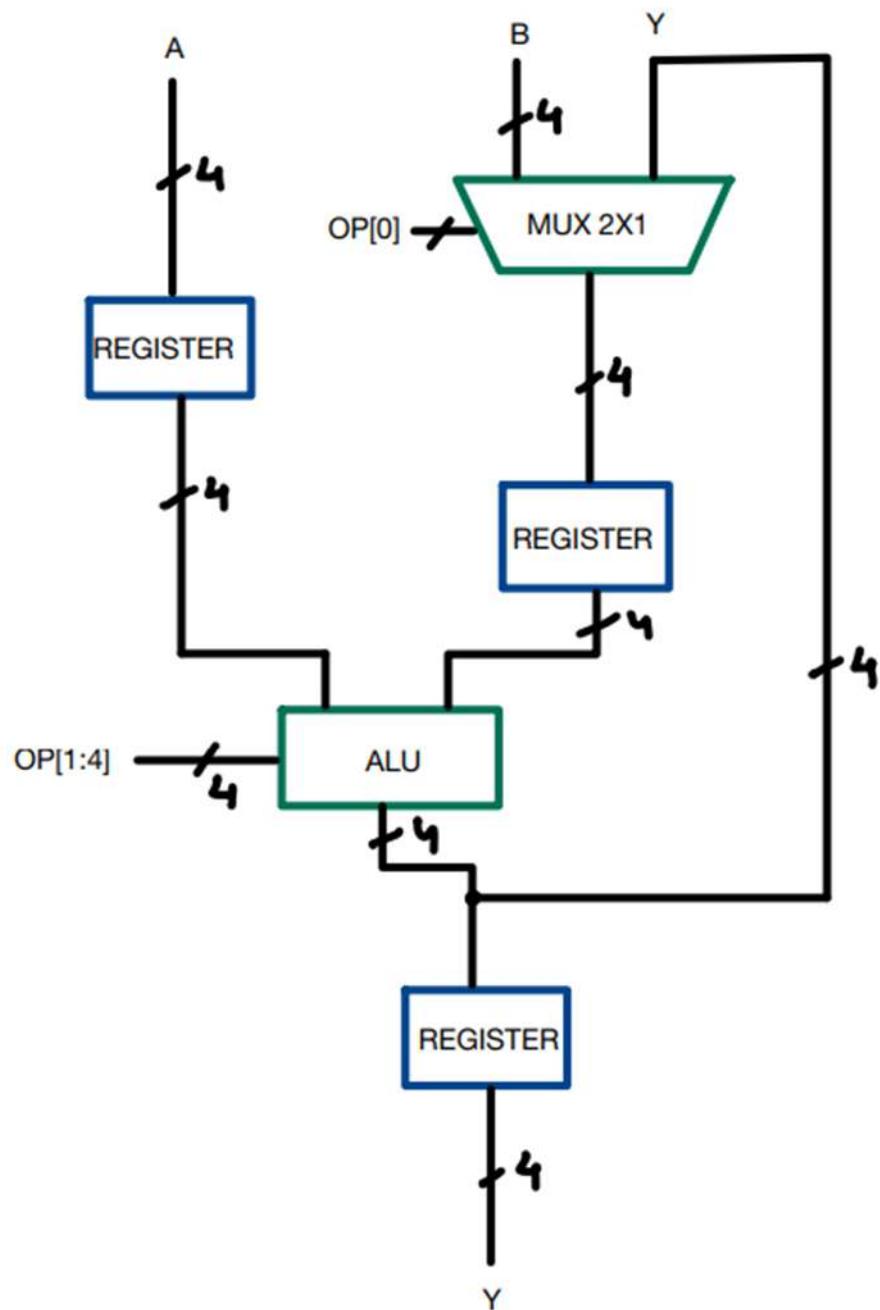
## **4-bit ALU Design**

**Phase 2**

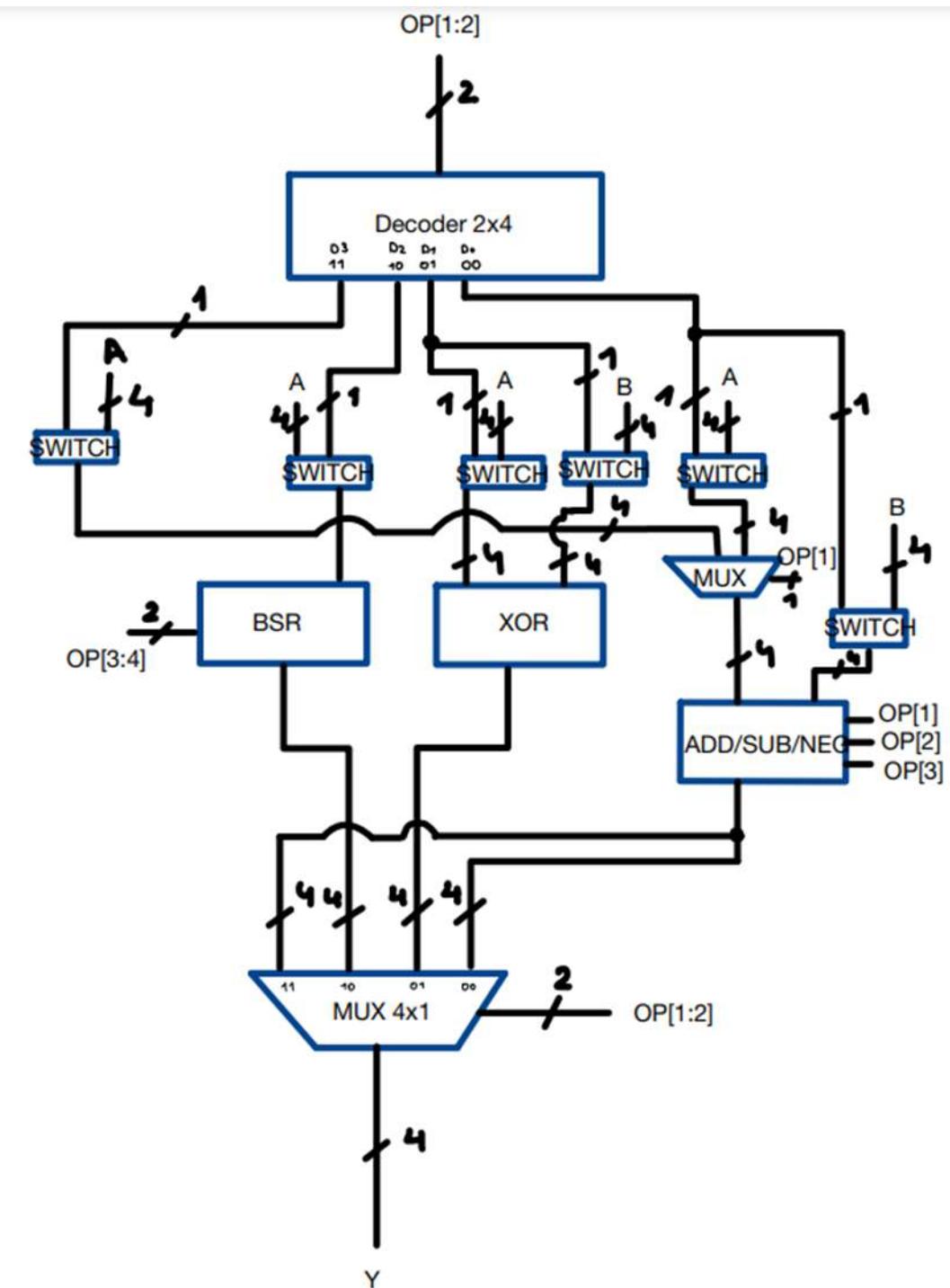
**Implementation Stage**

## Reminder!

### The Block Diagram:



In the following ALU diagram we show more of the implementations/specifications:



## DECODER

Functionality:

We used the decoder to enable a specific circuit in order to minimize the total power, according to the Opcode table below:

Decoder input: bit2, bit1

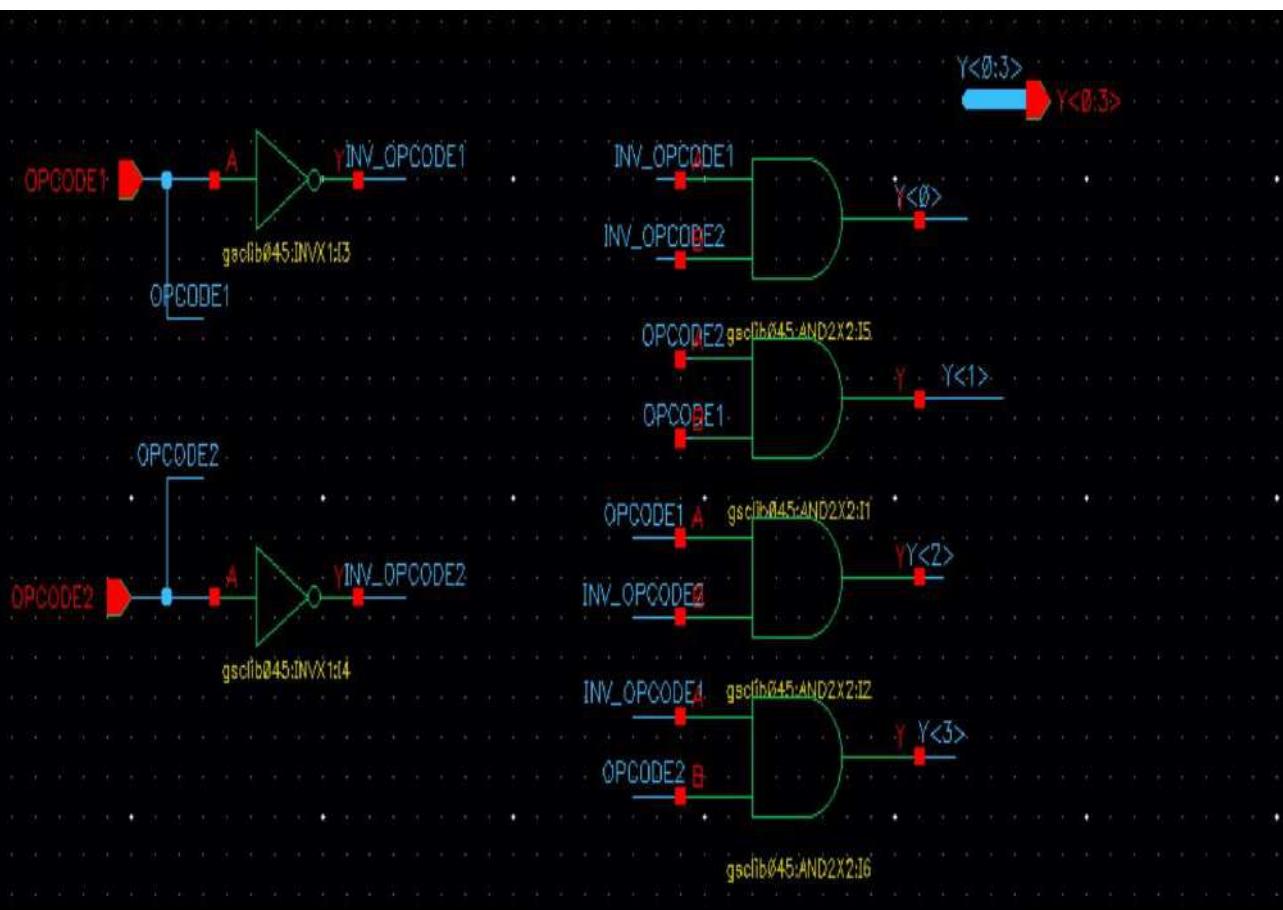
The OPCODE assignment:

	operation	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
ADD/ SUB/ NEG	ADD	X	0	0	0	
	SUB	X	1	0	0	
	NEG	X	1	1	1	
BRS	Shift 4	0	0	1	0	
	Shift 1	0	1	1	0	
	Shift 2	1	0	1	0	
	Shift 3	1	1	1	0	
Logic	XOR	X	X	0	1	

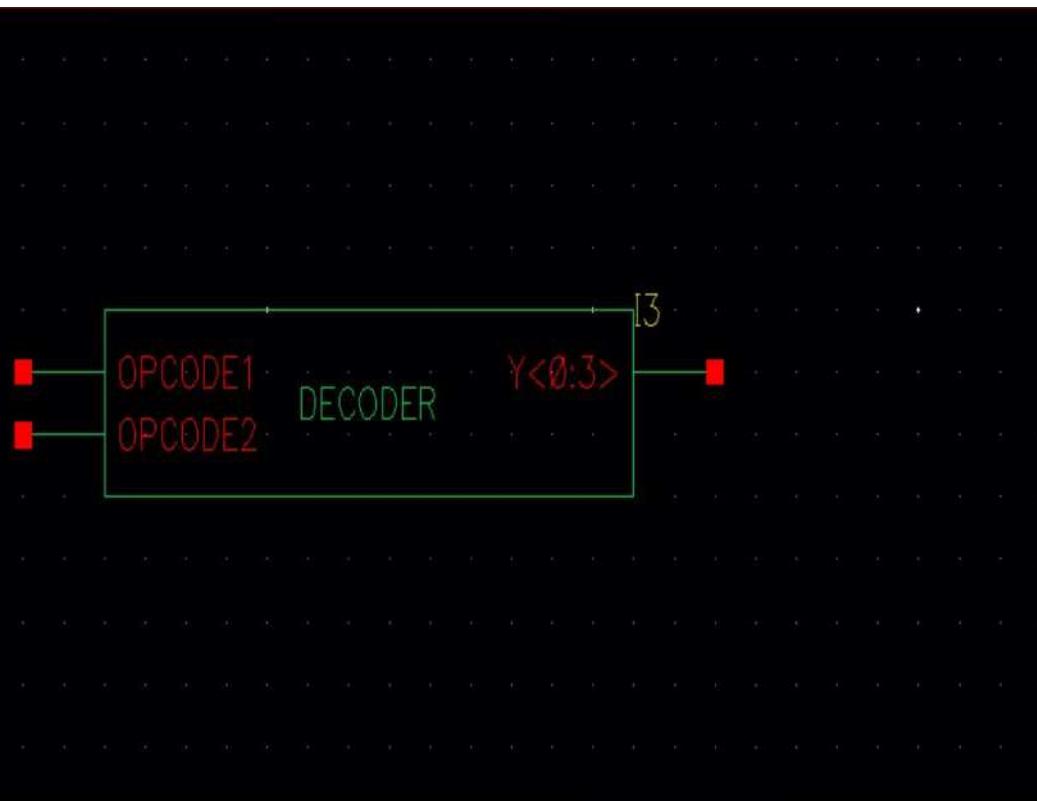
If the OPCODE first bit (in position 0) is 0 we will take input B •

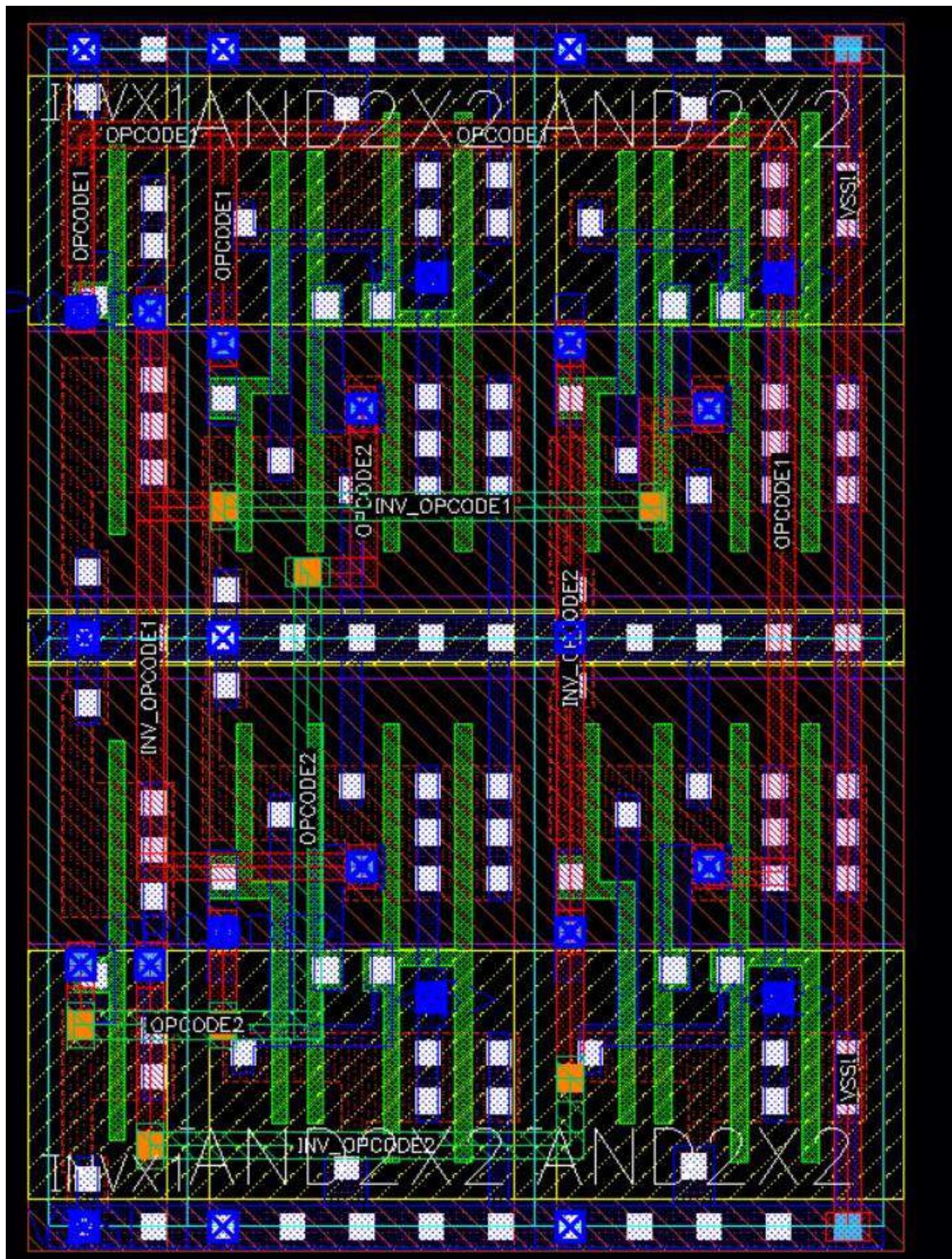
If the OPCODE first bit (in position 0) is 1 we will take input Y •

## Schematic

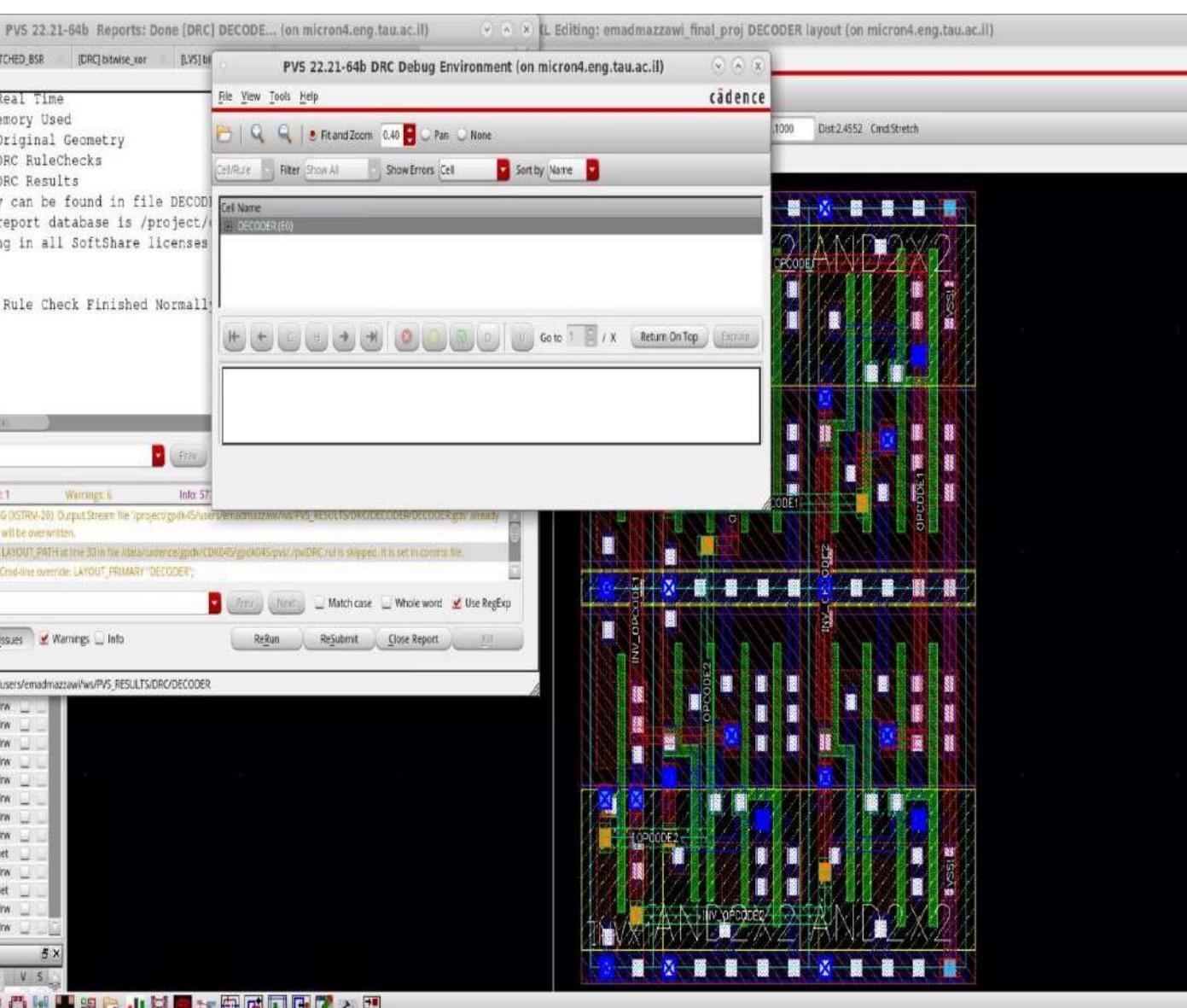


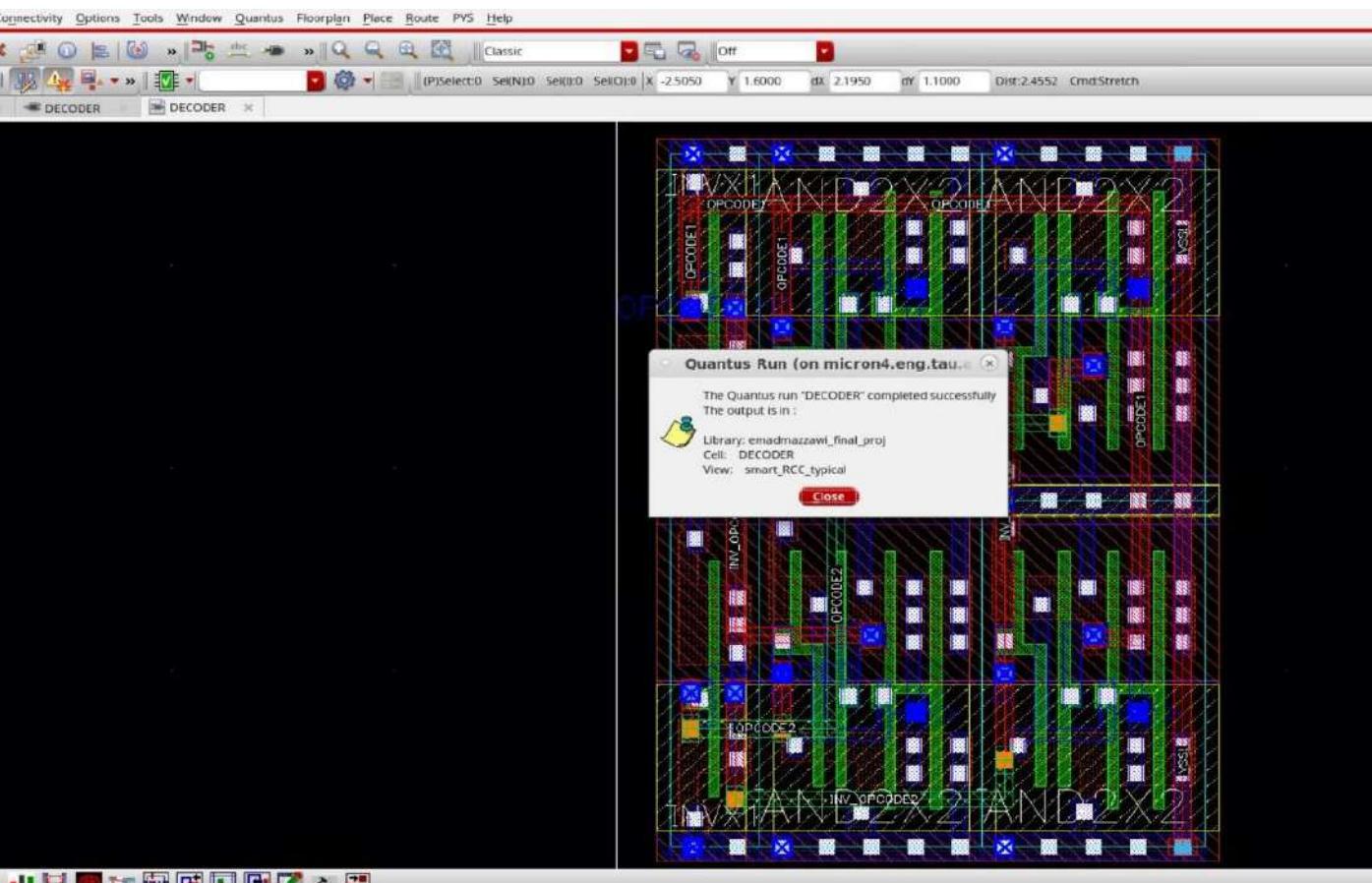
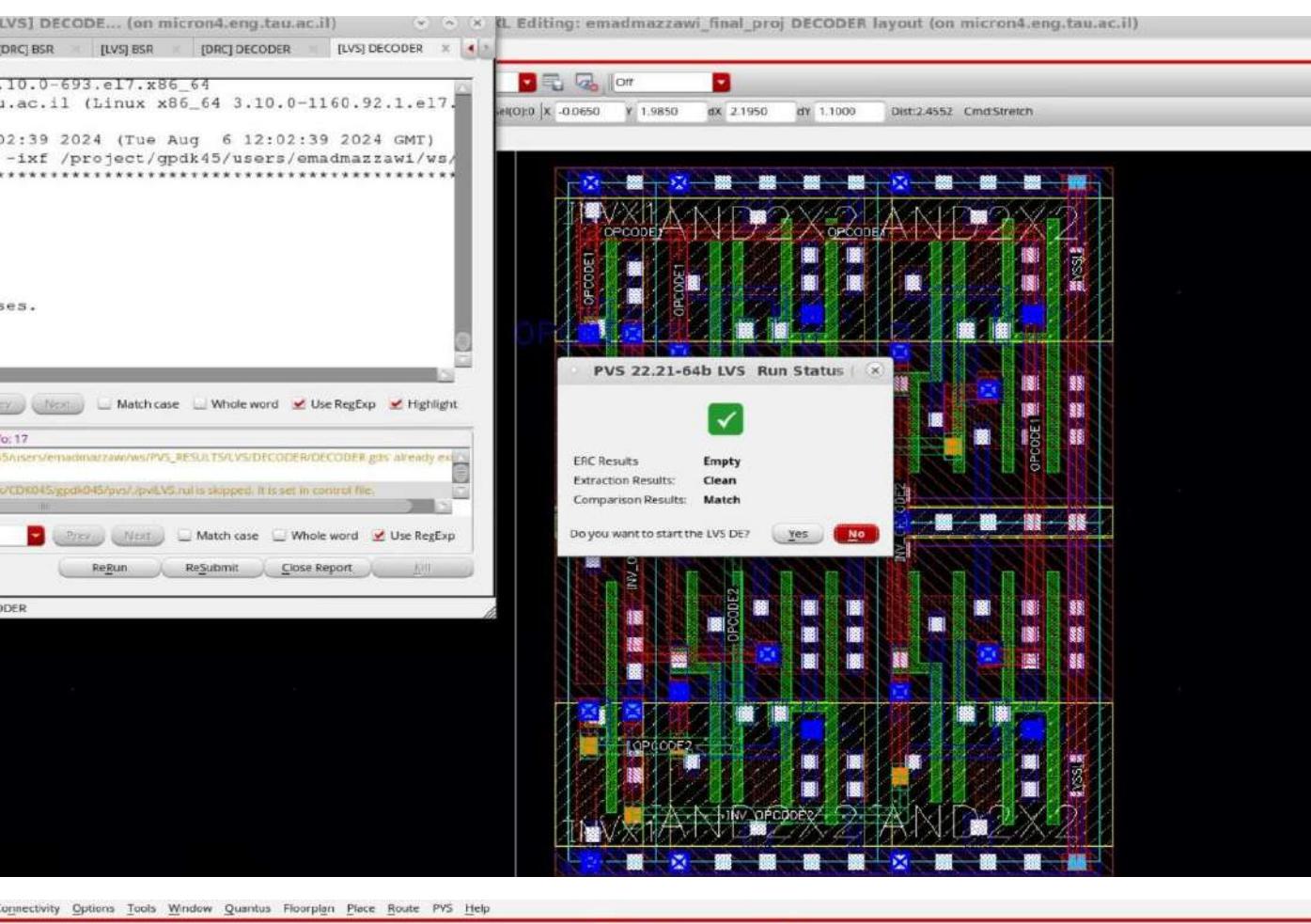
## Symbol & Layout





## DRC , LVS , QUANTUS



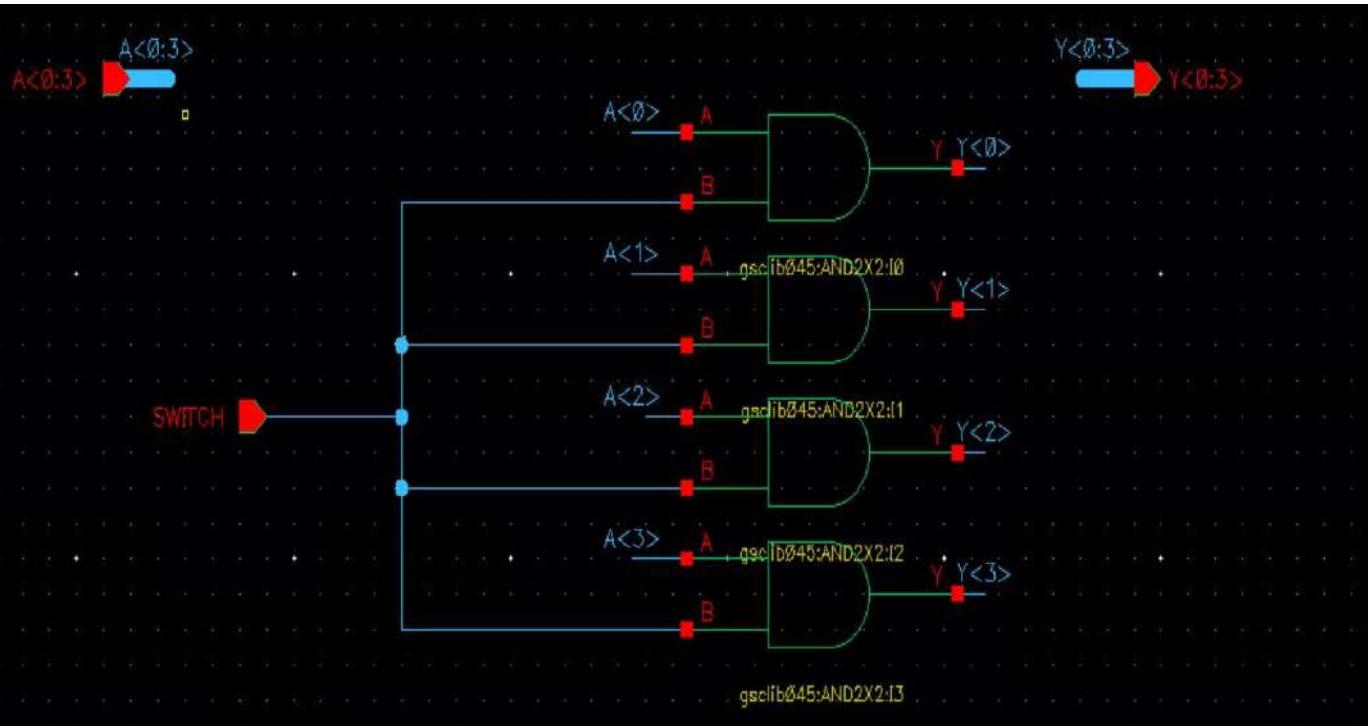


## SWITCH

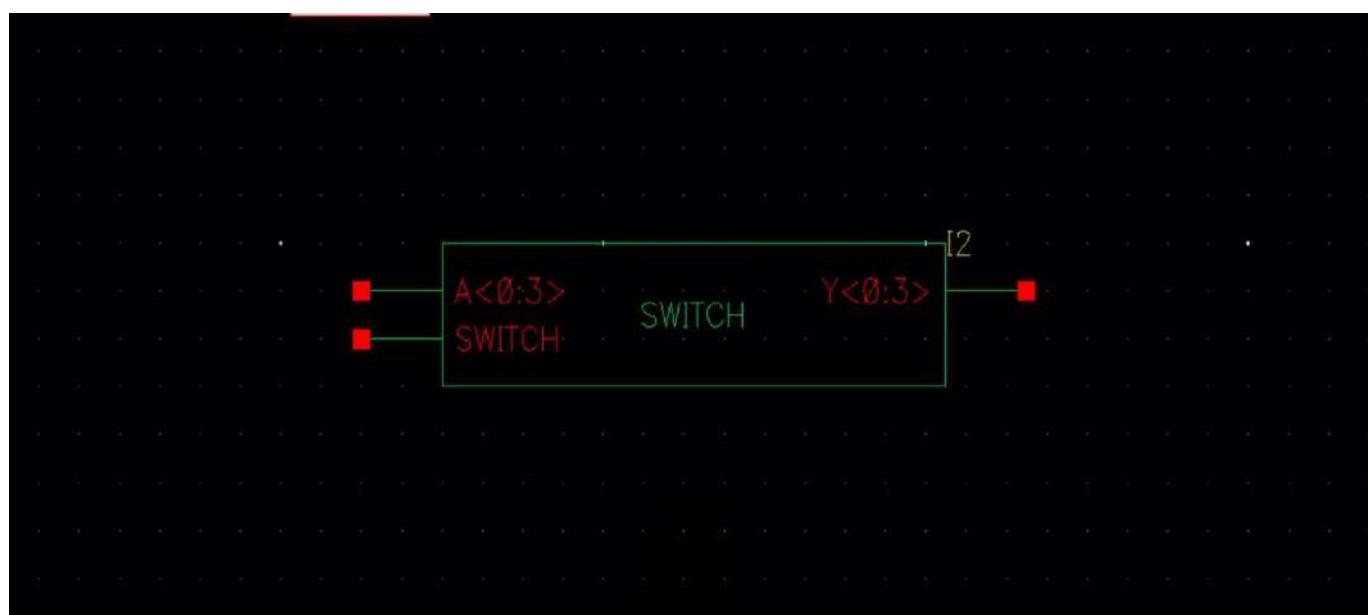
### Functionality

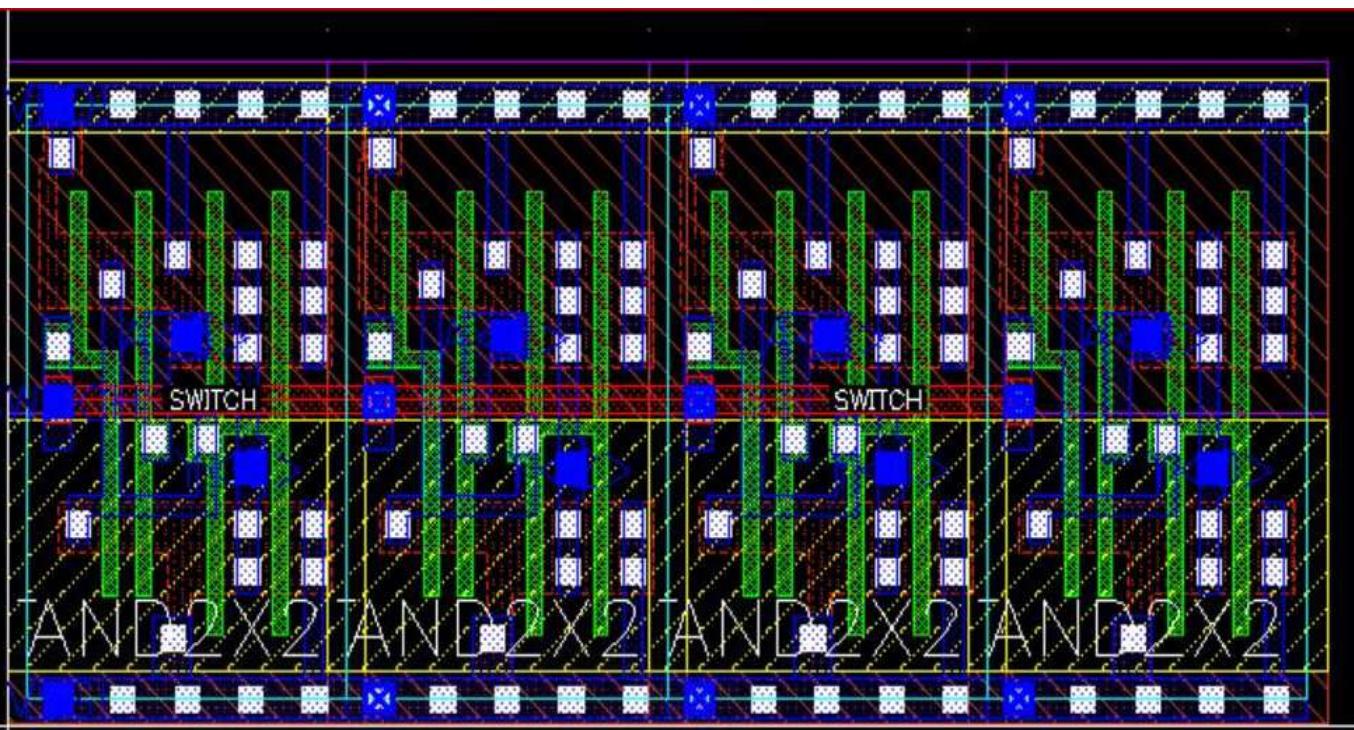
The switch component is connected to every single input in the ALU; Using the output of the decoder we determine which switches are on and the rest is off. Thus, one circuit of the ALU is working.

### Schematic



### Symbol & Layout





DRC , LVS , QUANTUS

PVS 22.21-64b Reports: Done [DRC] swit... (on micron4.eng.tau.ac.il)

[DRC] SWITCHED\_BSR [LVS] SWITCHED\_BSR [LVS] switched\_xor [DRC] switched\_xor [DRC] swit... (on micron4.eng.tau.ac.il)

cadence

1 CPU Time : 2 (s)  
1 Real Time : 3 (s)  
Memory Used : 19 (M)  
1 Original Geometry : 72  
1 DRC RuleChecks : 562  
1 DRC Results : 0  
ary can be found in file swith.sum  
I report database is /project/gpdk45/  
king in all SoftShare licenses.

gn Rule Check Finished Normally. Mo

found: 1 Warnings: 6 Info: 573

WARNING (XSTRM-20): Output Stream file '/project/gpdk45/use' overwritten.

ARN: LAYOUT\_PATH at line 30 in file /data/cadence/gpdk/CD

Cell Name: + swith (E0)

File View Tools Help

Cell/Rule Filter Show All Show Errors Cell Sort by Name

Off

Sel(0):0 X: 5.2300 Y: -0.3100 dx: 2.1950 dy: 1.1000 Dist: 2.4552 Cmd:

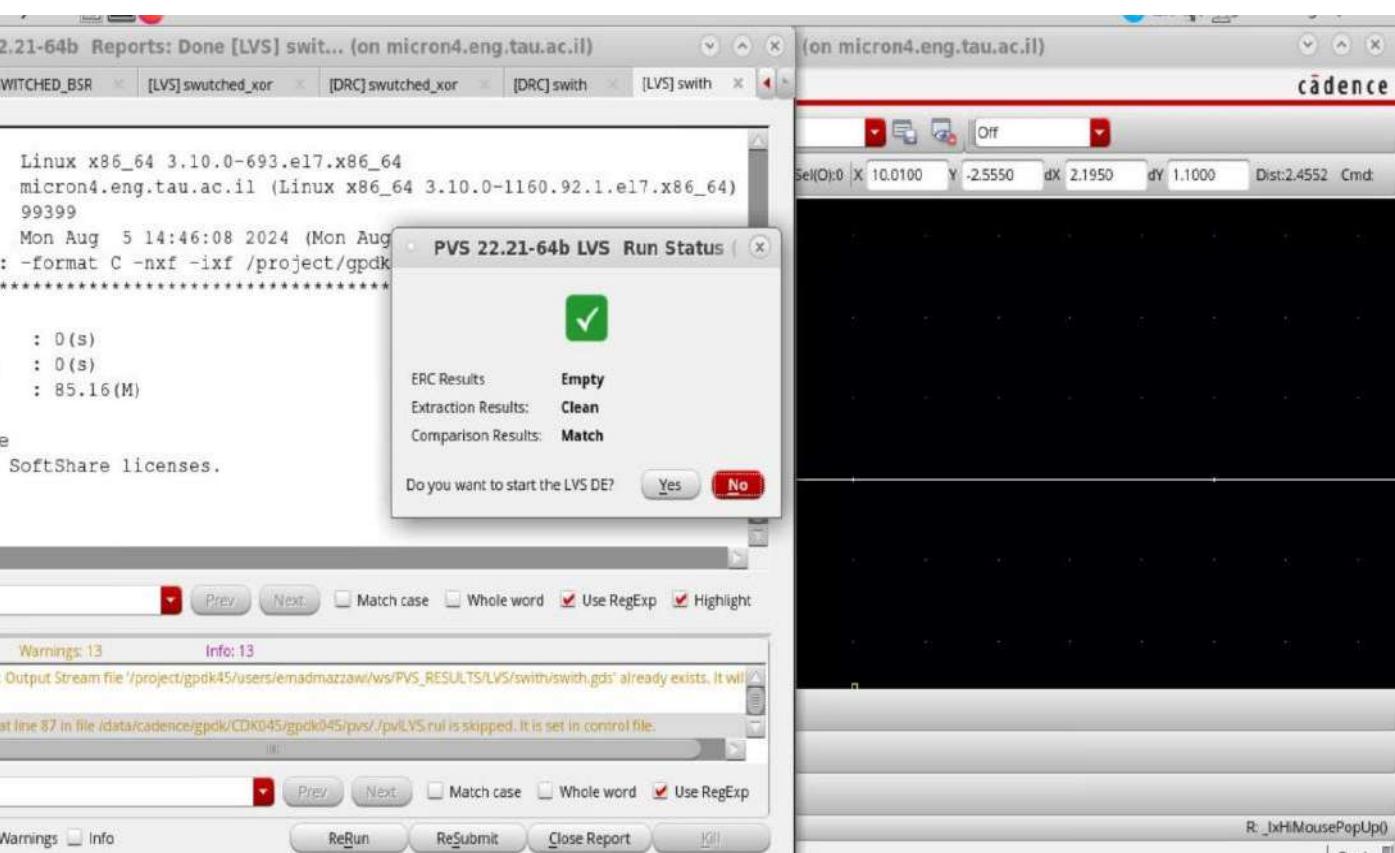
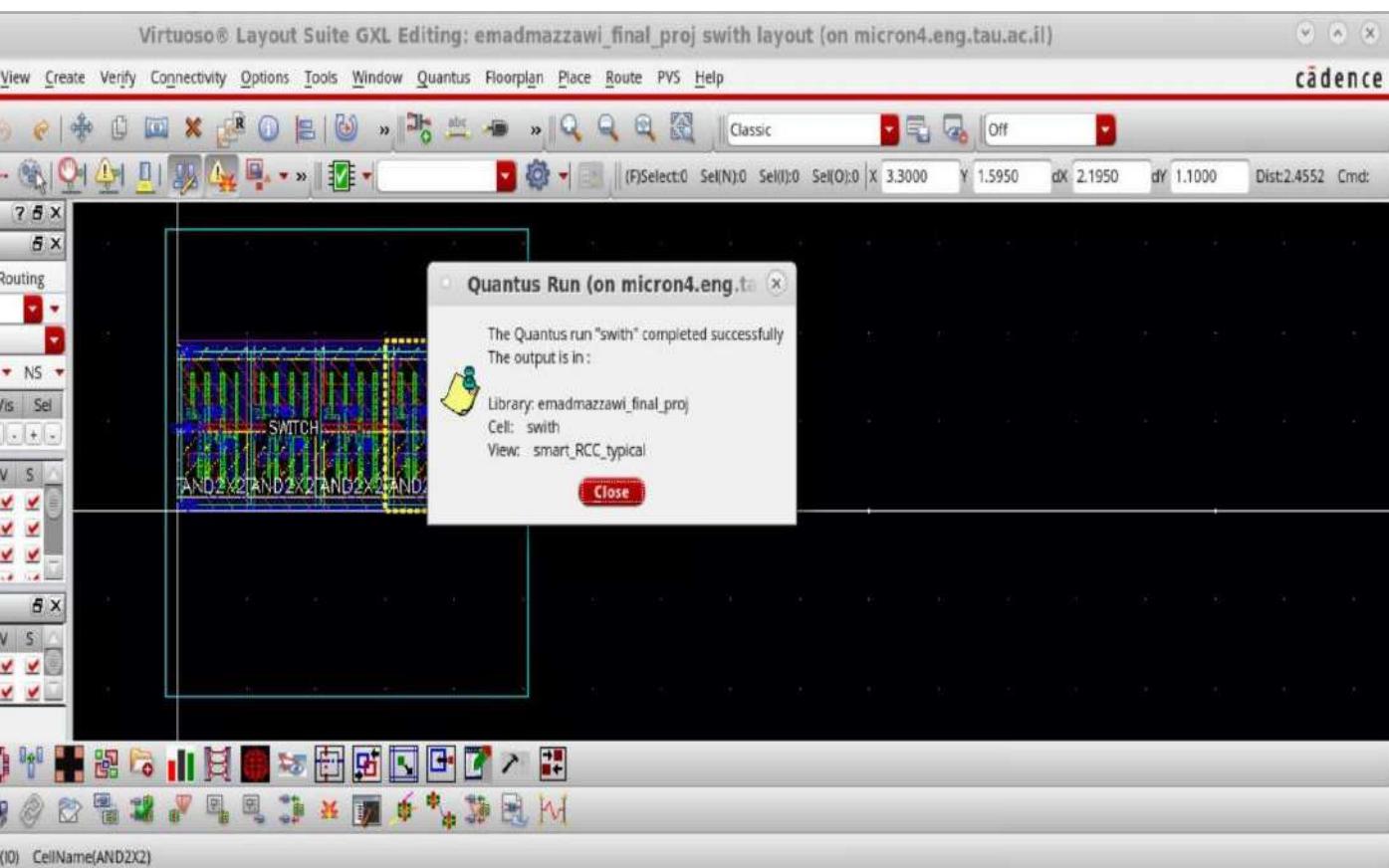
PVS 22.21-64b DRC Debug Environment (on micron4.eng.tau.ac.il)

cadence

R: \_JxHiMousePopUp()

Issues Warnings Info ReRun ReSubmit Close Report

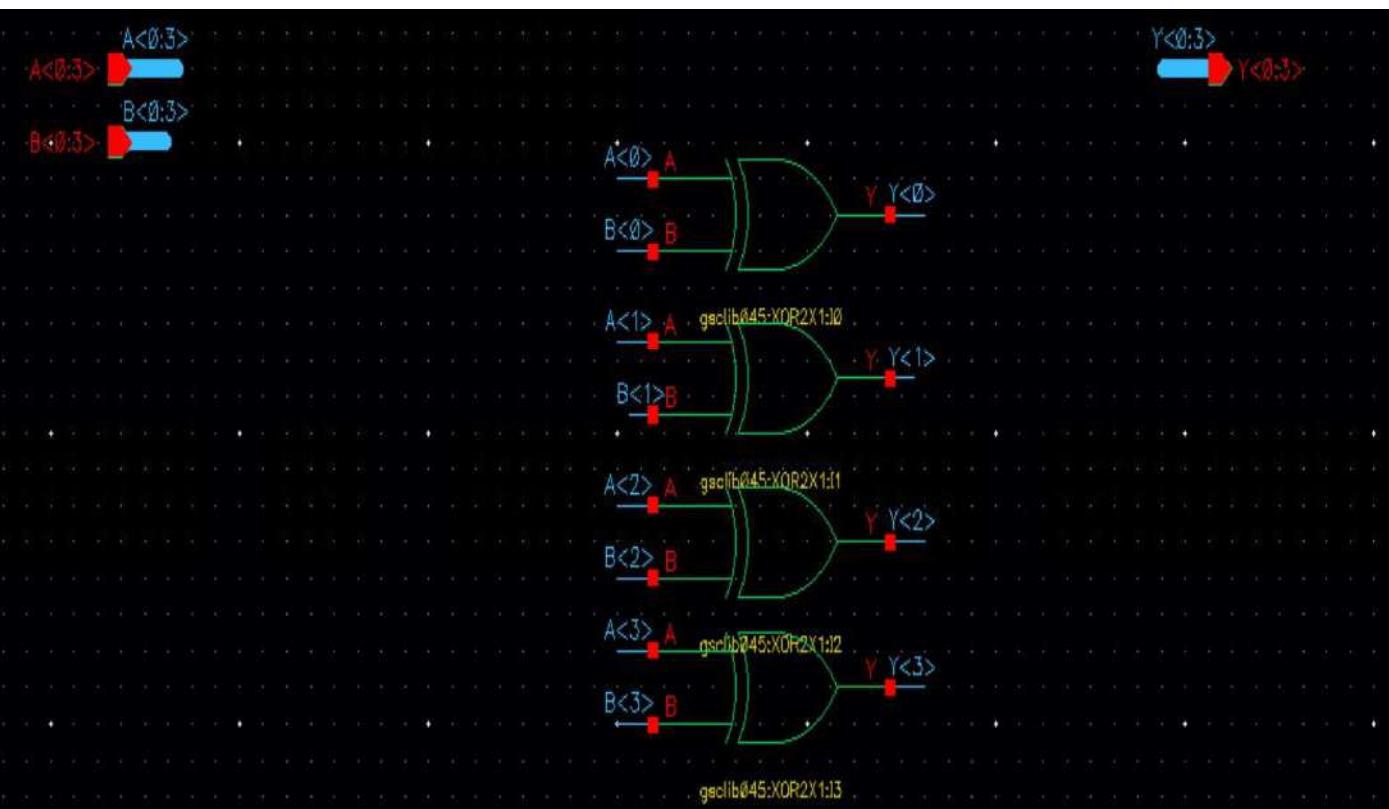
Cmd:



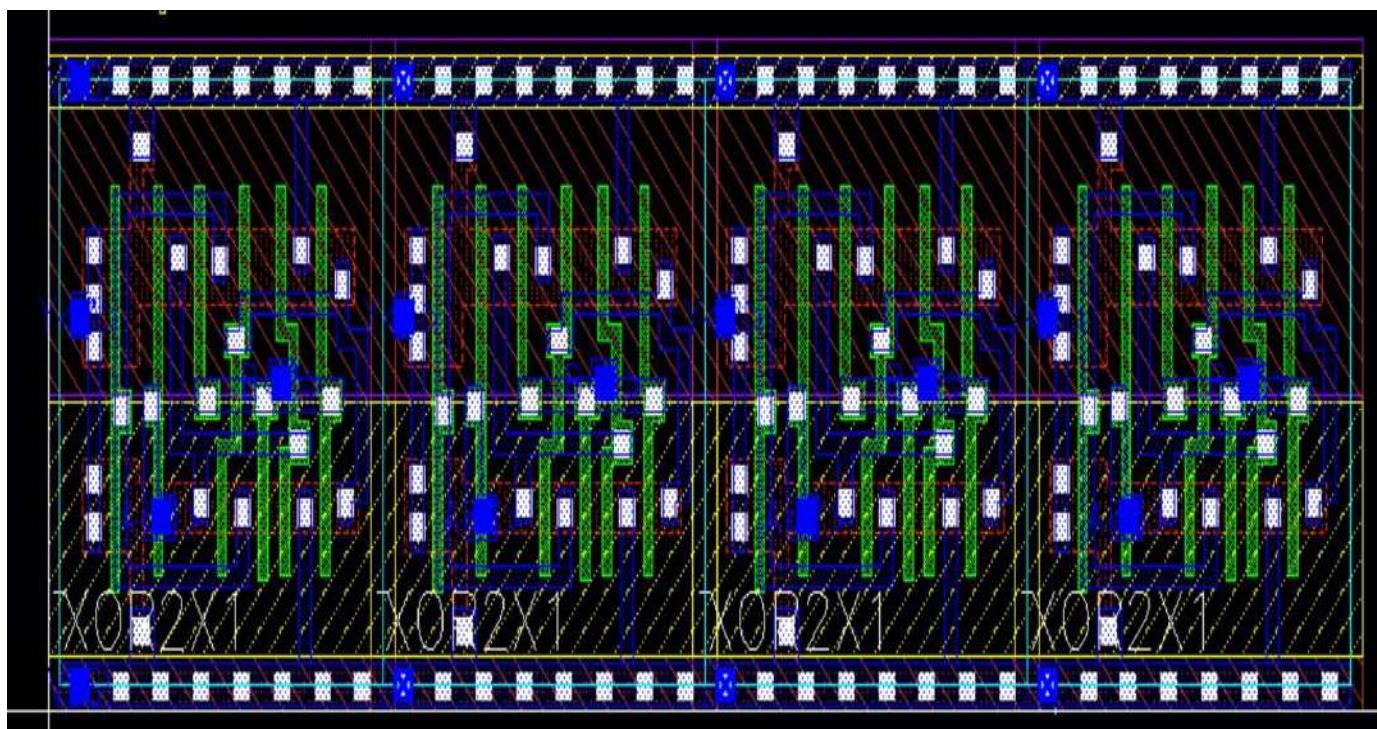
## Bitwise – XOR

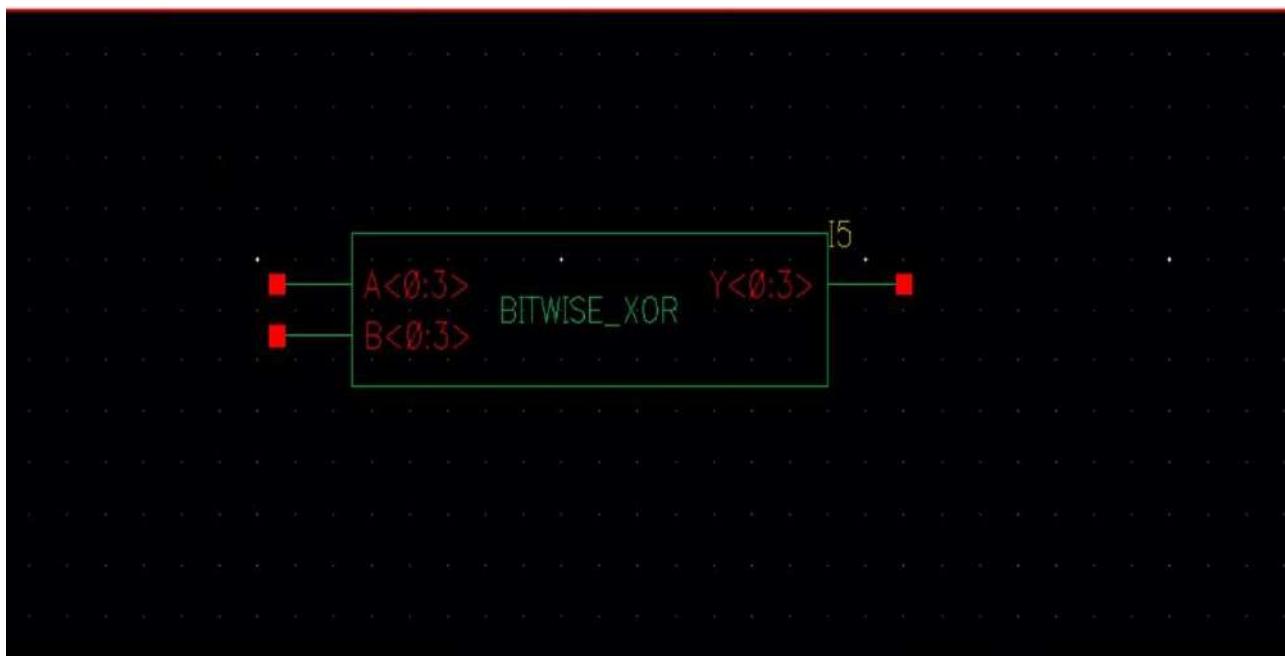
Functionality :  $Y_i = \text{XOR}(A_i, B_i)$

### Schematic



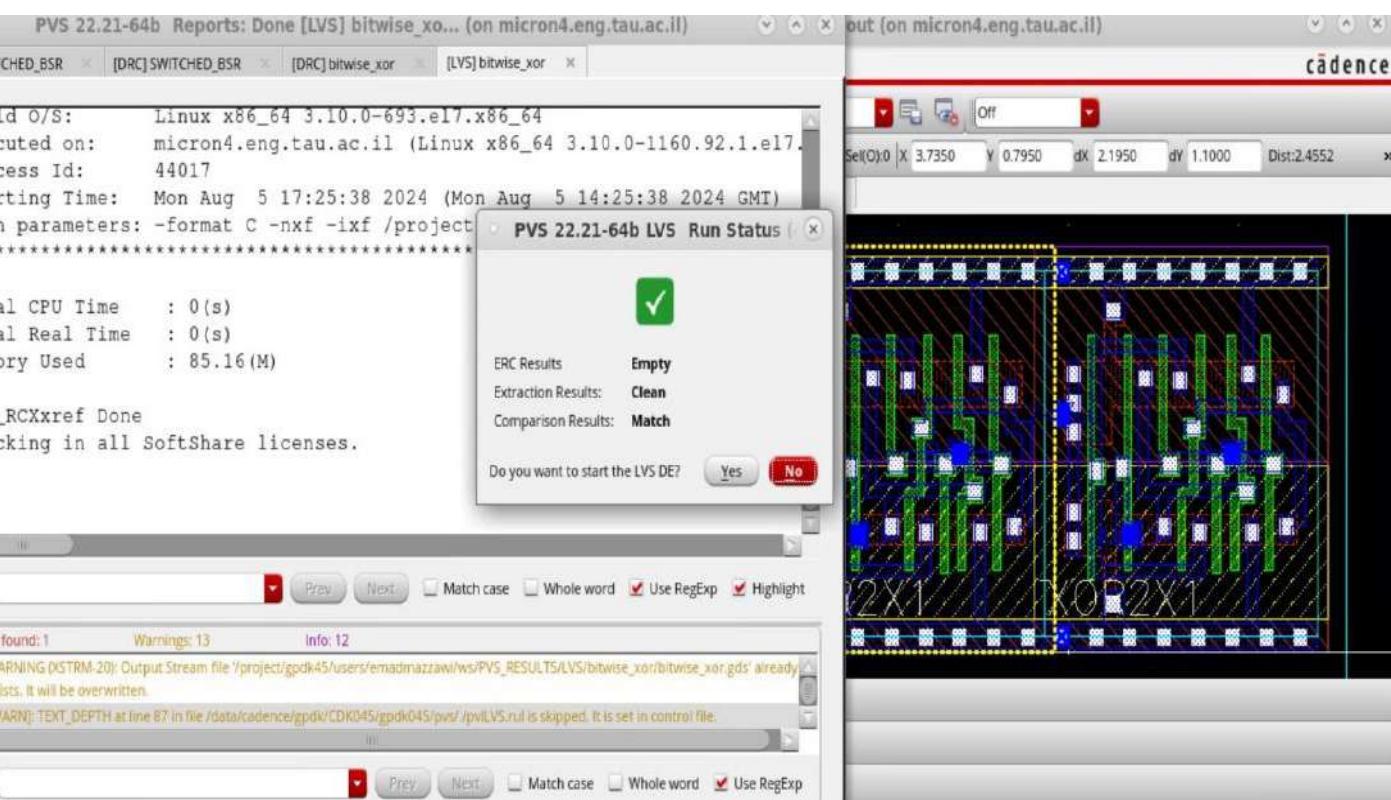
### Symbol & Layout





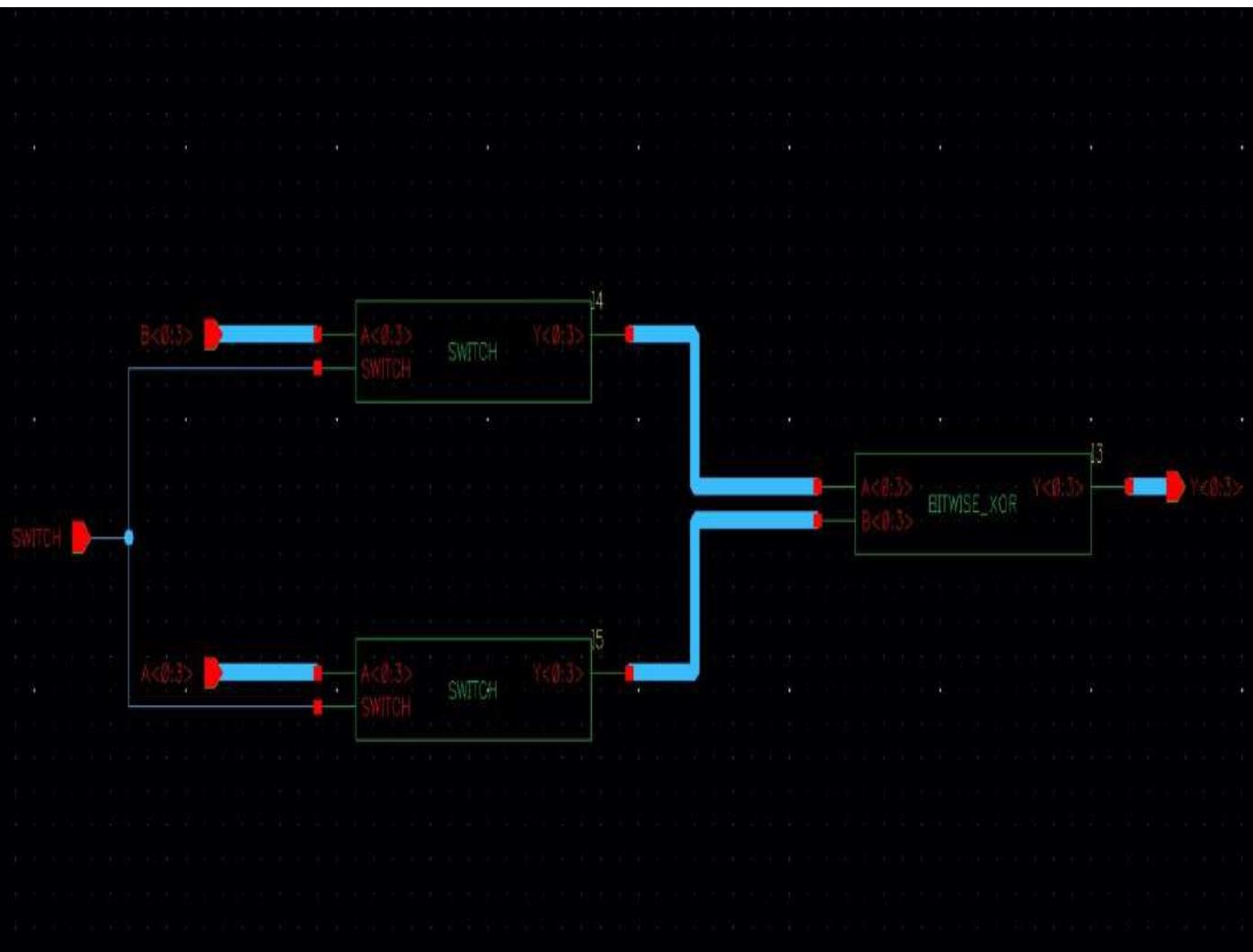
DRC , LVS , QUANTUS



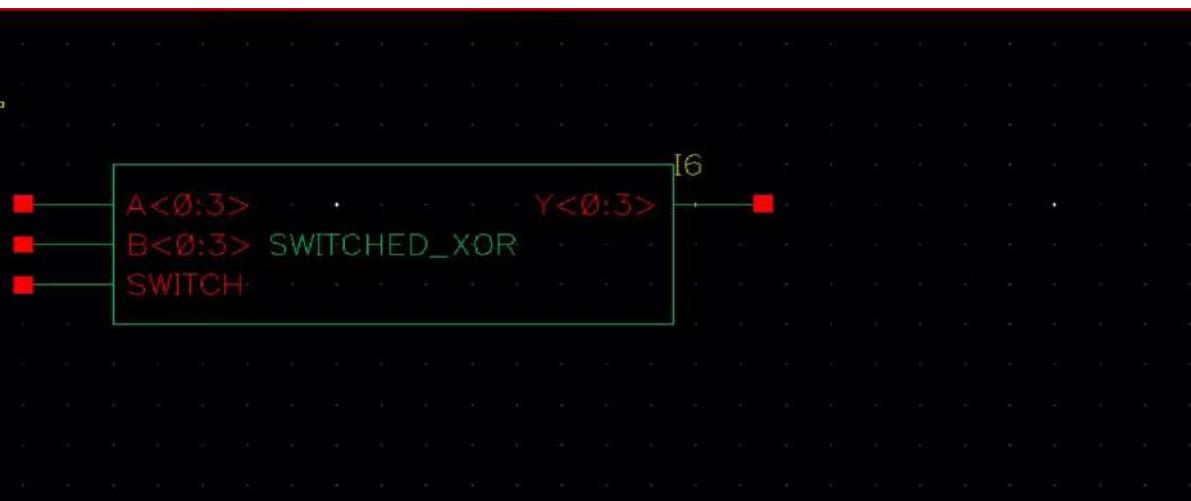


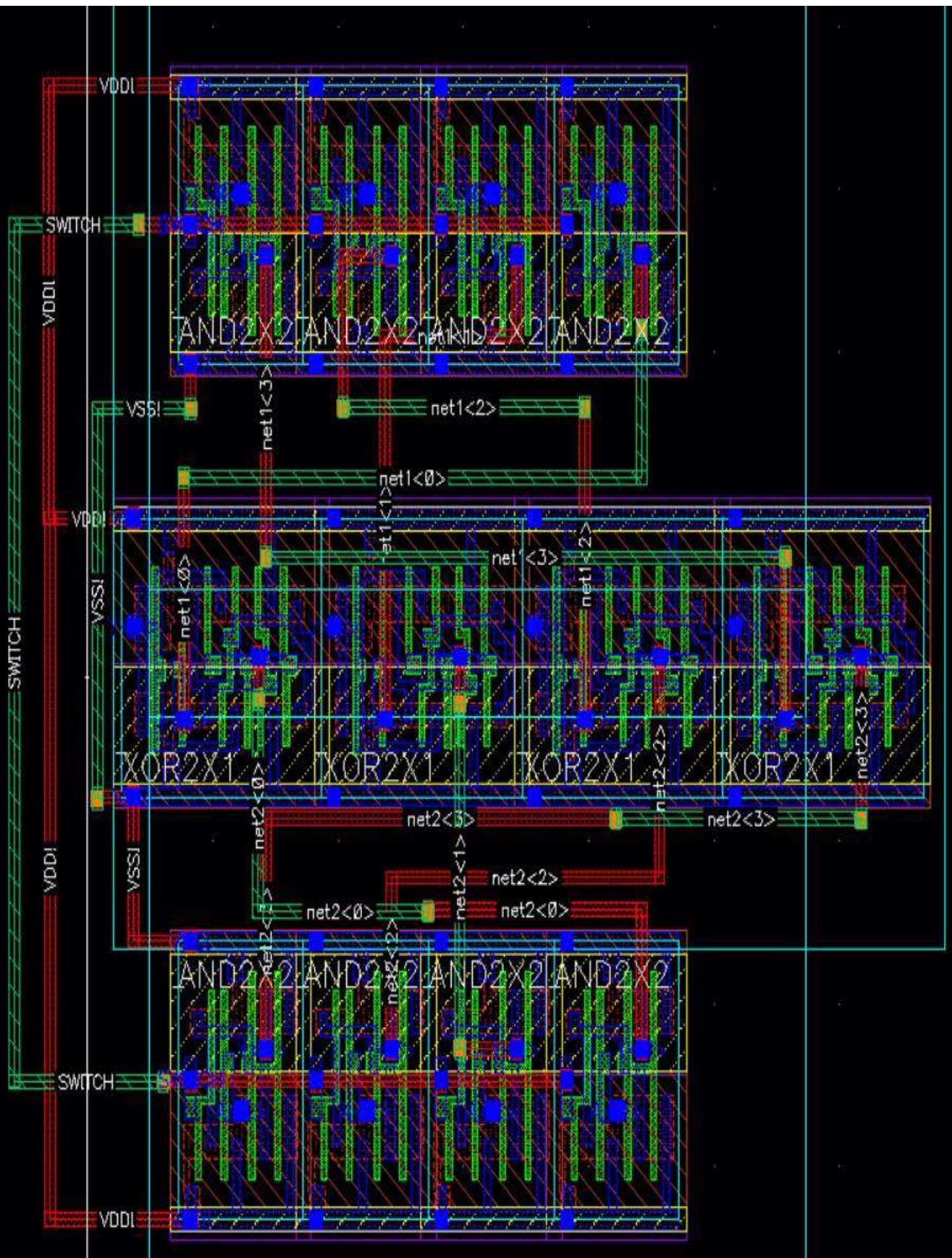
## Switched XOR

### Schematic

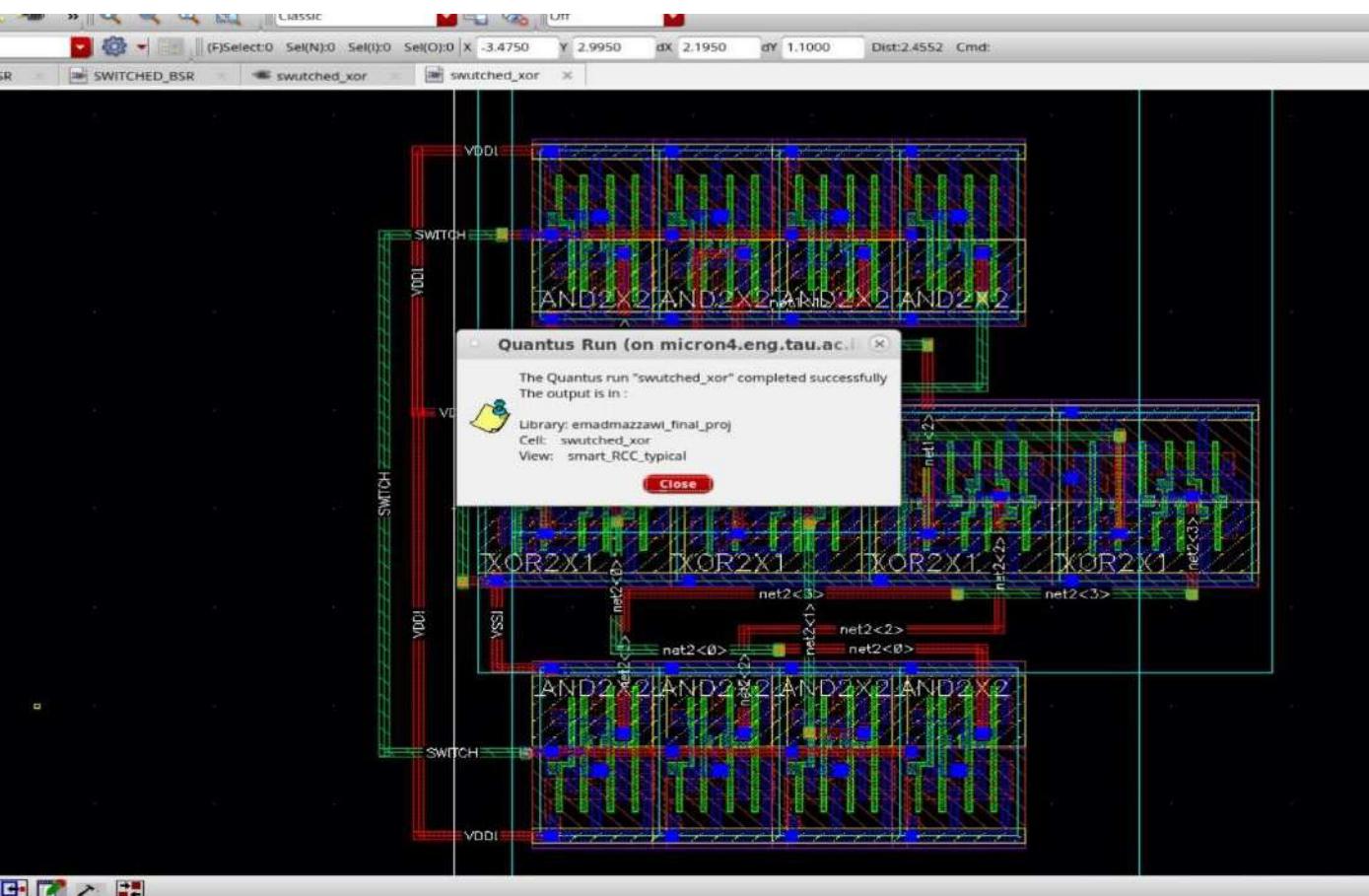
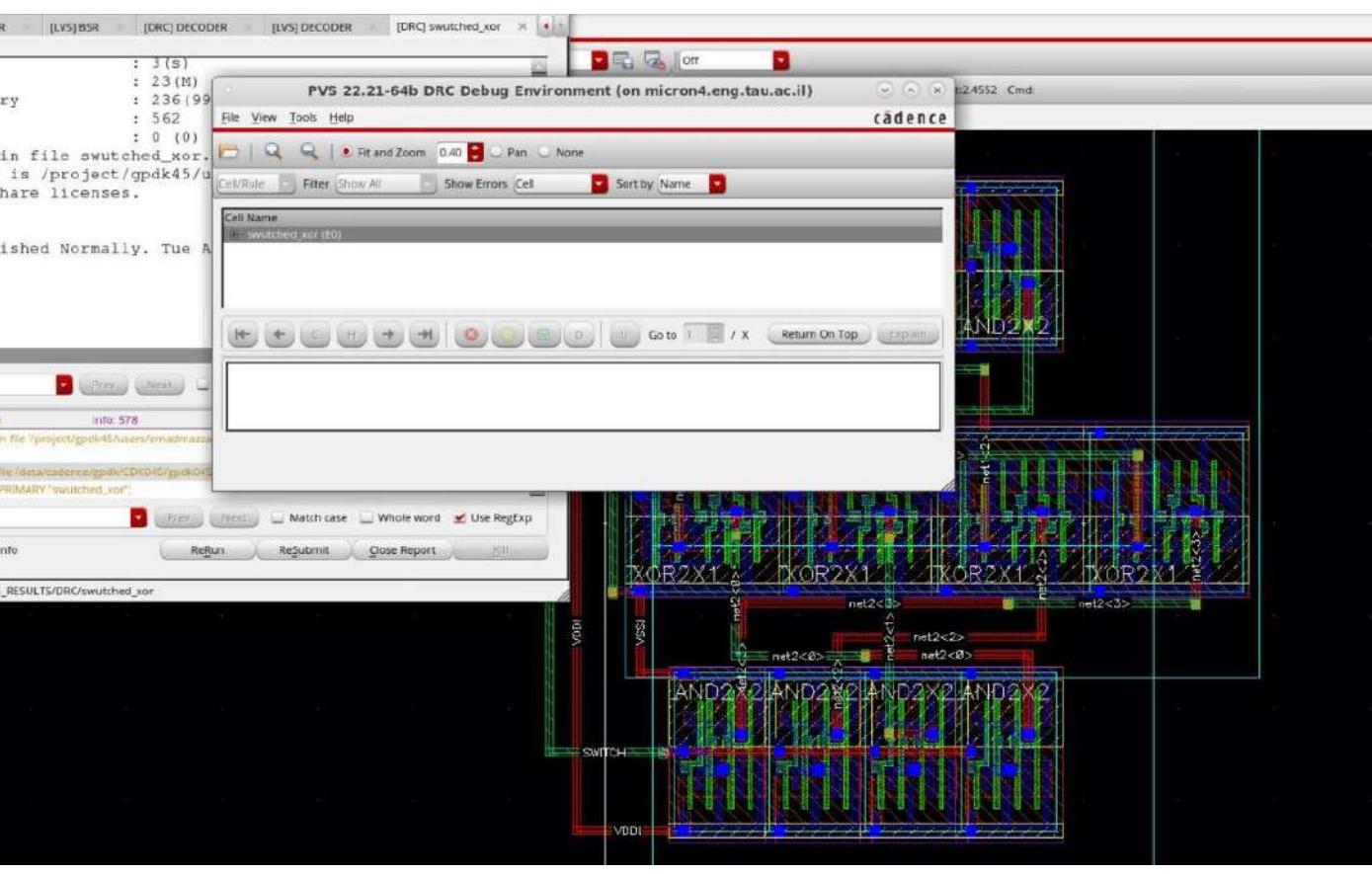


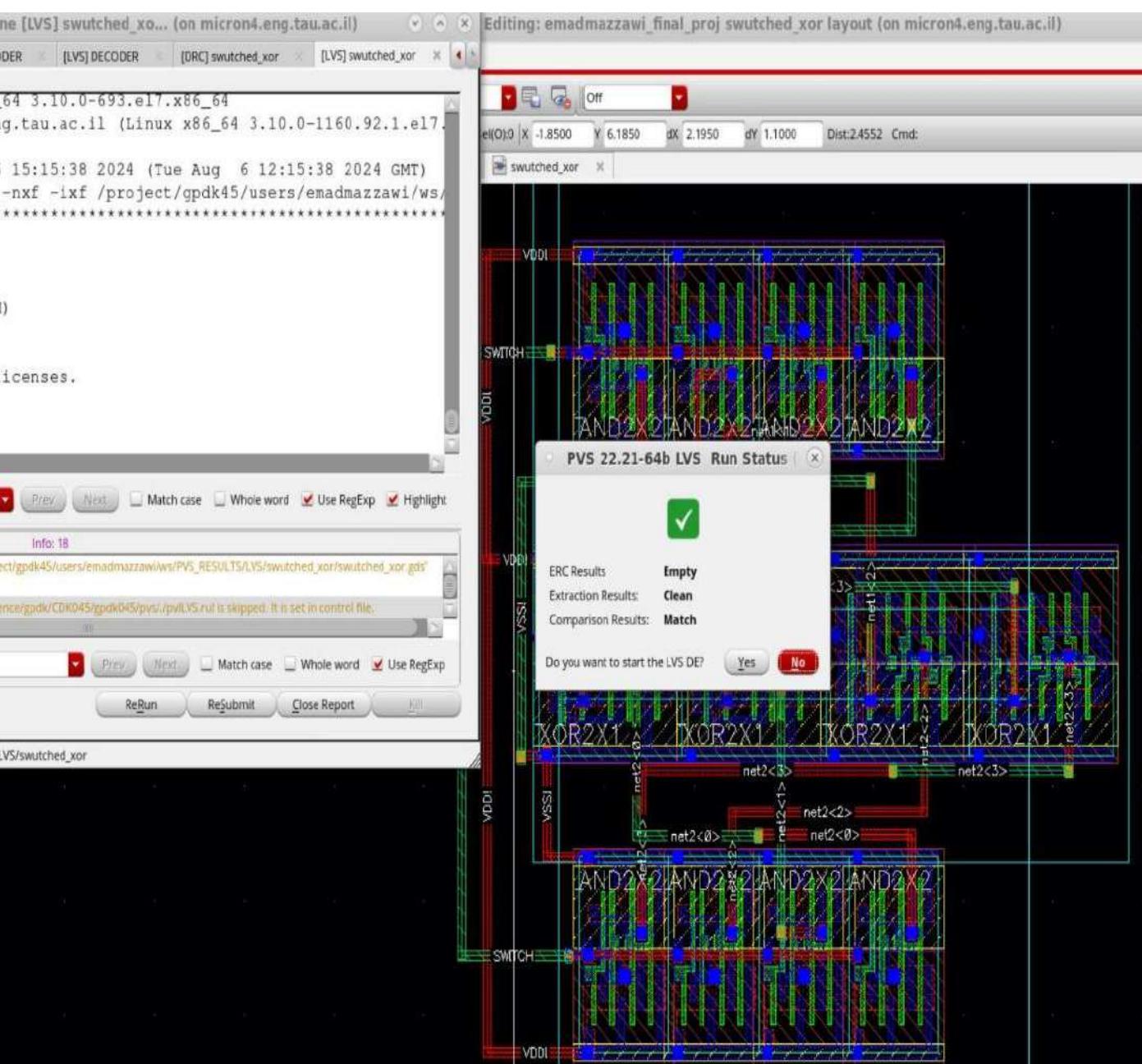
### Symbol & Layout





## DRC , LVS , QUANTUS





## Barrel - Shifter – Right

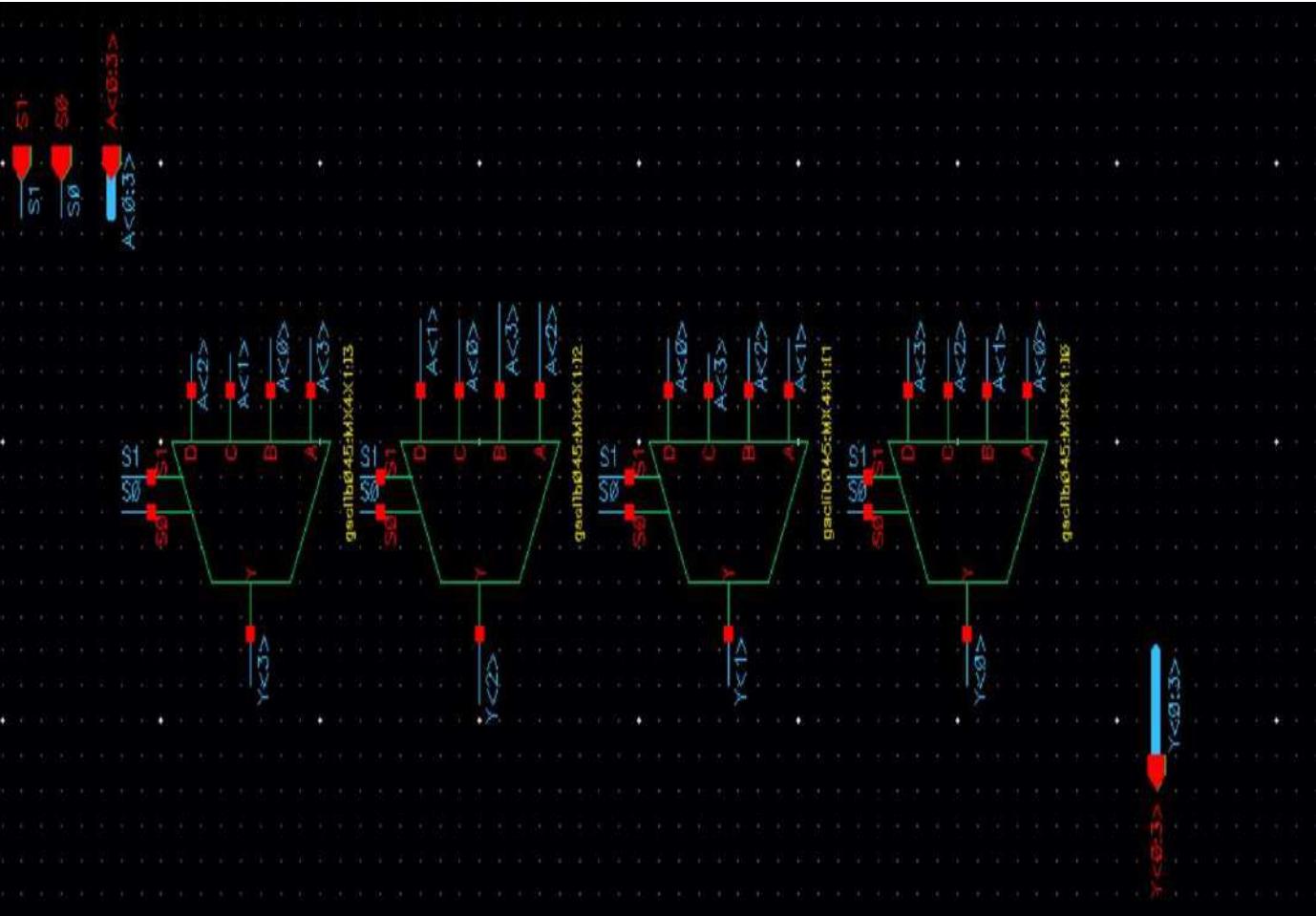
### Functionality

BSR returns a number whose binary representation is the input shifted by specified number of bits to the right , notice that barrel shifting by 4 is the same as not shifting at all.

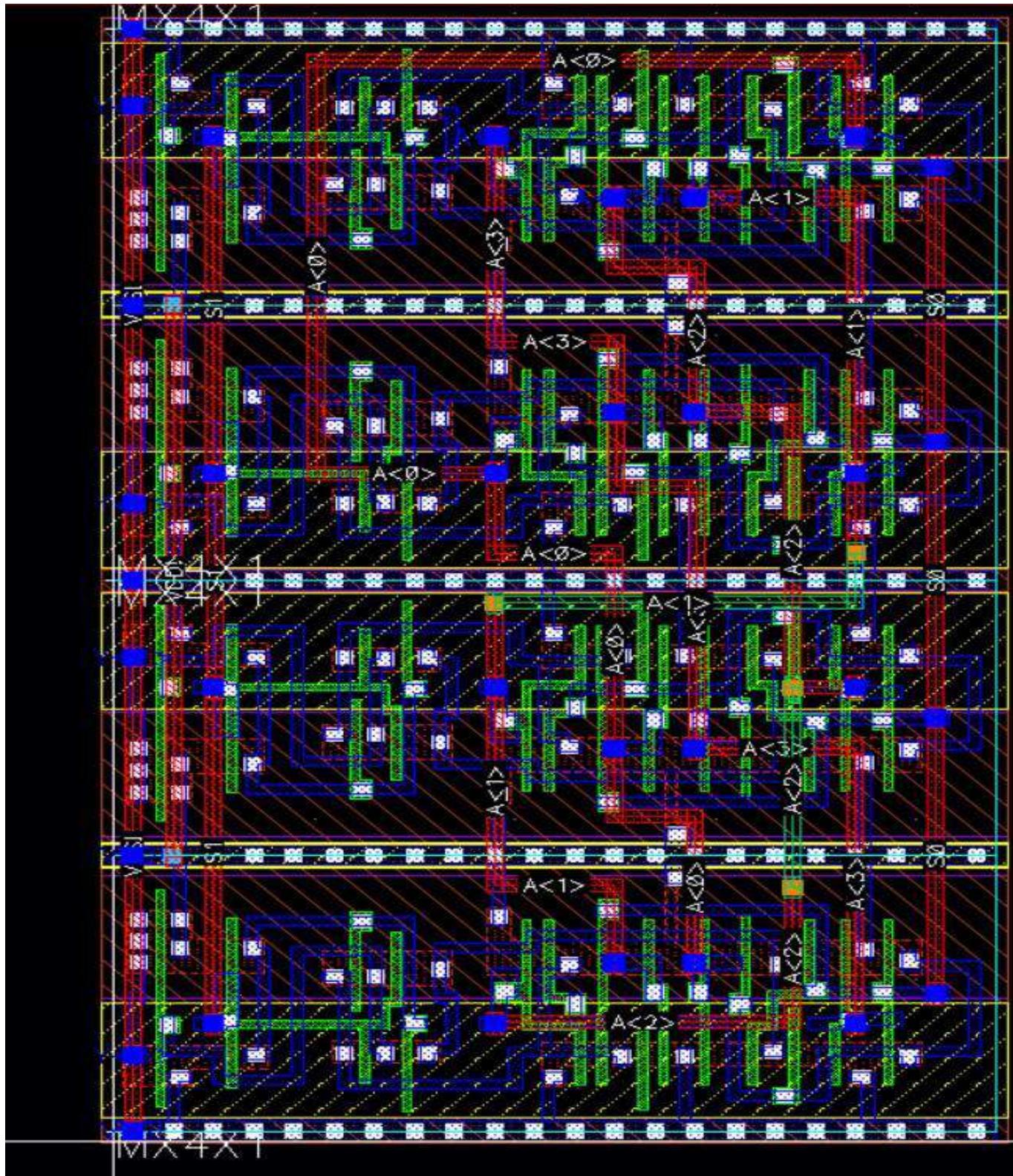
The amount of bits shifted is determined according to the Opcode: bit3 , bit4

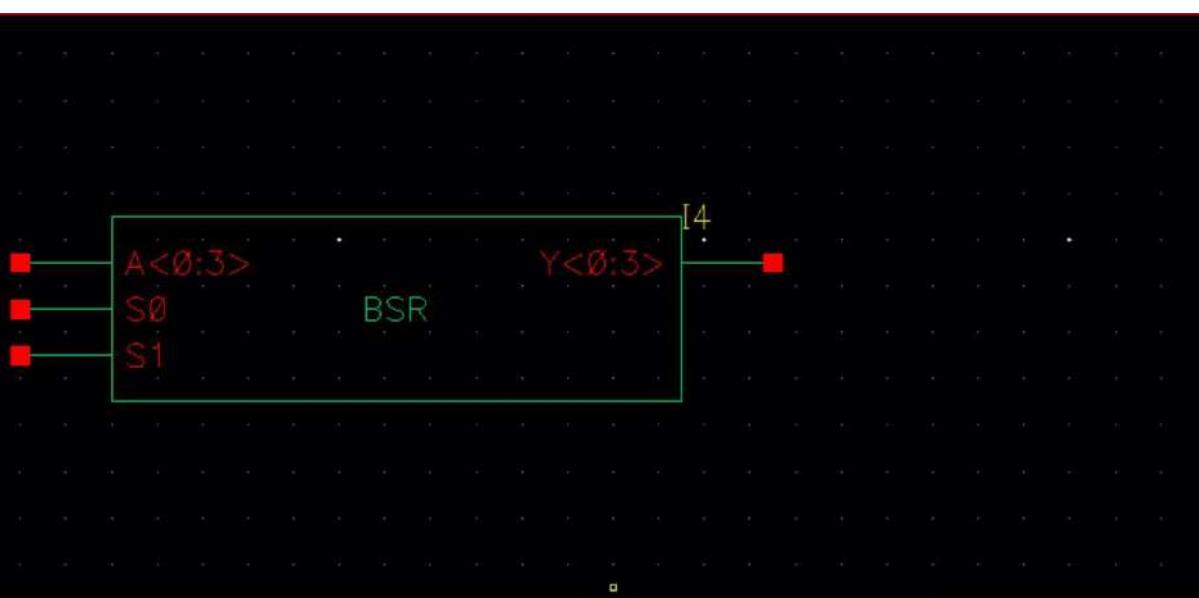
	operation	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
BRS	Shift 4	0	0	1	0	
	Shift 1	0	1	1	0	
	Shift 2	1	0	1	0	
	Shift 3	1	1	1	0	

### Schematic

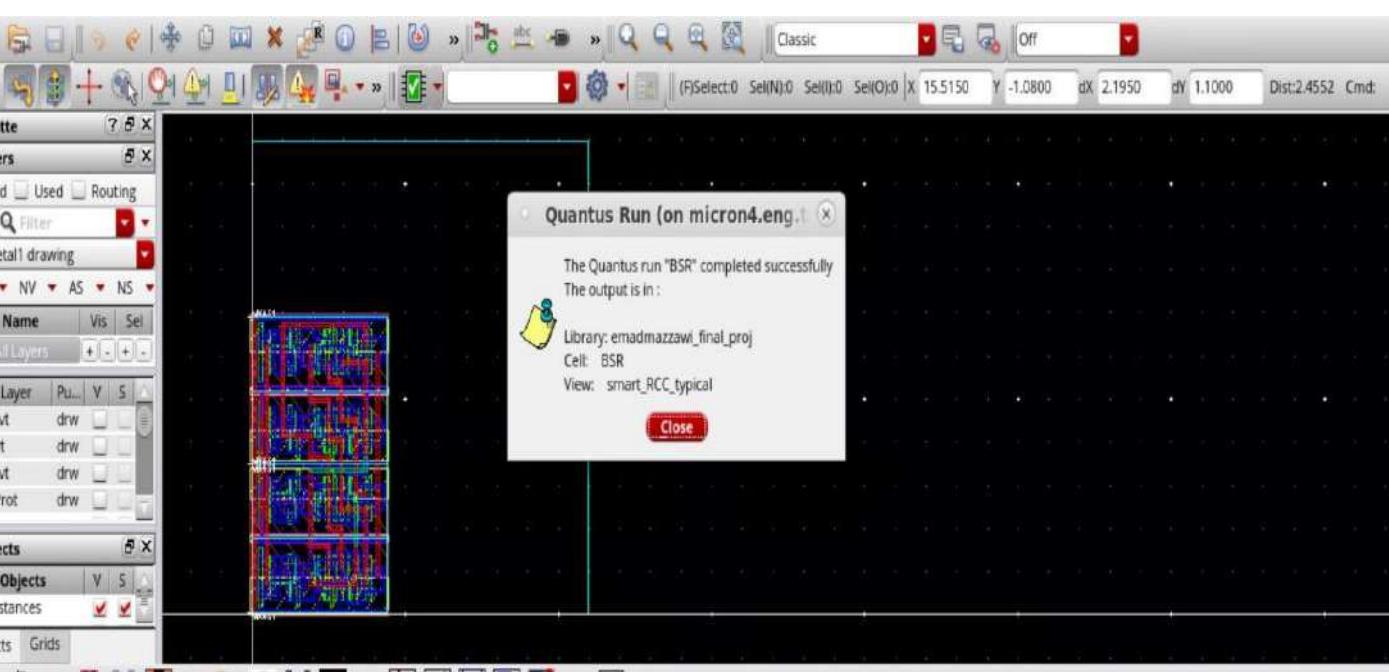
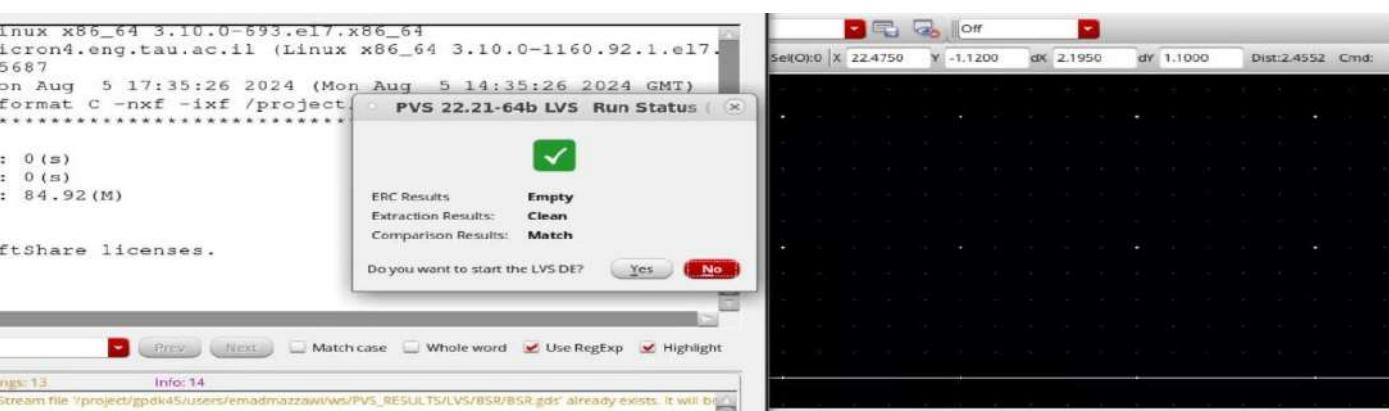


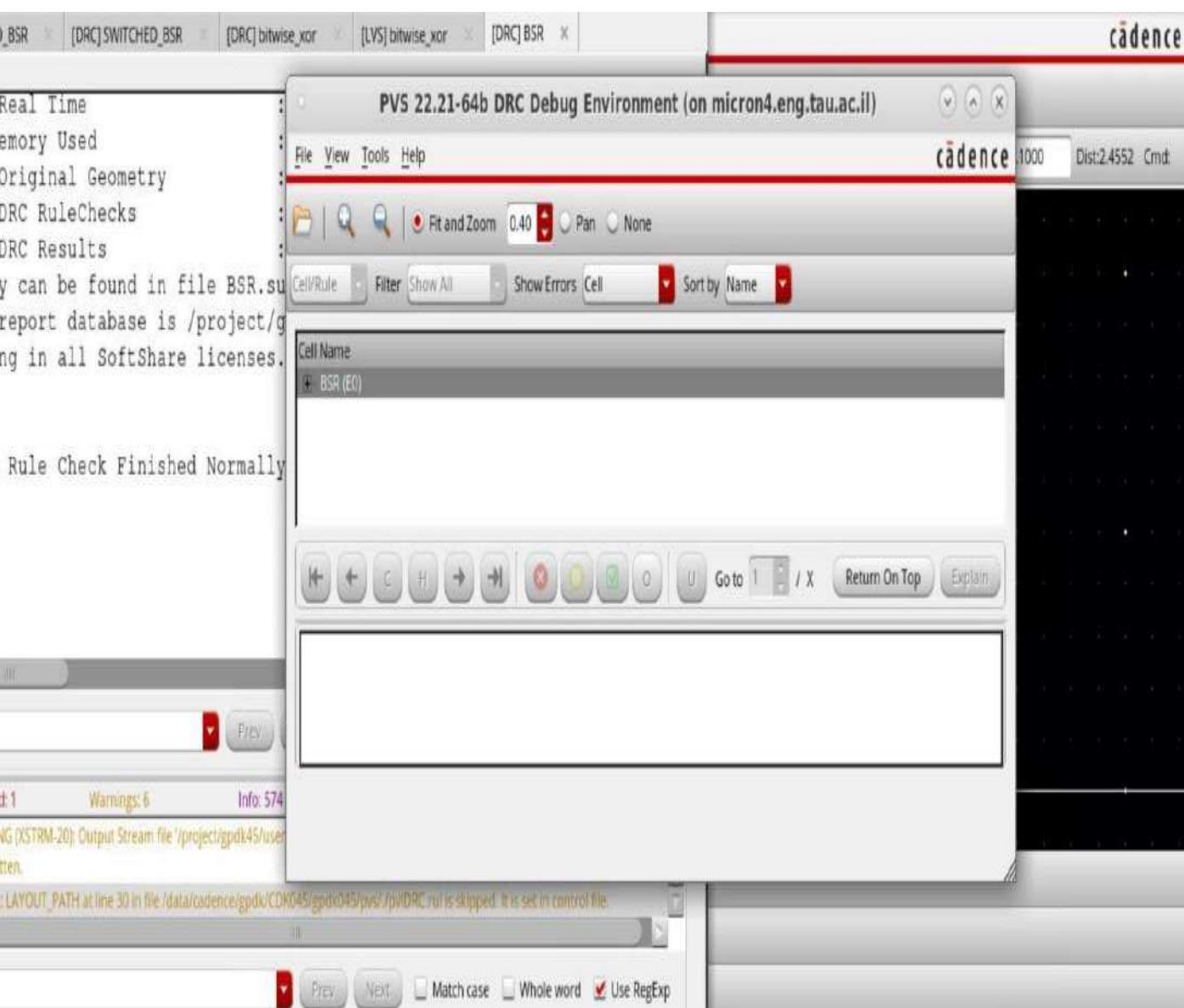
## Symbol & Layout





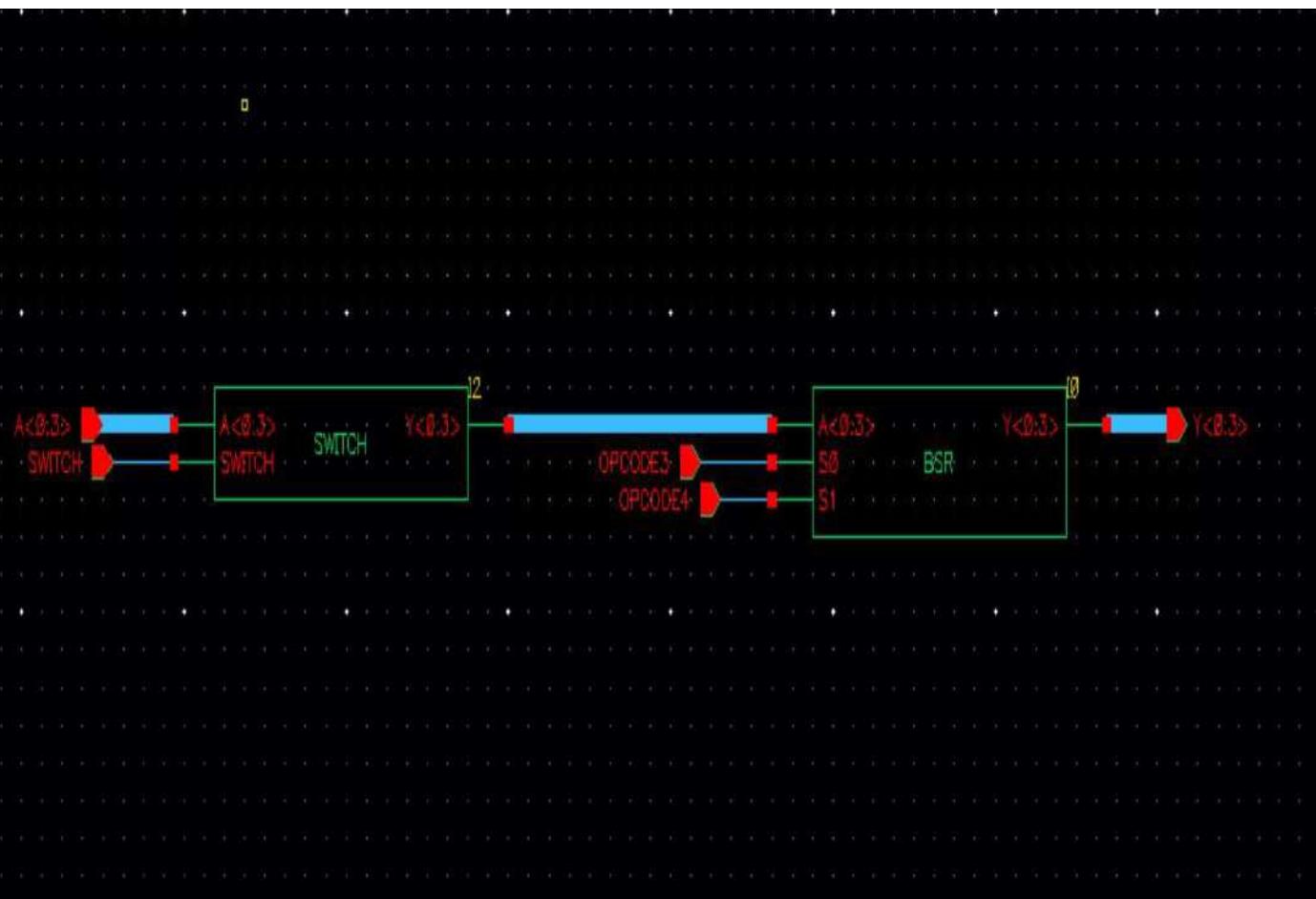
DRC , LVS , QUANTUS



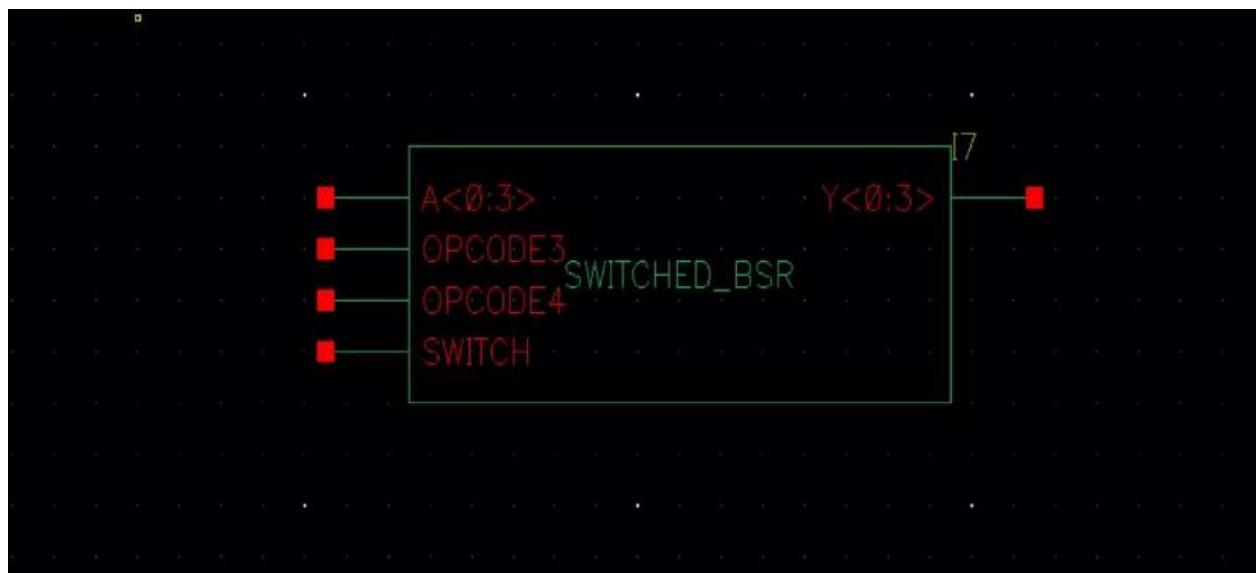


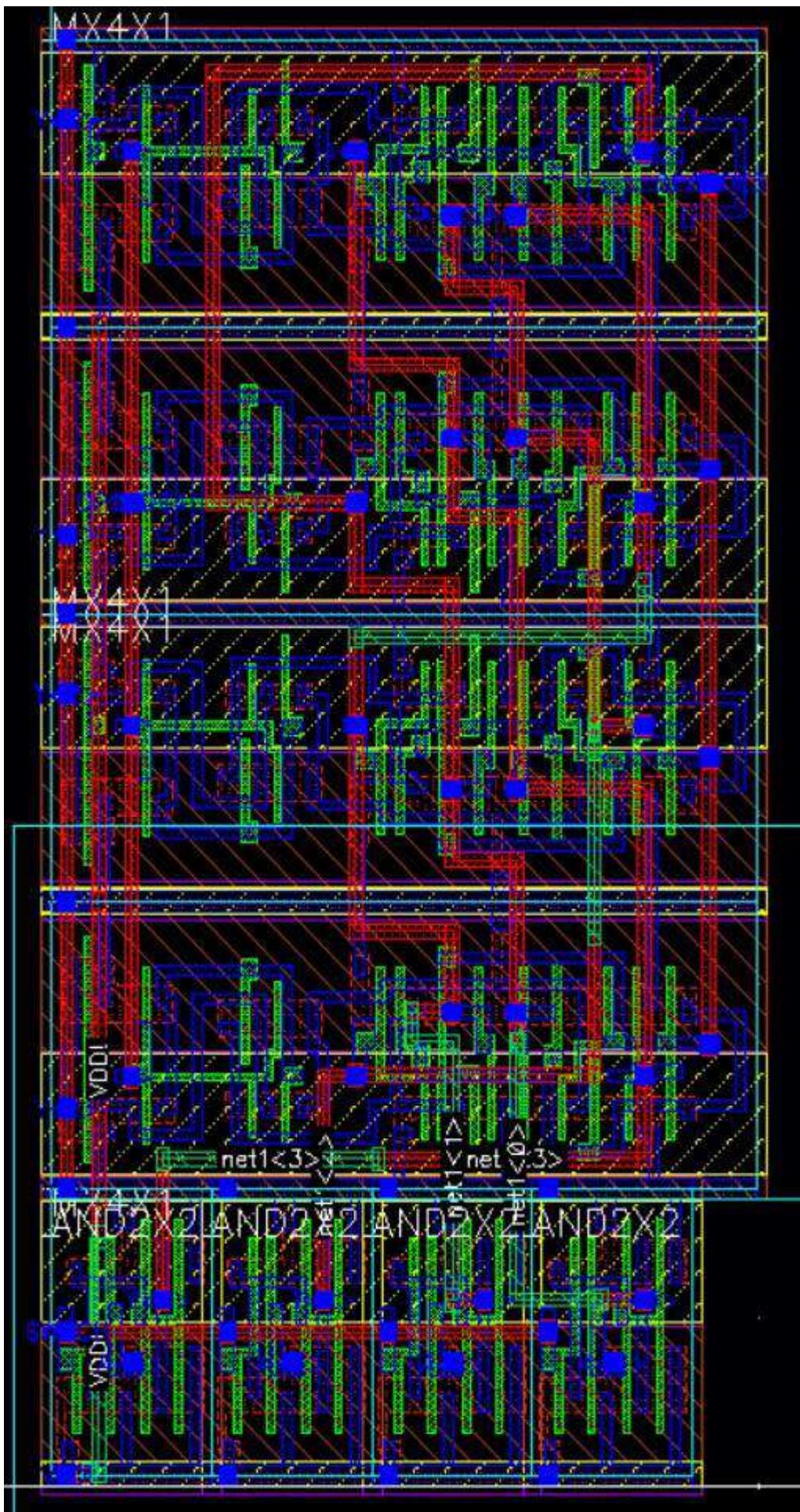
## Switched BSR

### Schematic

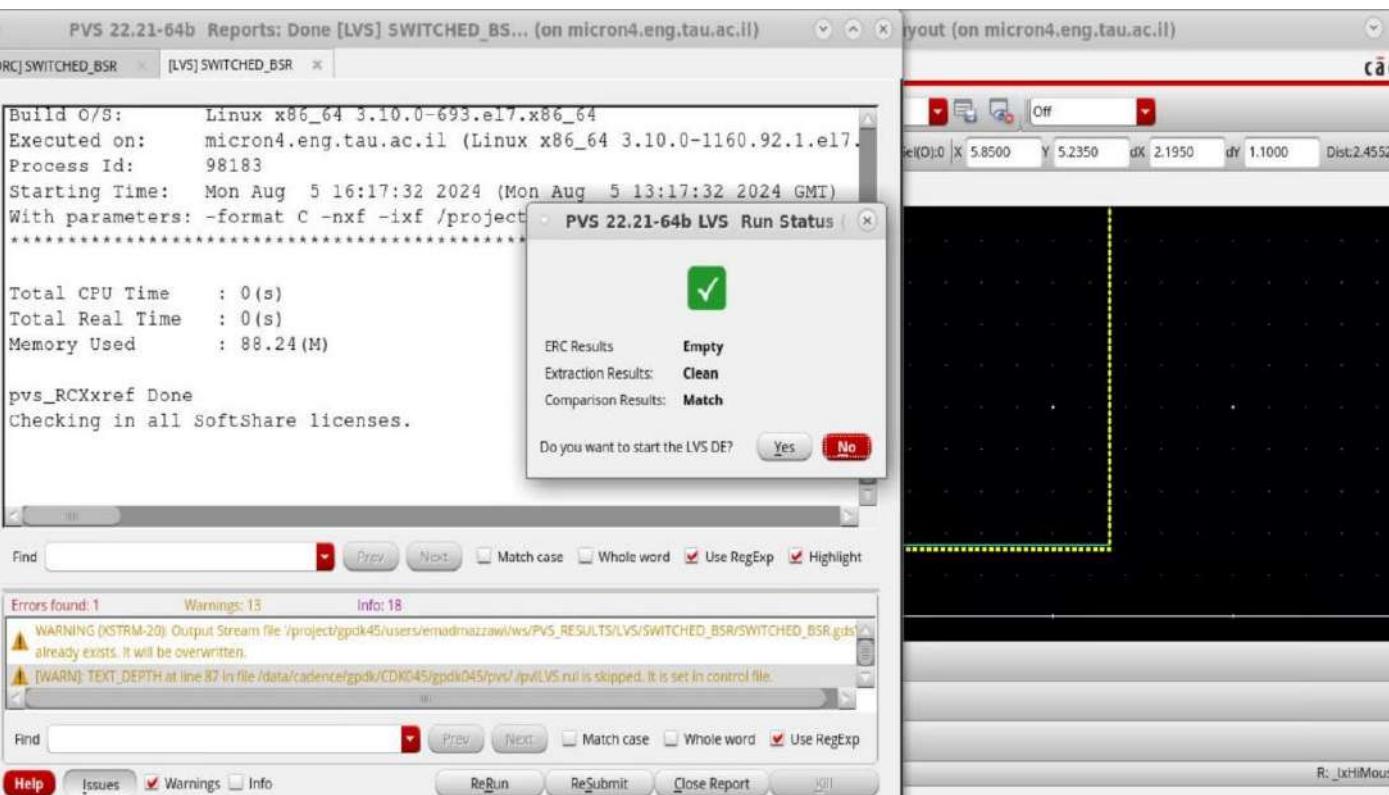
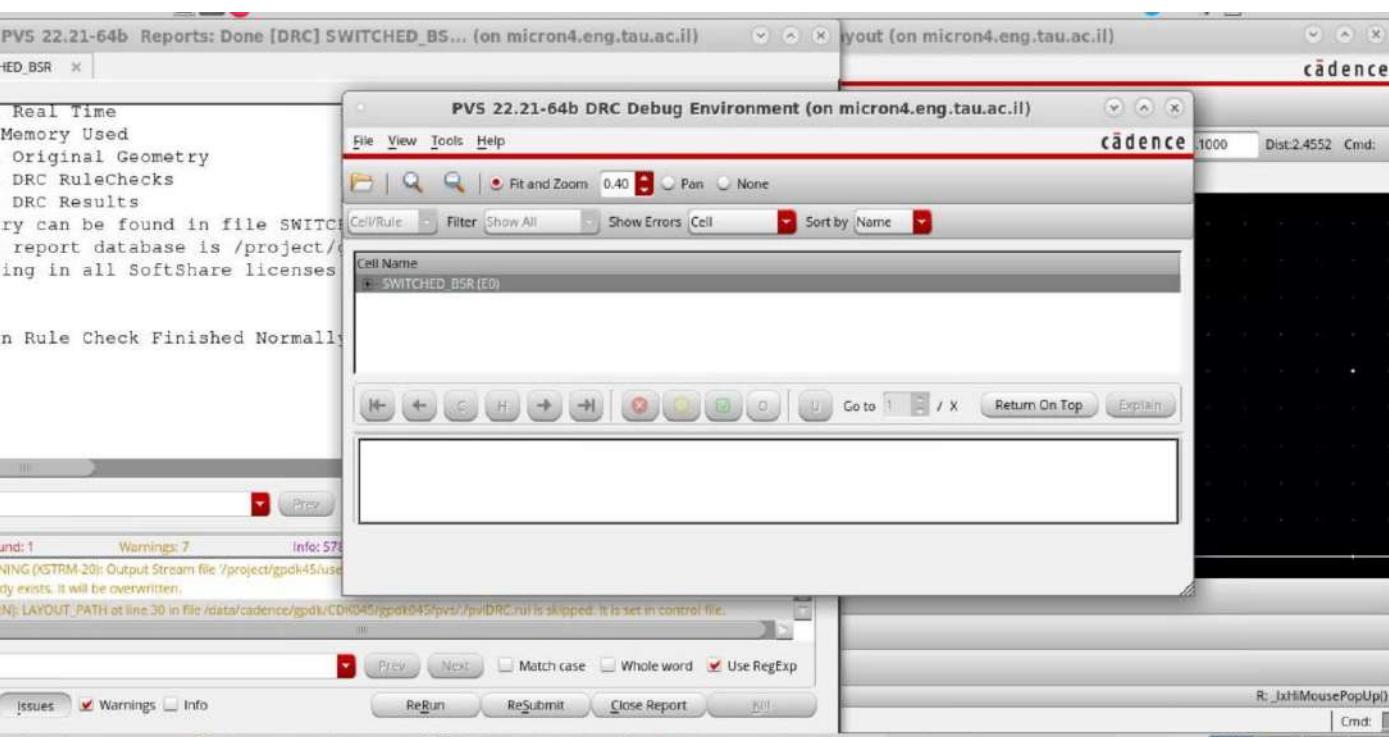


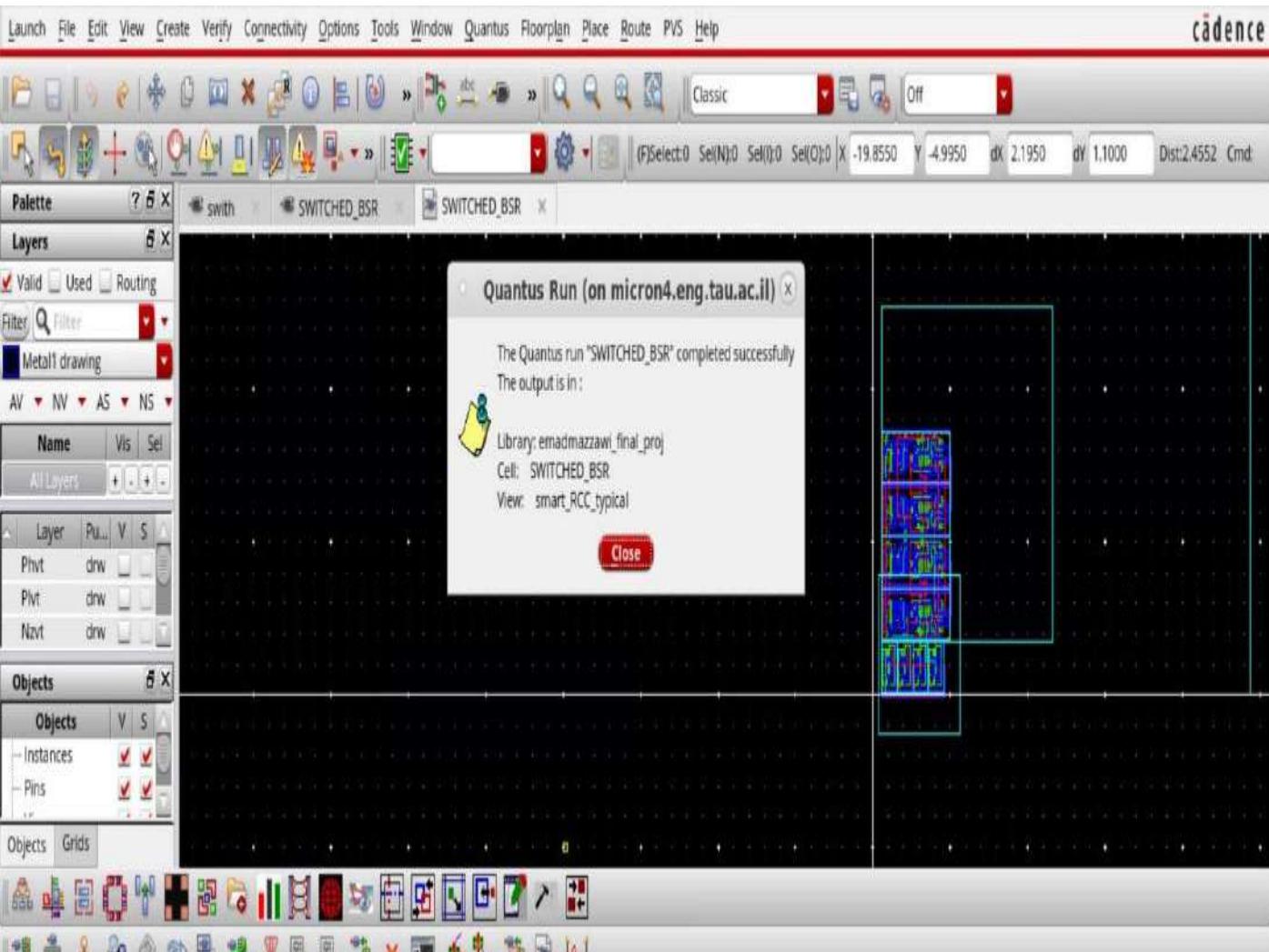
### Symbol & Layout





## DRC , LVS , QUANTUS

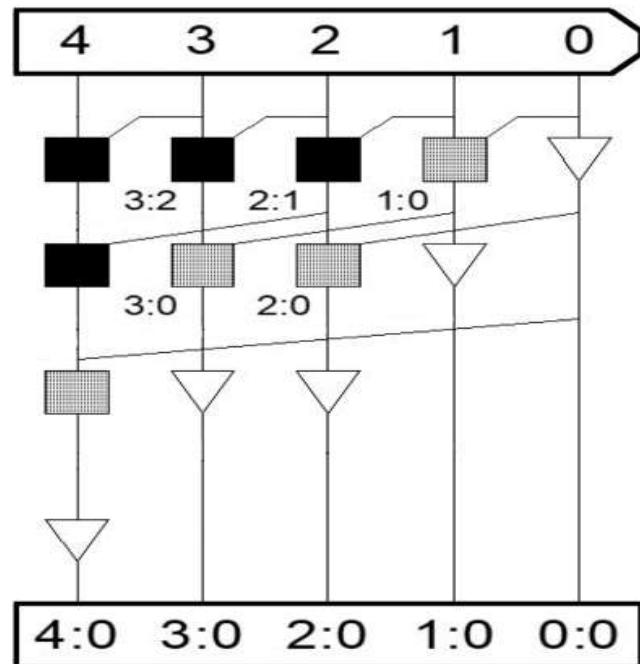




## ADDER

We implemented 4-bit Knowels-Adder , a parallel form of carry look-ahead adder. It generates the carry signals on  $O(\log N)$  time; which can be considered one of the fastest adder designs.

We built the black-cell, grey-cell and the GP-cell separately, and used the buffer from gsclib045 then connected them according to the schematic below:

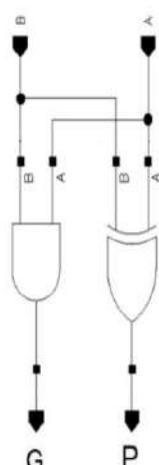
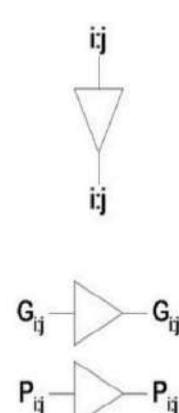
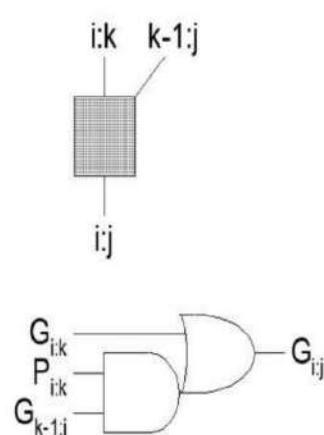
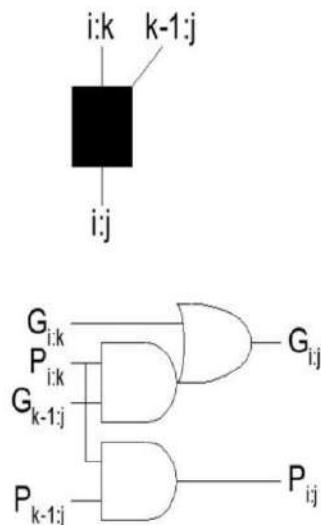


black-cell

grey-cell

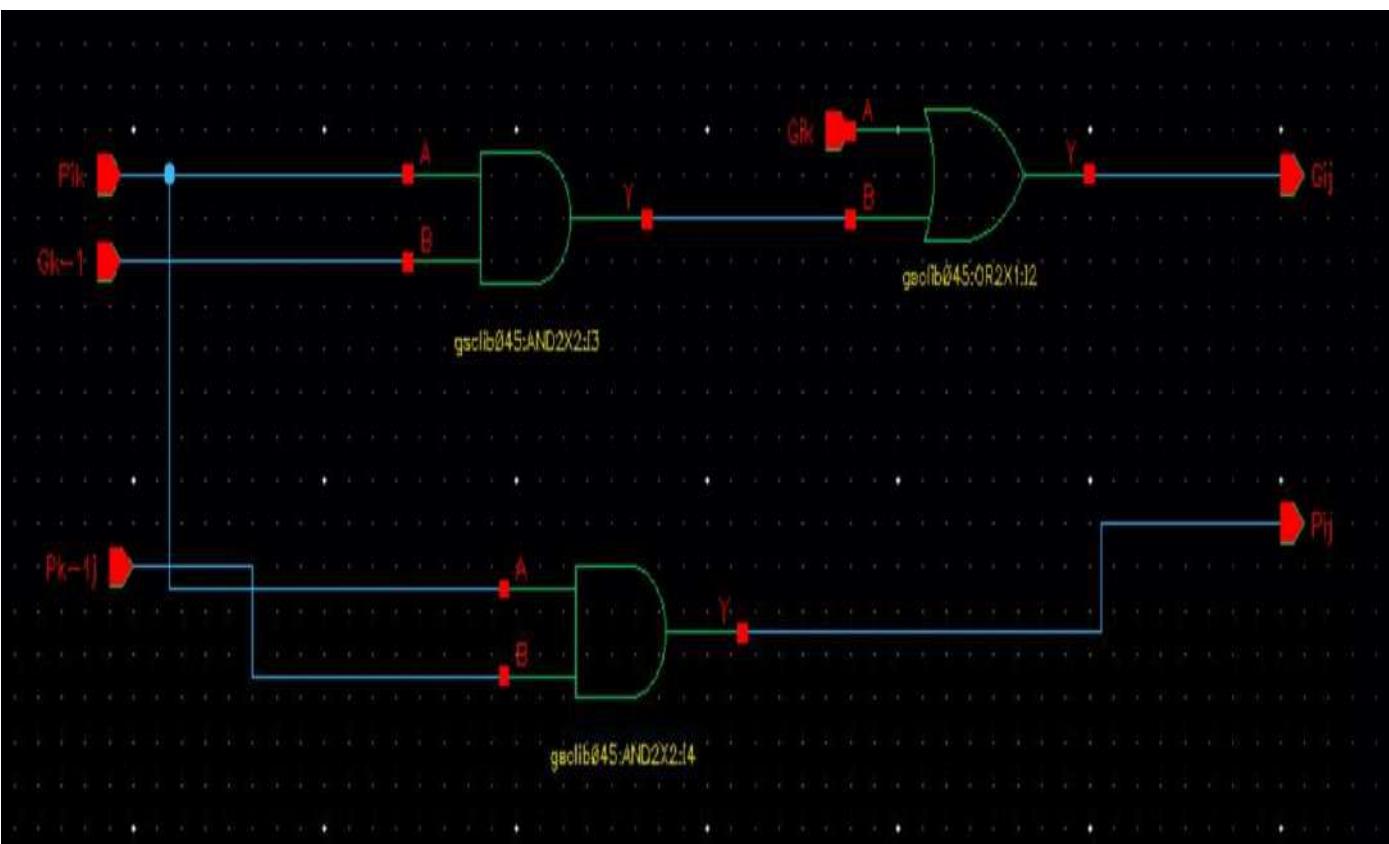
Buffer

GP-cell

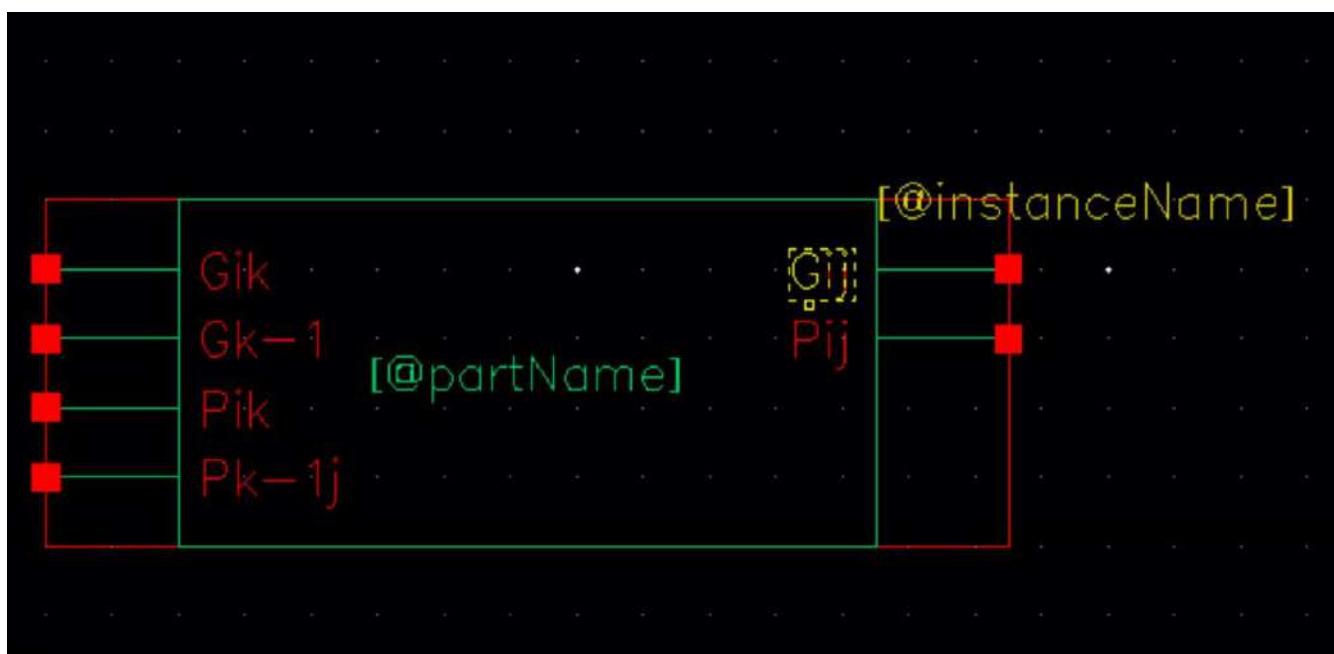


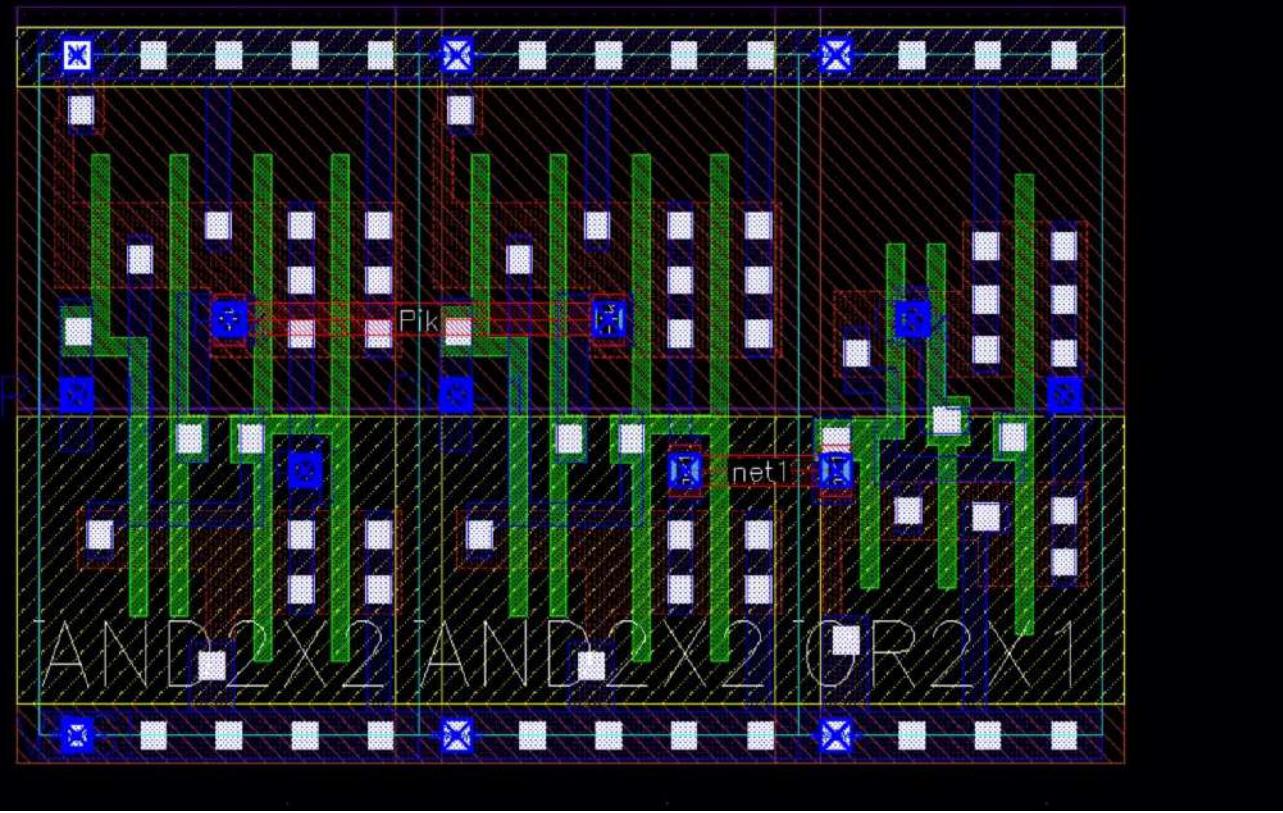
### Black-cell

#### Schematic



#### Symbol & Layout





## DRC , LVS , QUANTUS

Build O/S: Linux x86\_64 3.10.0-693.el7.x86\_64  
 Executed on: micron4.eng.tau.ac.il (Linux x86\_64 3.10.0-1160.92.1.el7.x86\_64)  
 Process Id: 71177  
 Starting Timer: Wed Aug 7 15:21:29 2024 (Wed Aug 7 12:21:29 2024 GMT)  
 With parameters: -format C -nxfs -ixfs /project/gpdk45/users/emilnassra/ws/PVS\_RESULTS

Total CPU Time : 0(s)  
 Total Real Time : 0(s)  
 Memory Used : 86.80(M)

pvs\_RCXref Done  
 Checking in all SoftShare licenses.

Errors found: 1      Warnings: 12      Info: 15

ERC Results: Empty  
 Extraction Results: Clean  
 Comparison Results: Match

PVS 22.21-64b LVS Run Status

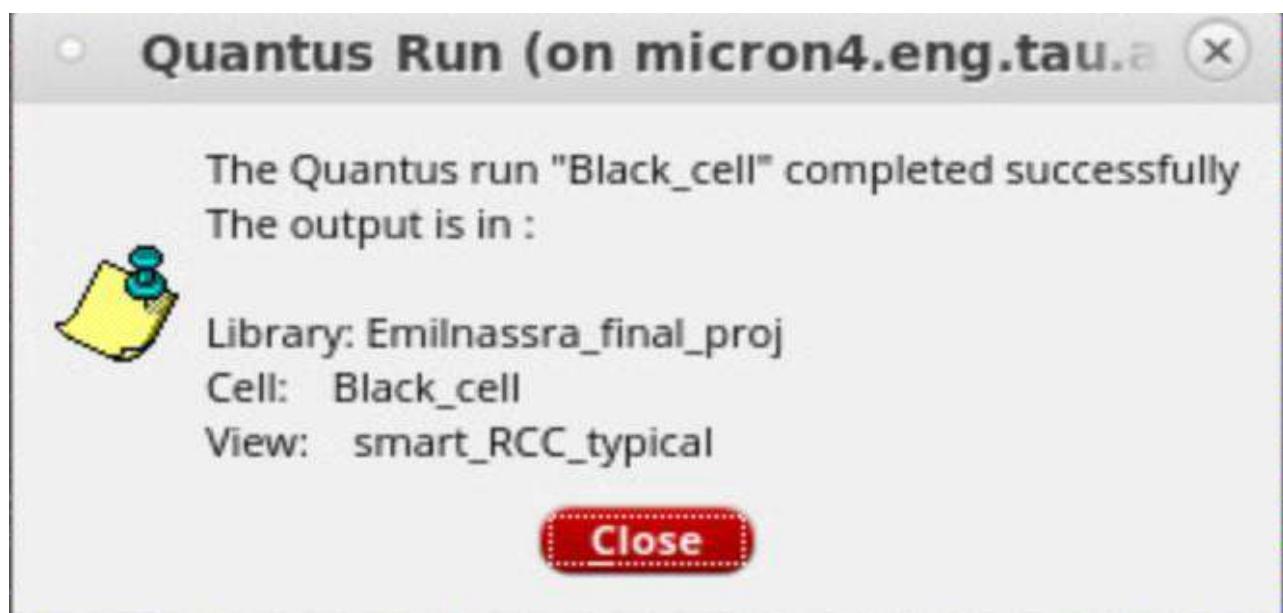
Do you want to start the LVS DE? Yes No

/project/gpdk45/users/emilnassra/ws/PVS\_RESULTS/LVS/Black\_cell

Objects

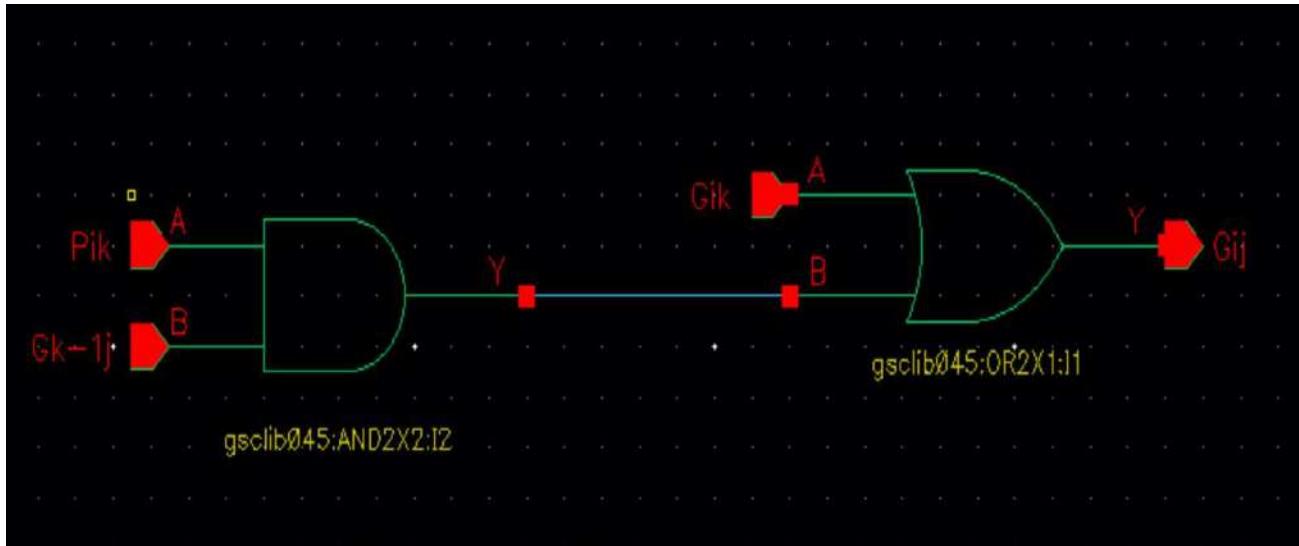
Objects	V	S
Instances	✓	✓
Pins	✓	✓
Vias	✓	✓

Objects Grids

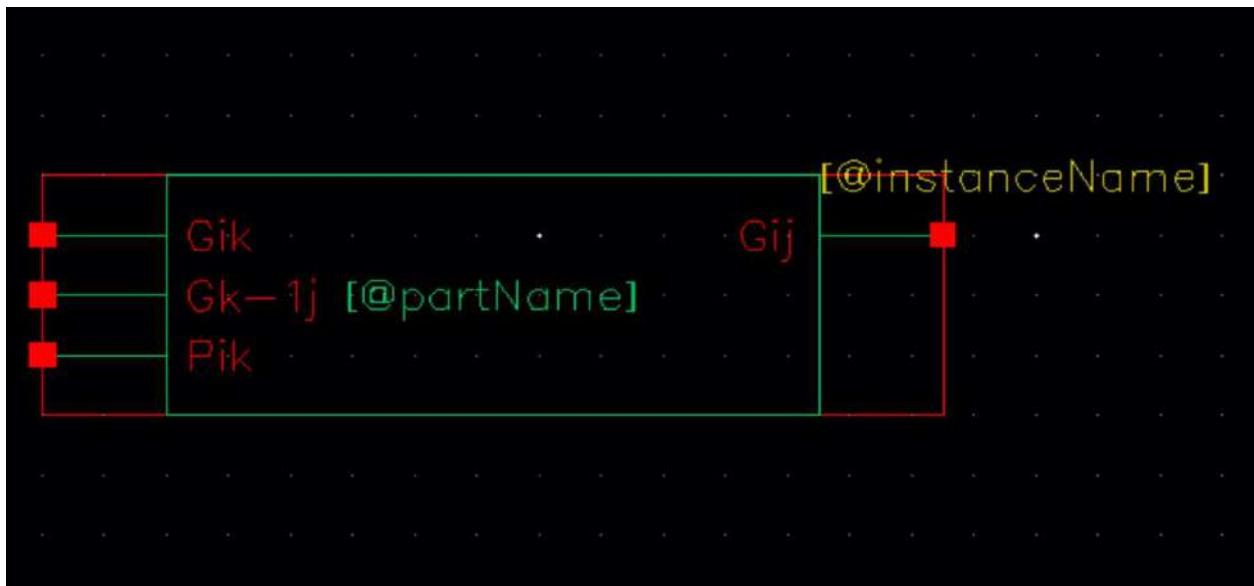


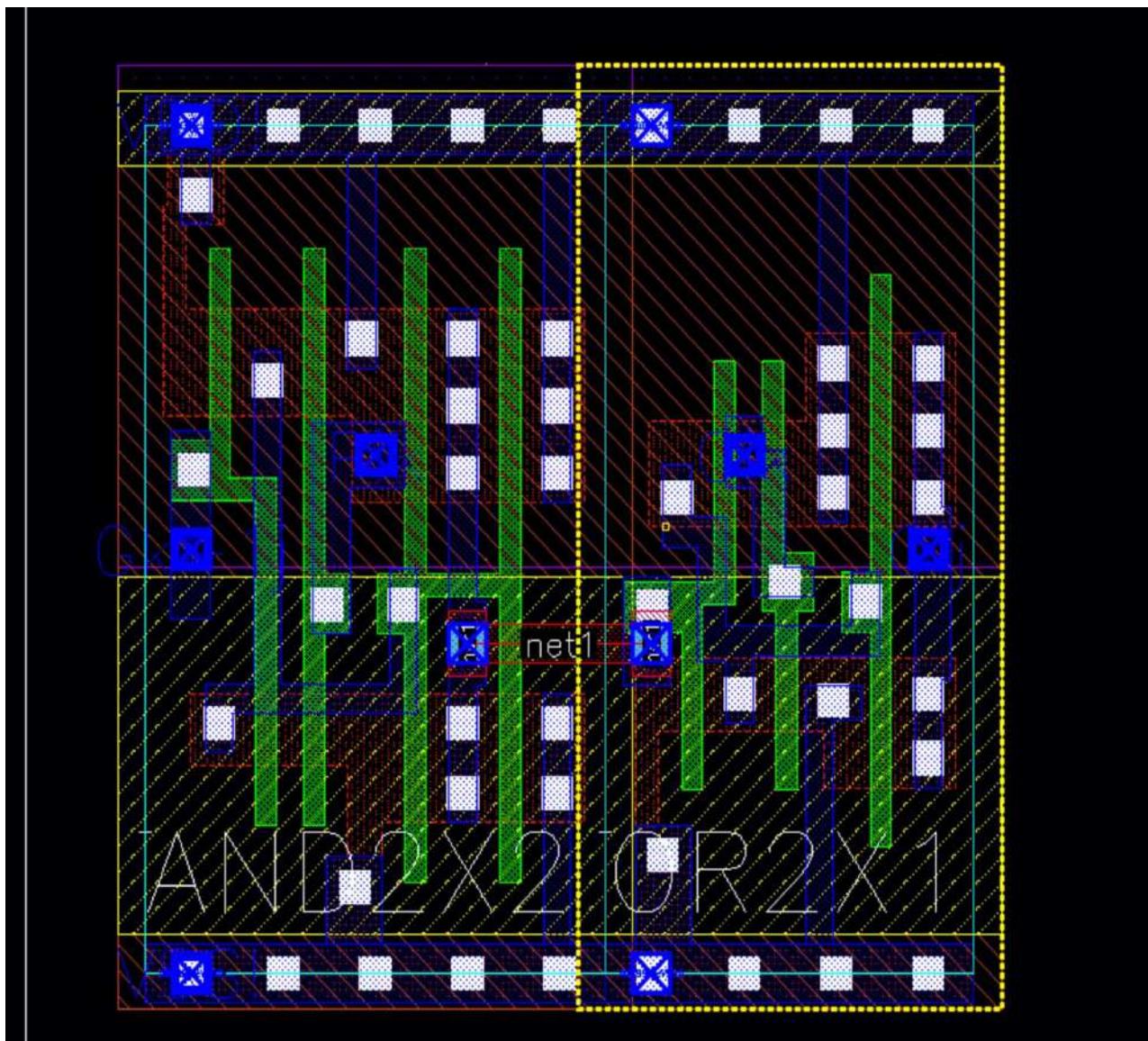
## Grey-cell

### Schematic

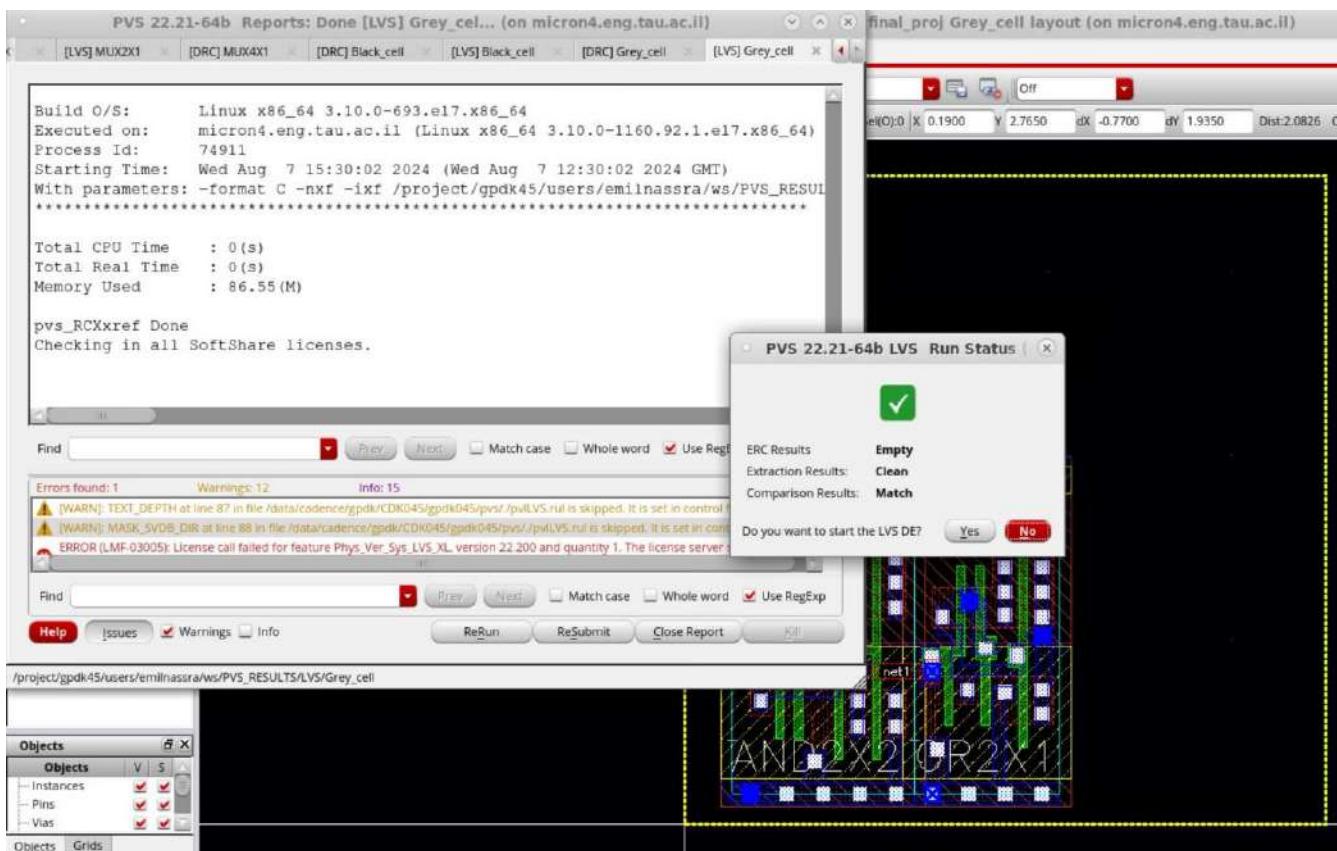


### Symbol & Layout





## DRC , LVS , QUANTUS



## Quantus Run (on micron4.eng.tau.ac.il)

The Quantus run "Grey\_cell" completed successfully  
The output is in :



Library: Emilnassra\_final\_proj  
Cell: Grey\_cell  
View: smart\_RCC\_typical

**Close**

# PVS 22.21-64b DRC Debug Environment (on micron4.eng.tau.ac.il)

cadence

File View Tools Help

Fit and Zoom 0.40 Pan None

Cell/Rule Filter Show All Show Errors Cell Sort by Name

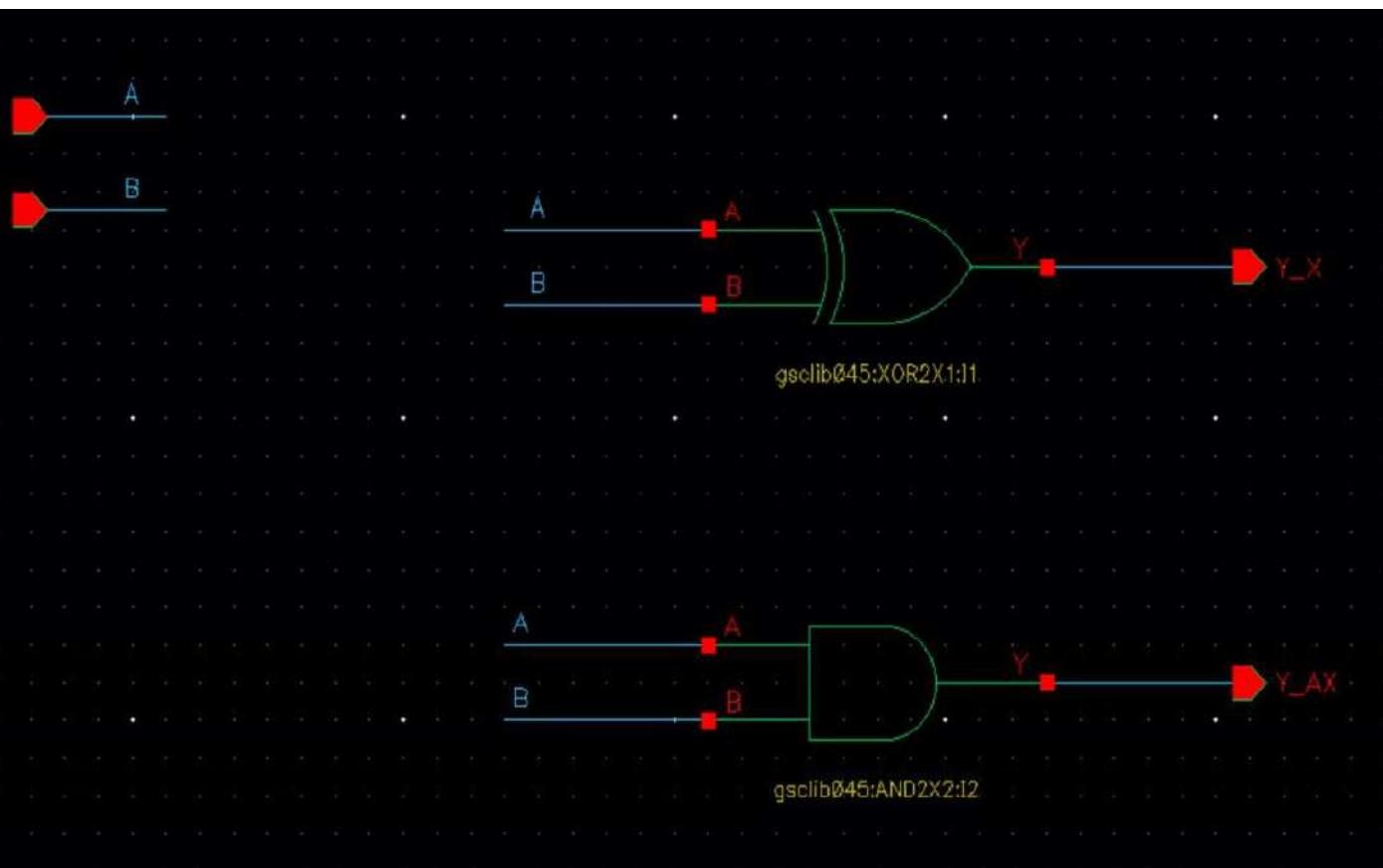
Cell Name

+ Grey\_cell (E0)

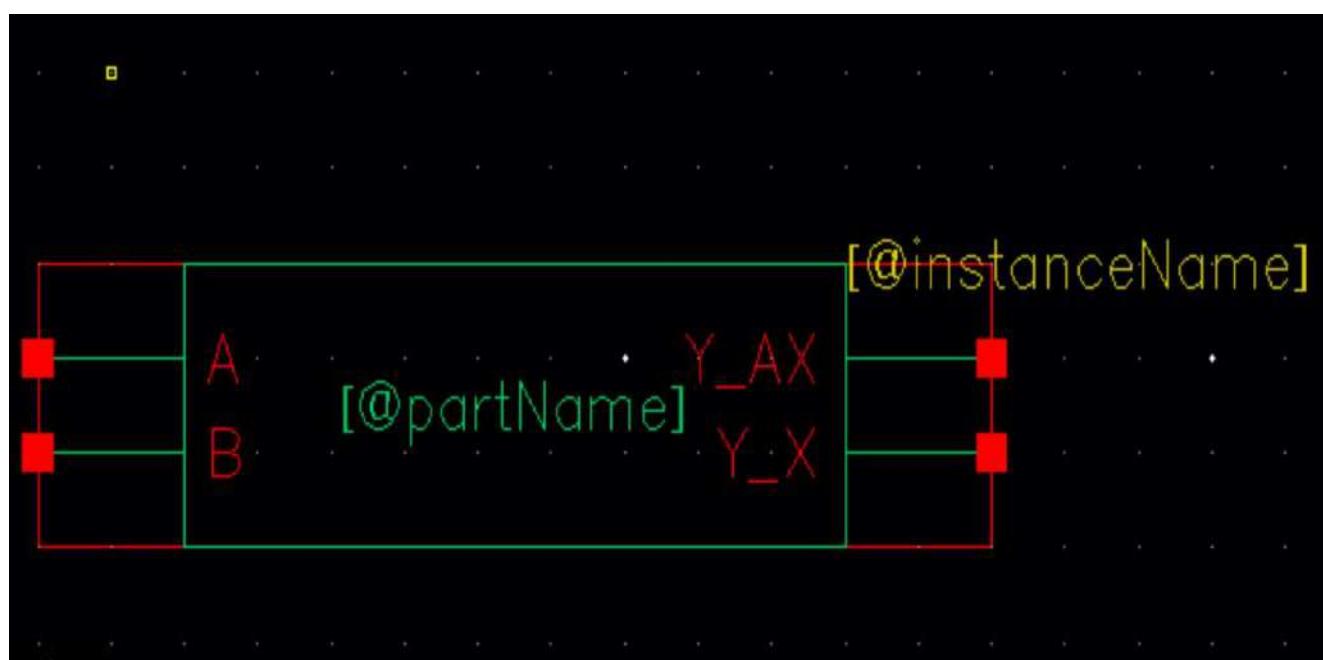
H C H + X Y O U Go to 1 / X Return On Top Explain

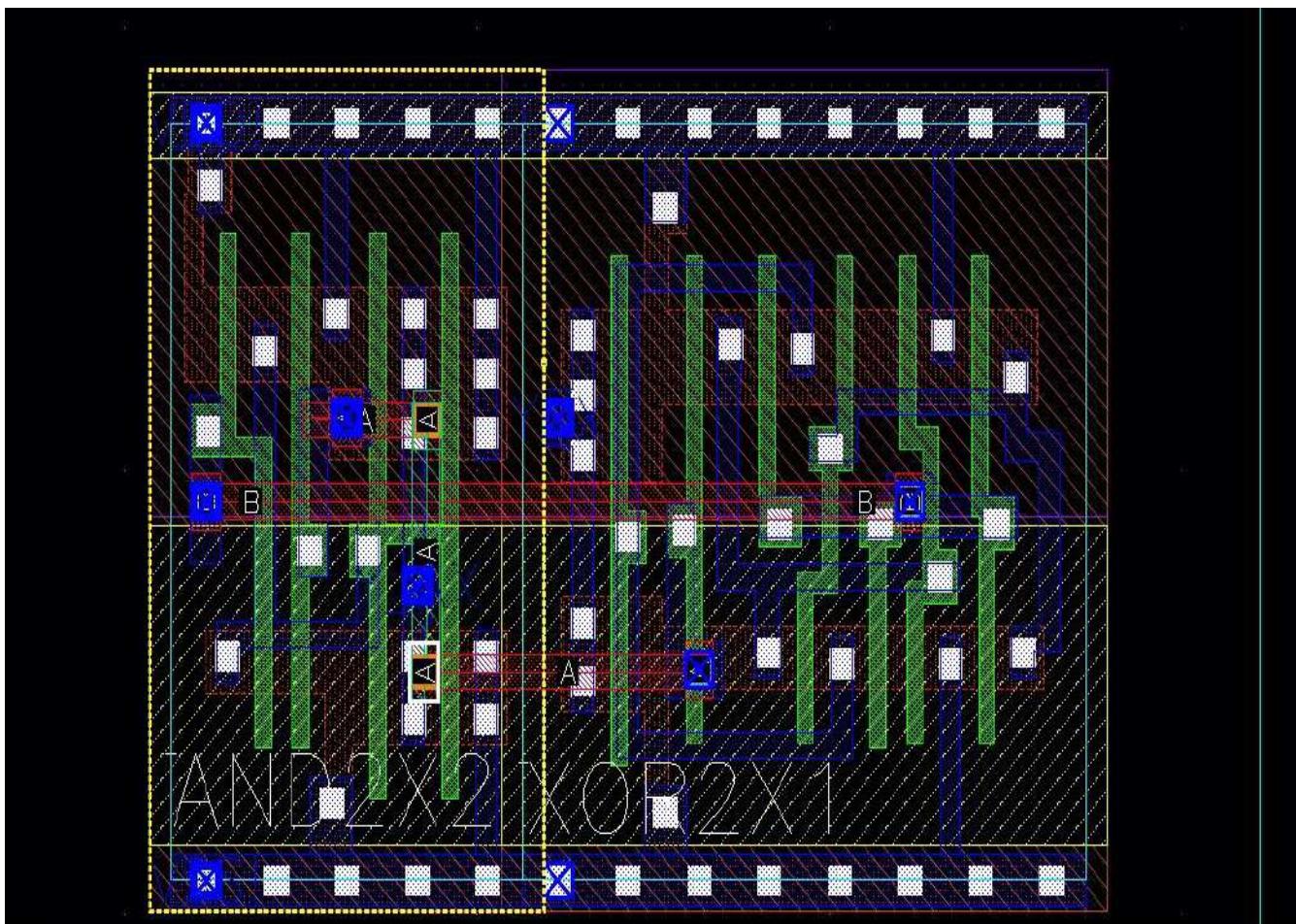
## GP-cell

### Schematic

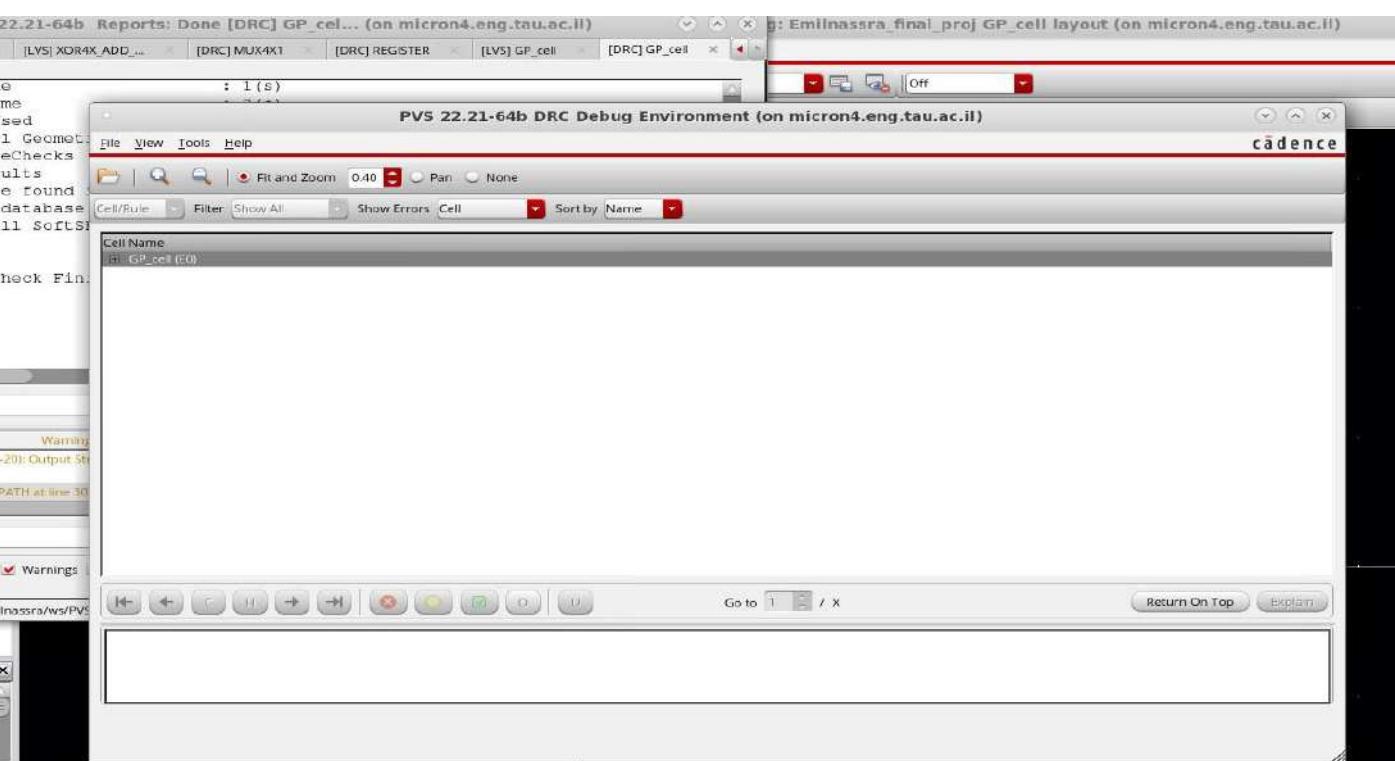


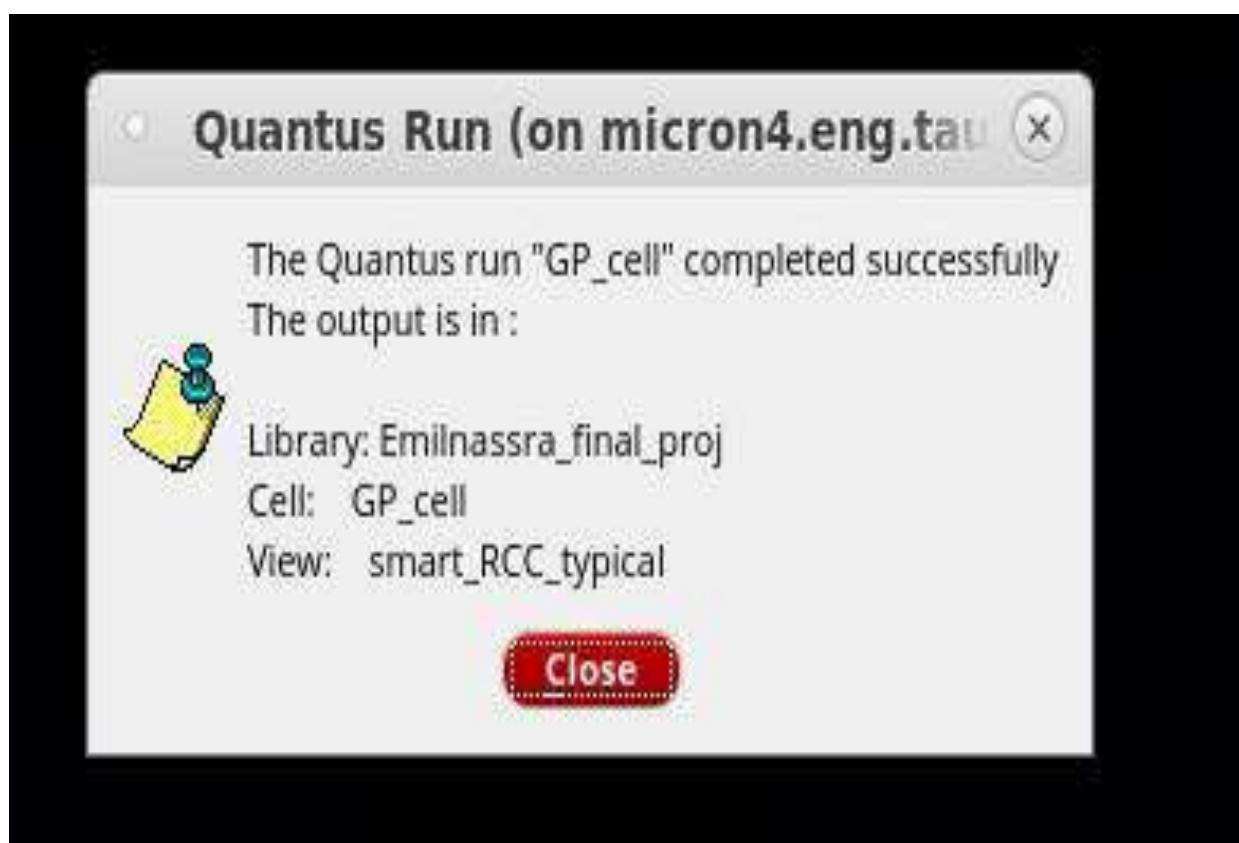
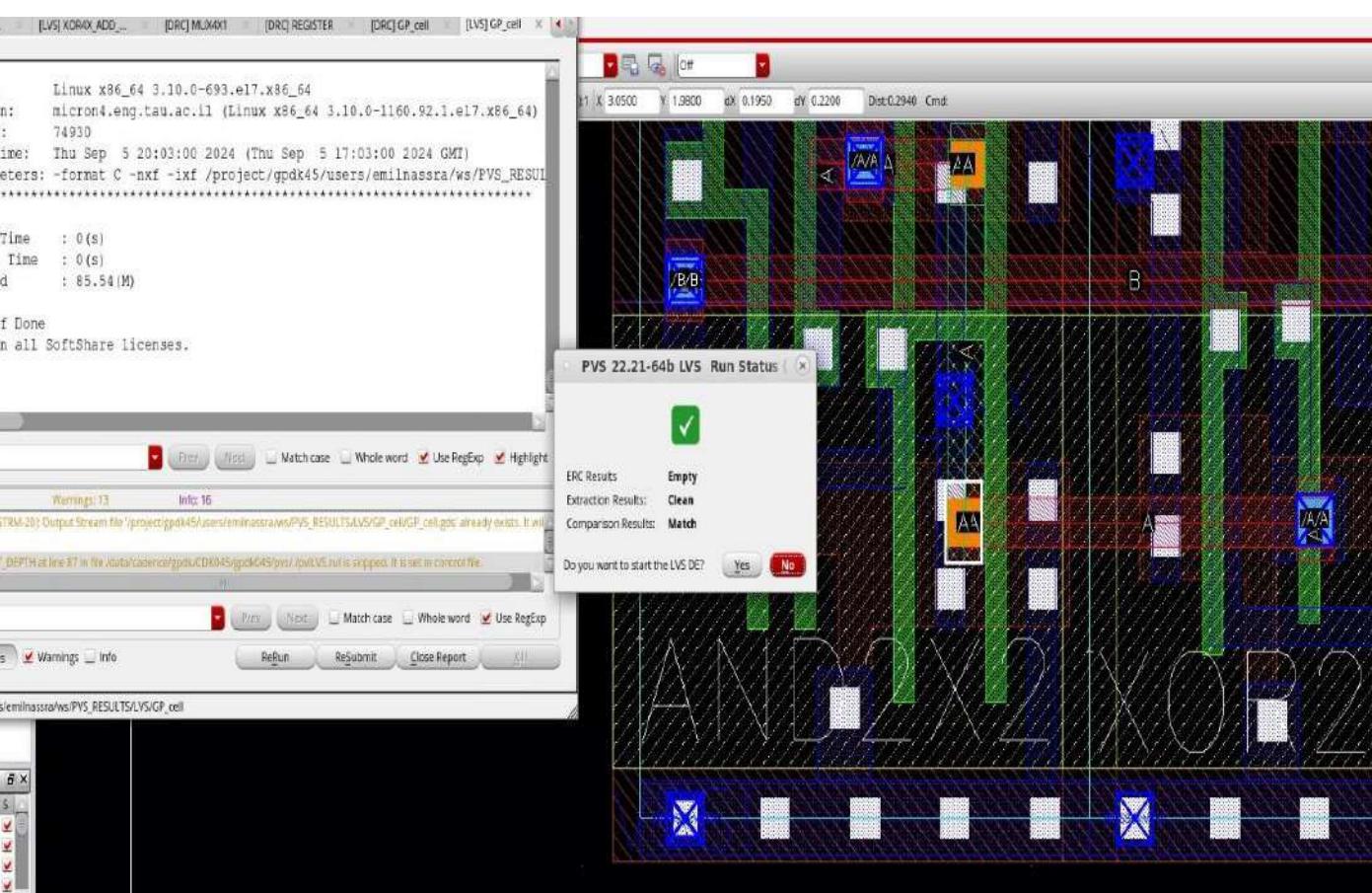
### Symbol & Layout





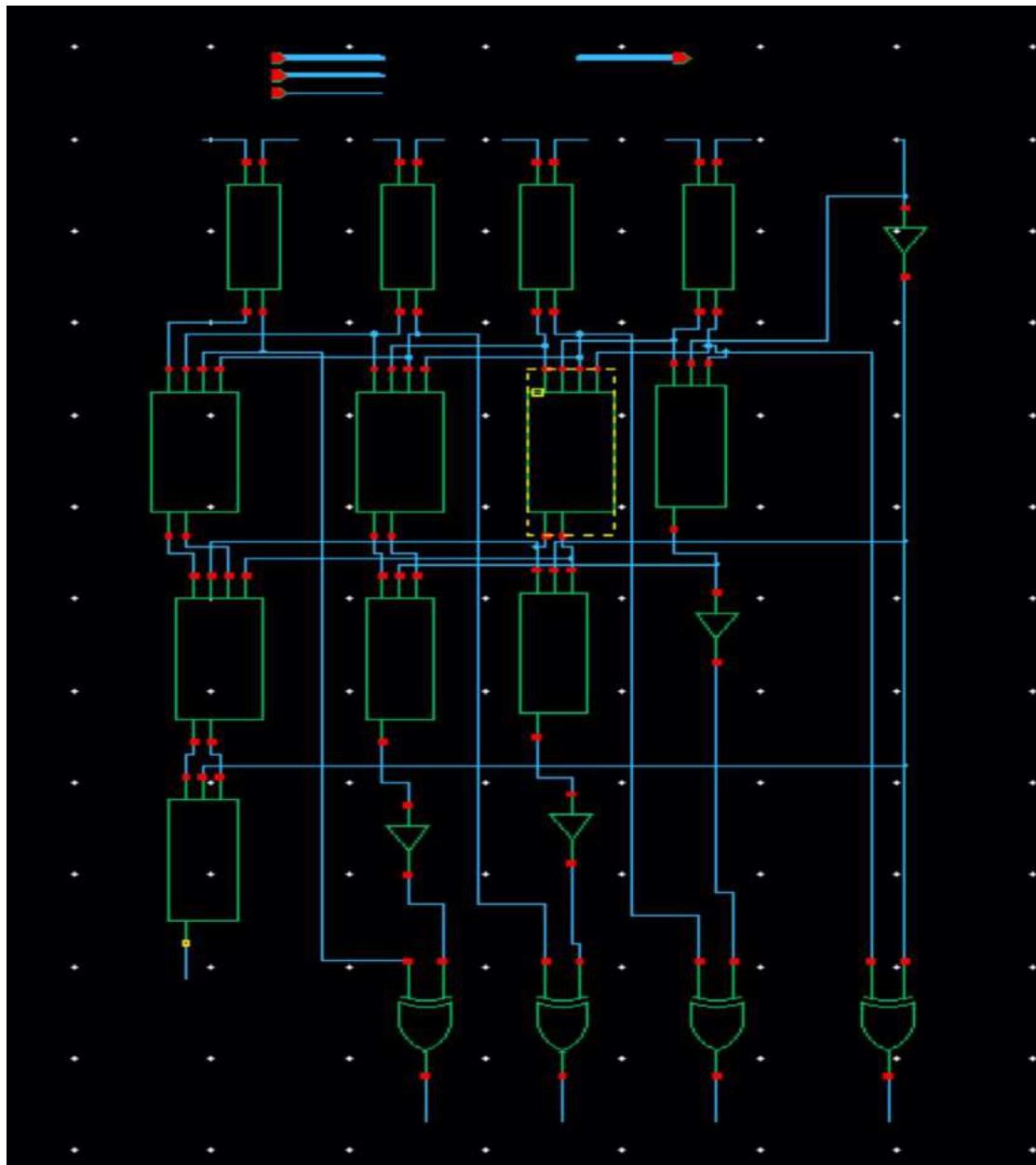
## DRC , LVS , QUANTUS





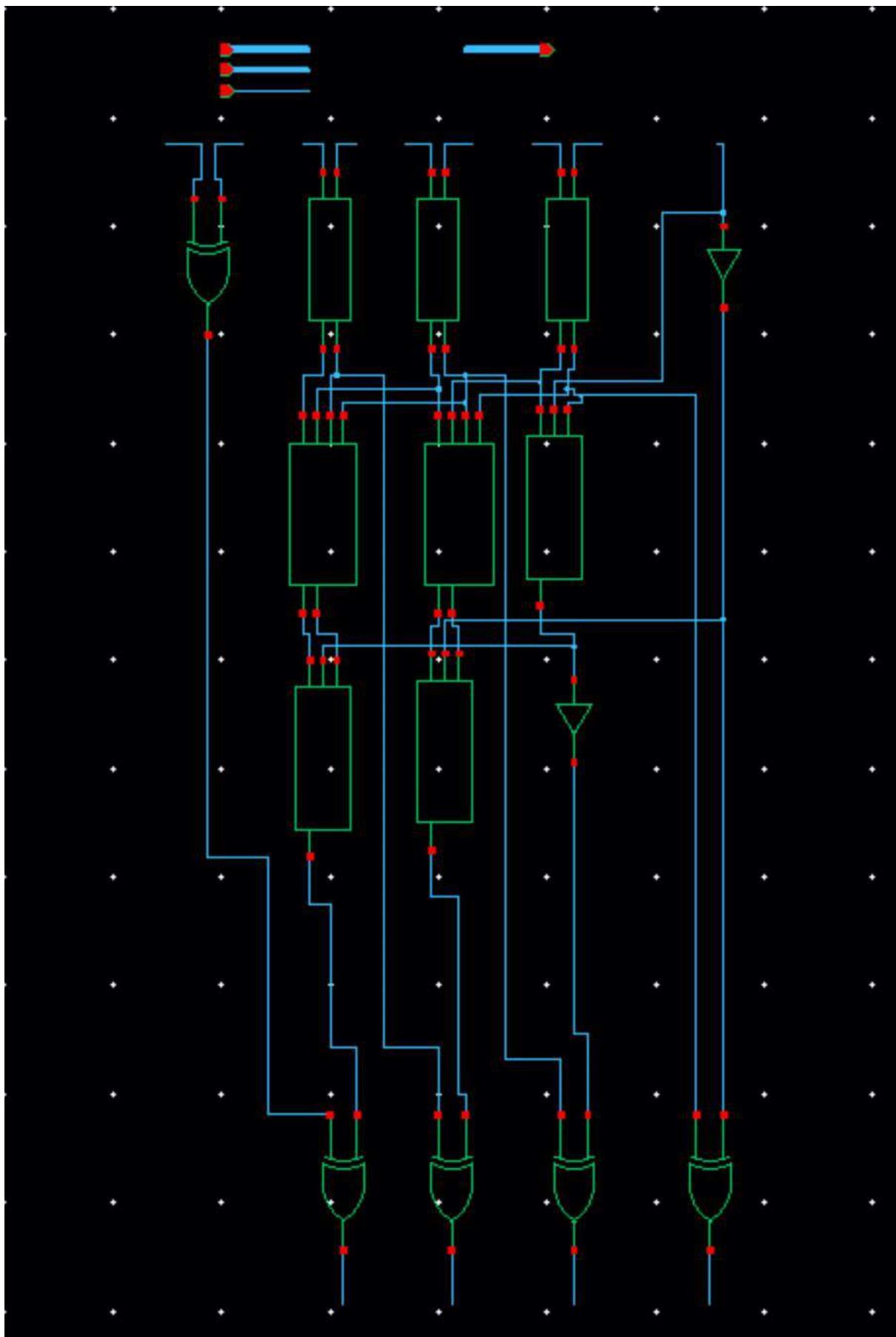
## Knowles Adder

Schematic

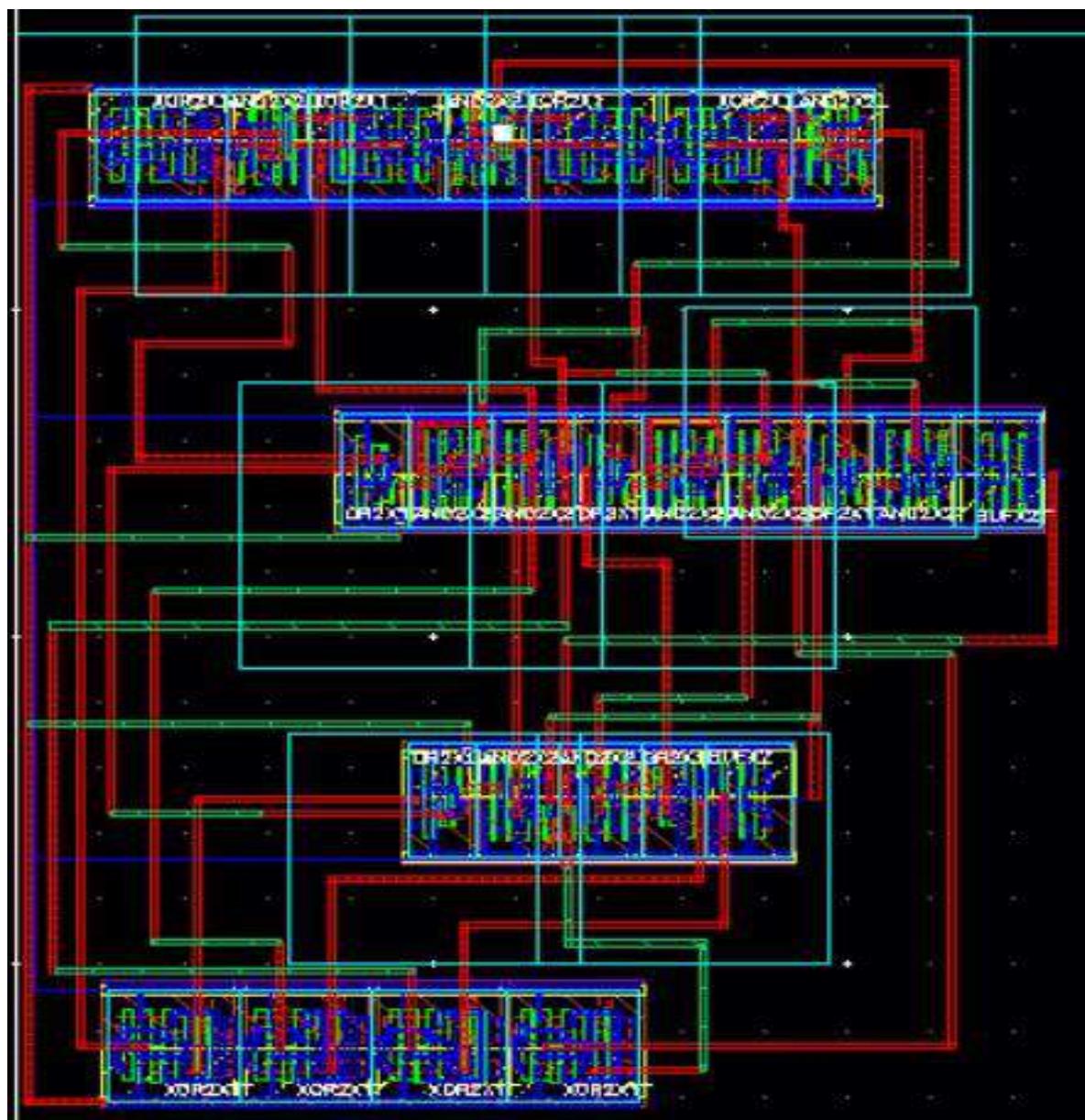
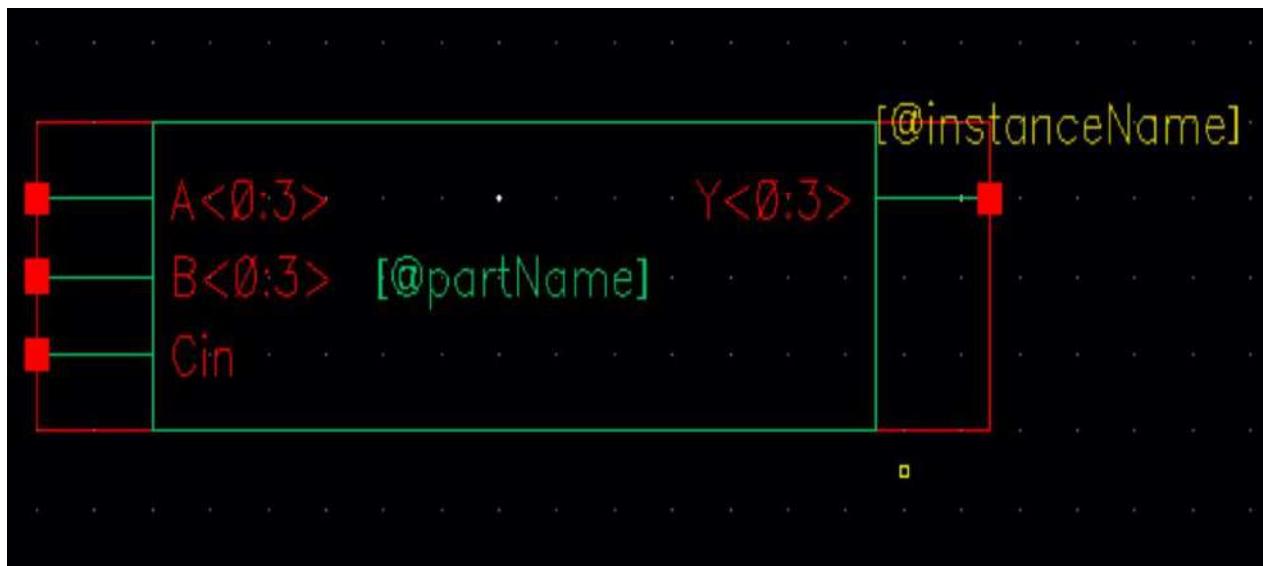


We noticed that the last path is not essential for a 4-bit adder, it would be more efficient to replace it with one  $\text{XOR}(A_3, B_3)$  gate, and we would get the same results.

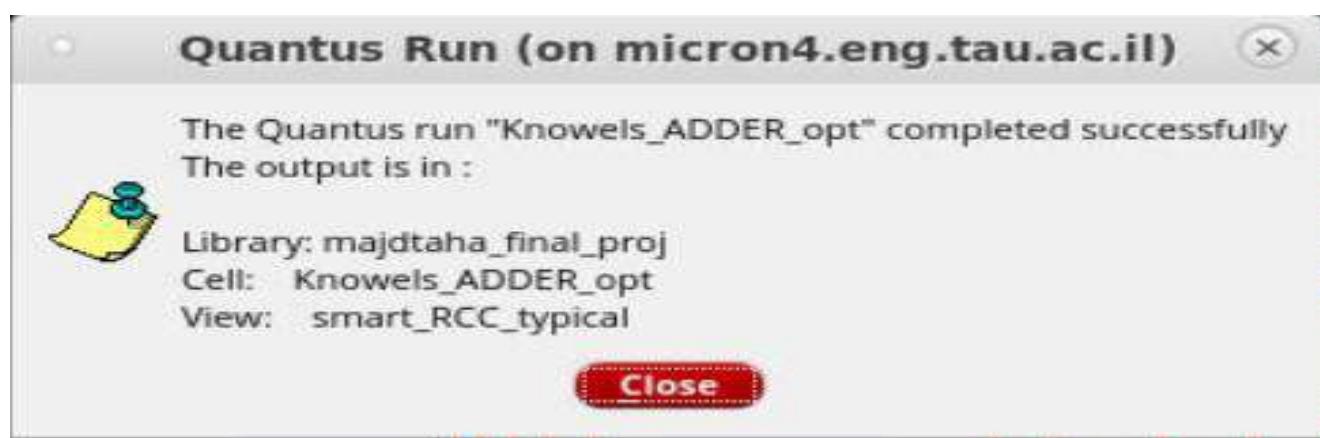
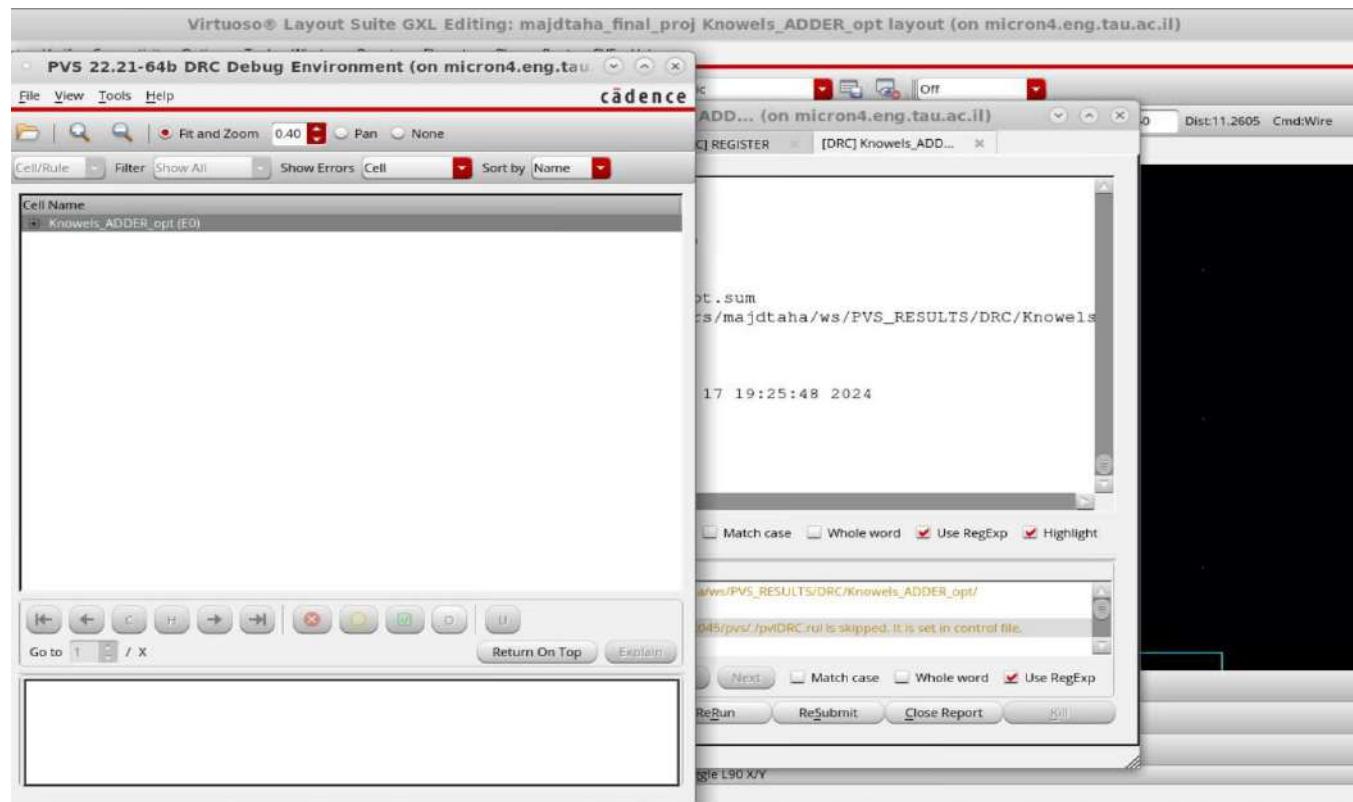
Below the schematic of the adder we implemented:



## Symbol & Layout



## DRC , LVS , QUANTUS



PVS 22.21-64b Reports: Done [LVS] Knowels\_ADD... (on micron4.eng.tau.ac.il)

REGISTER    [DRC] Black\_cell    [DRC] Knowels\_ADD...    [LVS] Knowels\_ADD...

\*\*\*\*\*

7:00 PDT 2023)

s\_2221]

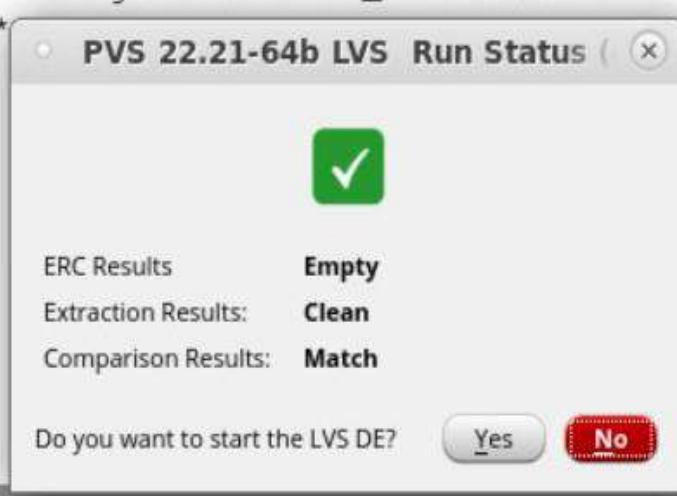
86\_64

x86\_64 3.10.0-1160.92.1.el7.x86\_64)

Sep 19 09:21:35 2024 GMT)

gpdk45/users/majdtaha/ws/PVS\_RESULTS/LVS/Knowels\_ADDER\_opt/svdb/Knowe

\*\*\*\*\*

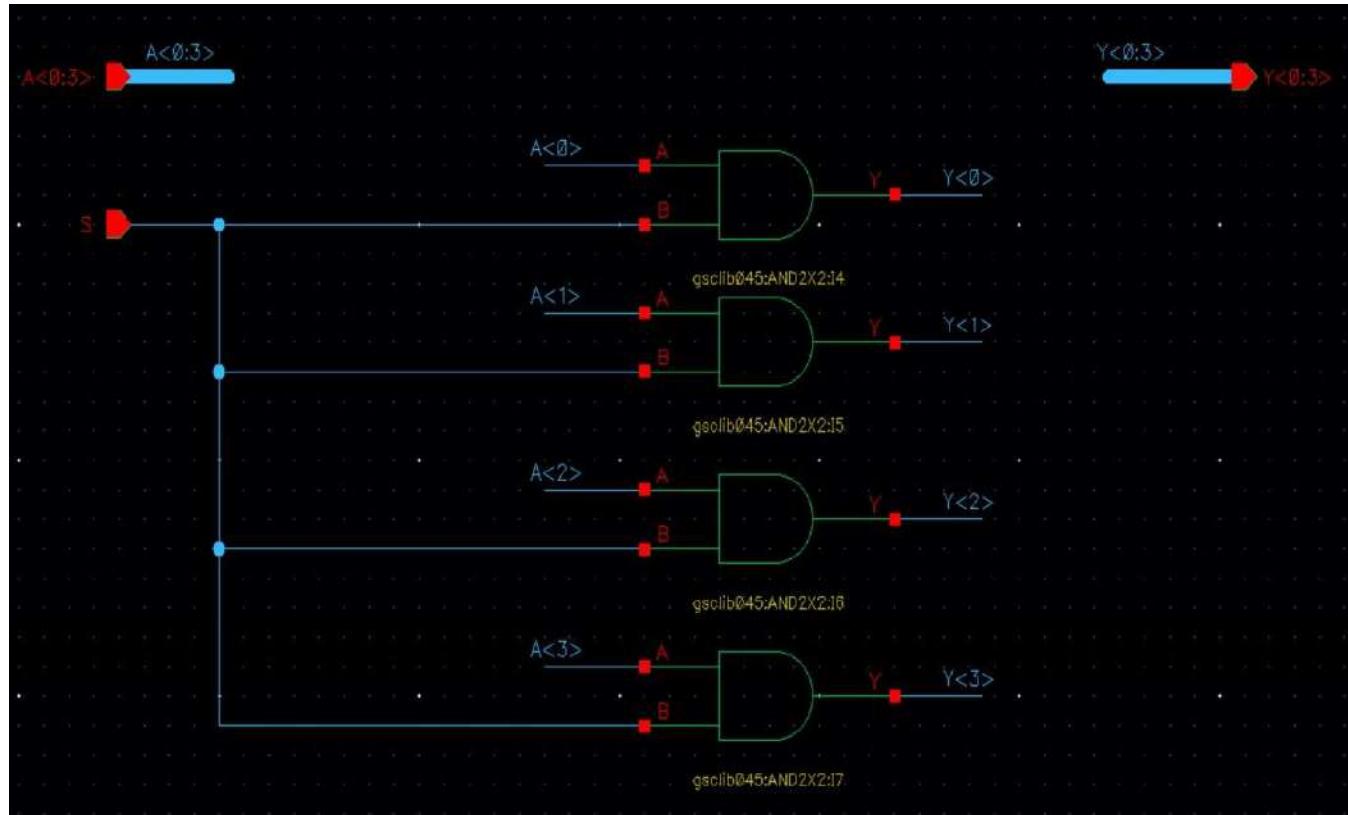


## AND4X\_ADD\_SUB\_NEG

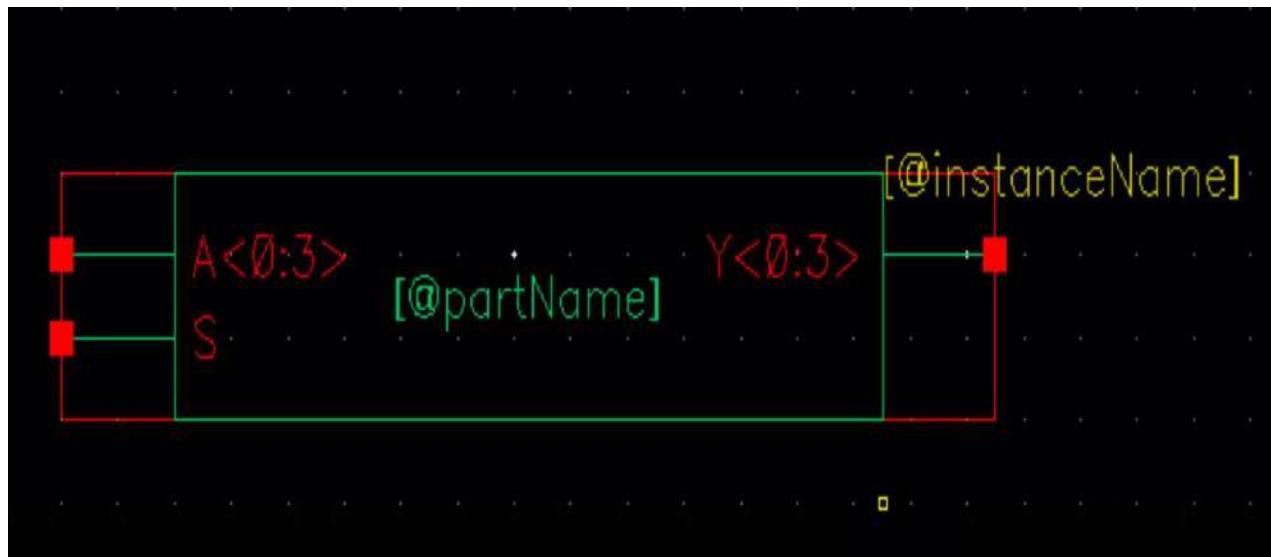
### Functionality

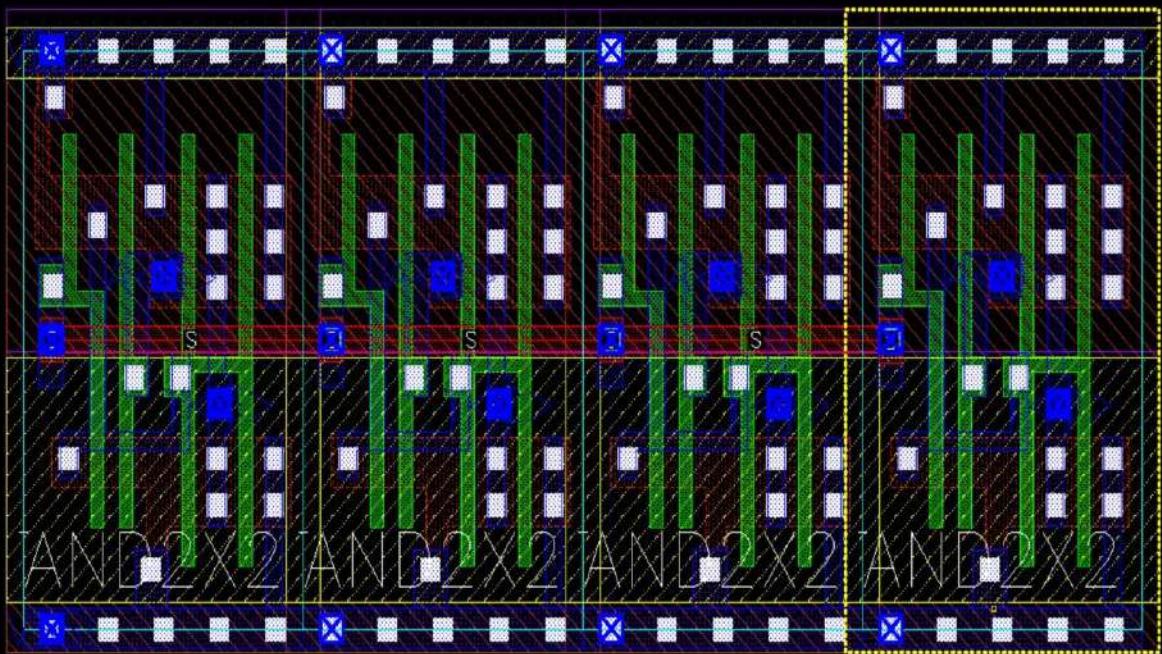
$$Y_i = \text{AND}(S, A_i)$$

### Schematic



### Symbol & Layout





DRC , LVS , QUANTUS

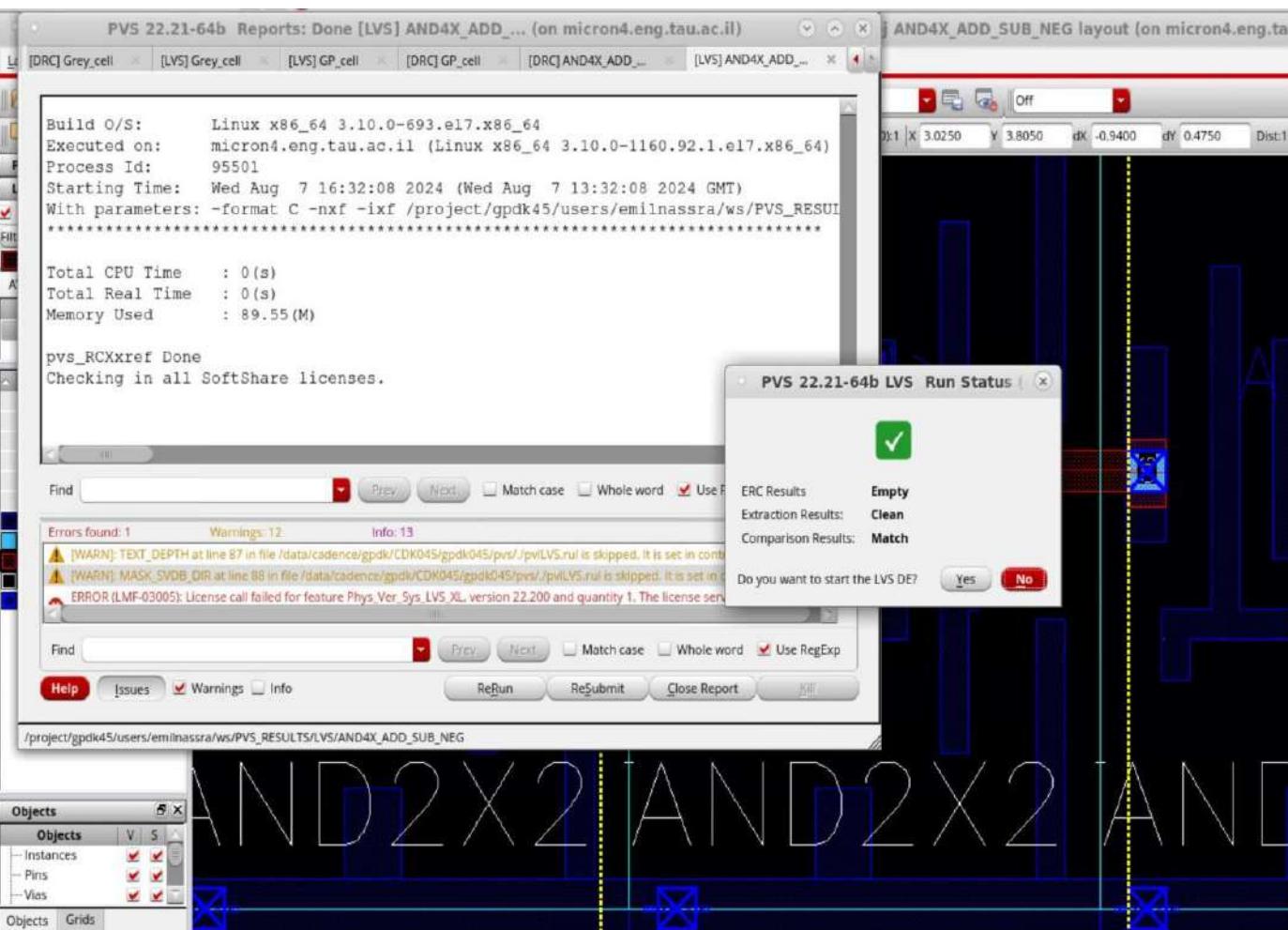
### Quantus Run (on micron4.eng.tau.ac.il)

The Quantus run "AND4X\_ADD\_SUB\_NEG" completed successfully  
The output is in :



Library: Emilnassra\_final\_proj  
Cell: AND4X\_ADD\_SUB\_NEG  
View: smart\_RCC\_typical

[Close](#)

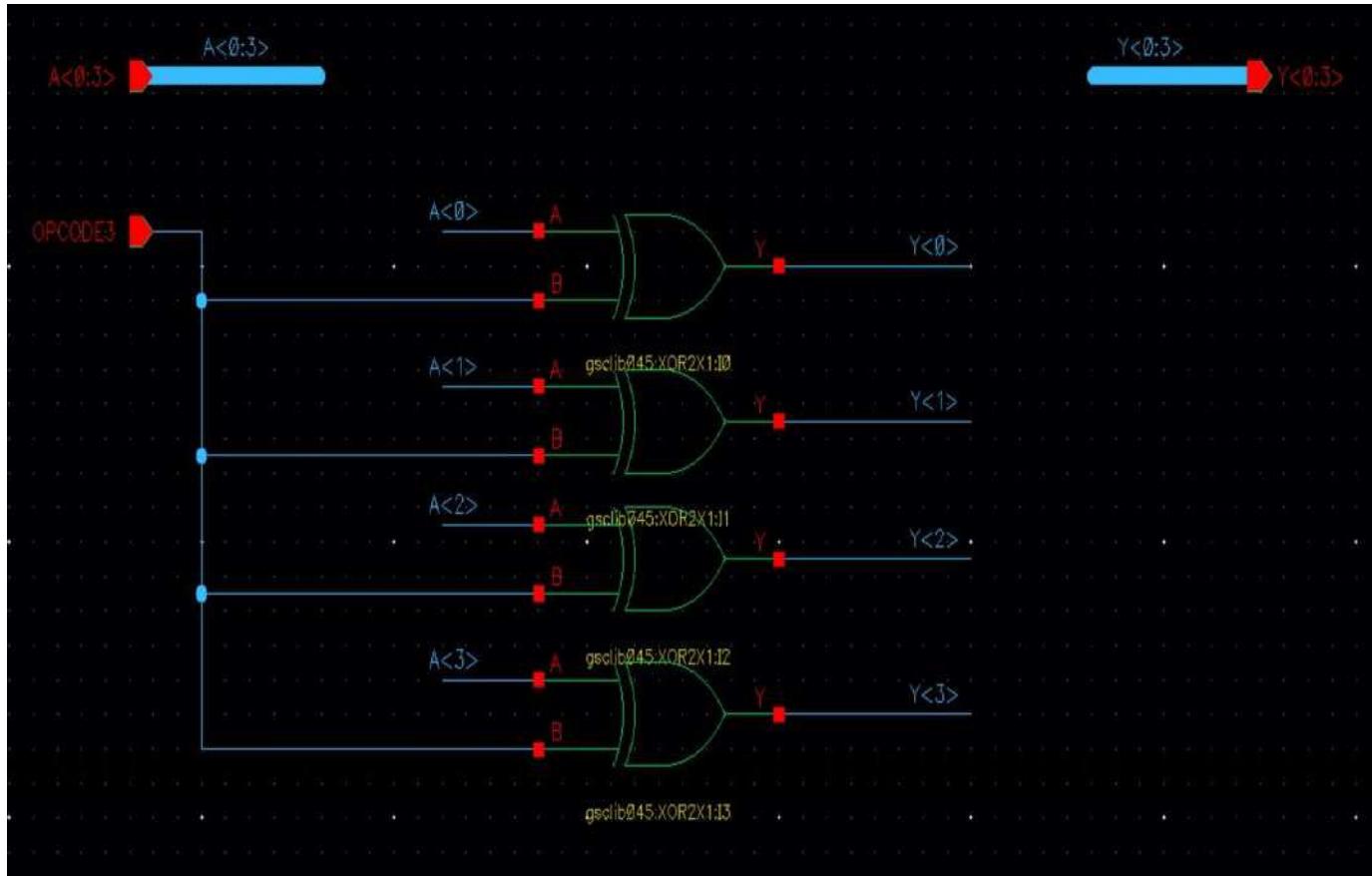


## XOR4X\_ADD\_SUB\_NEG

### Functionality

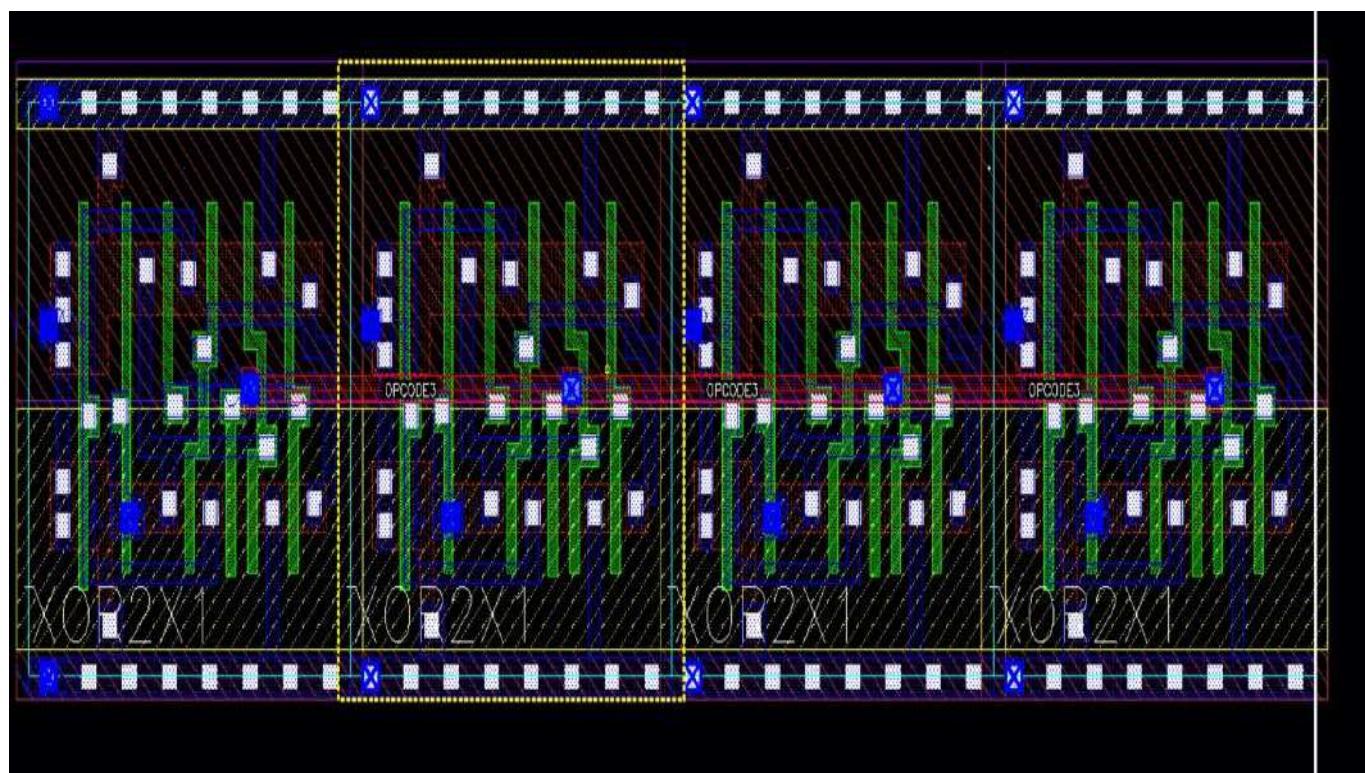
$$Y_i = \text{XOR}(\text{Opcode3}, A_i)$$

### Schematic



### Symbol & Layout





DRC , LVS , QUANTUS

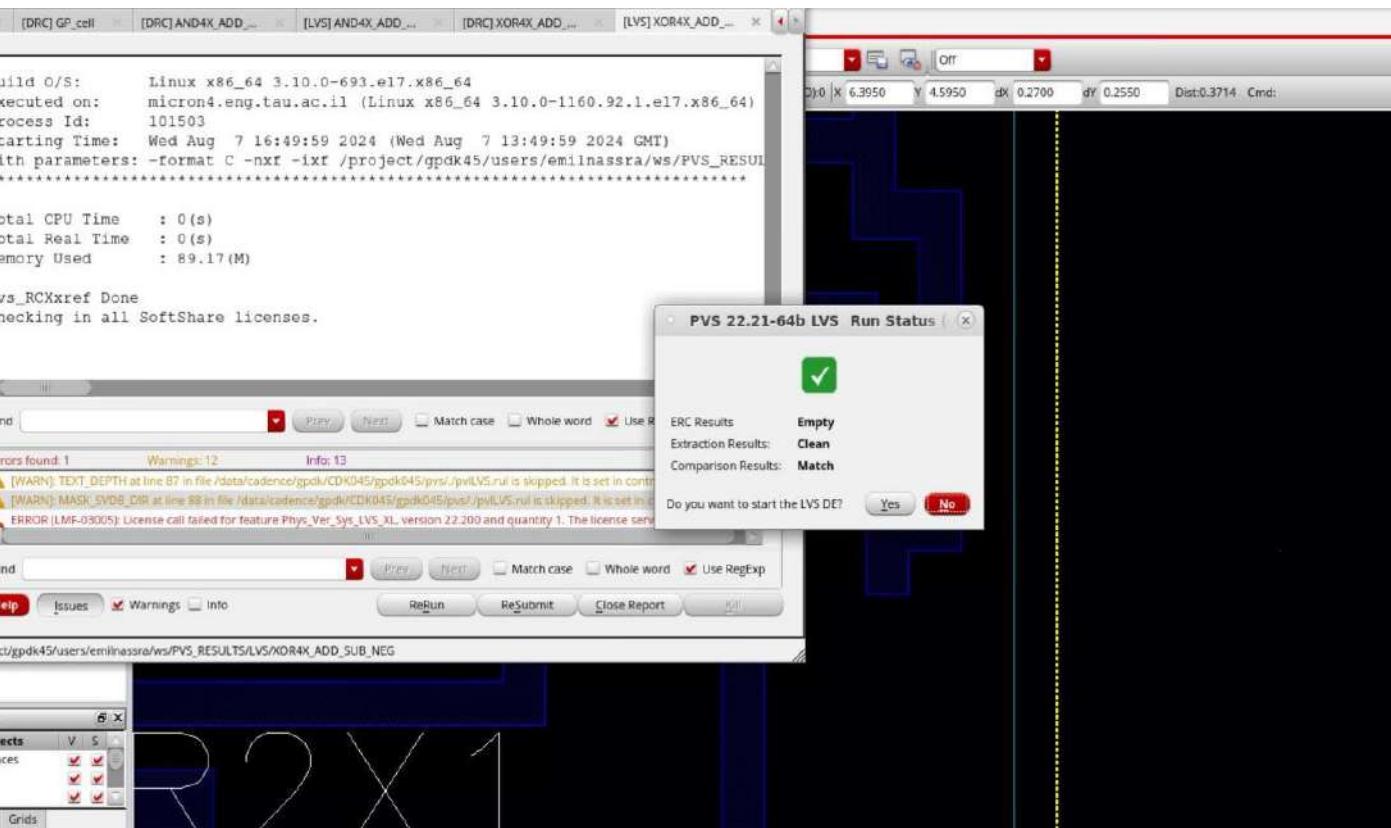
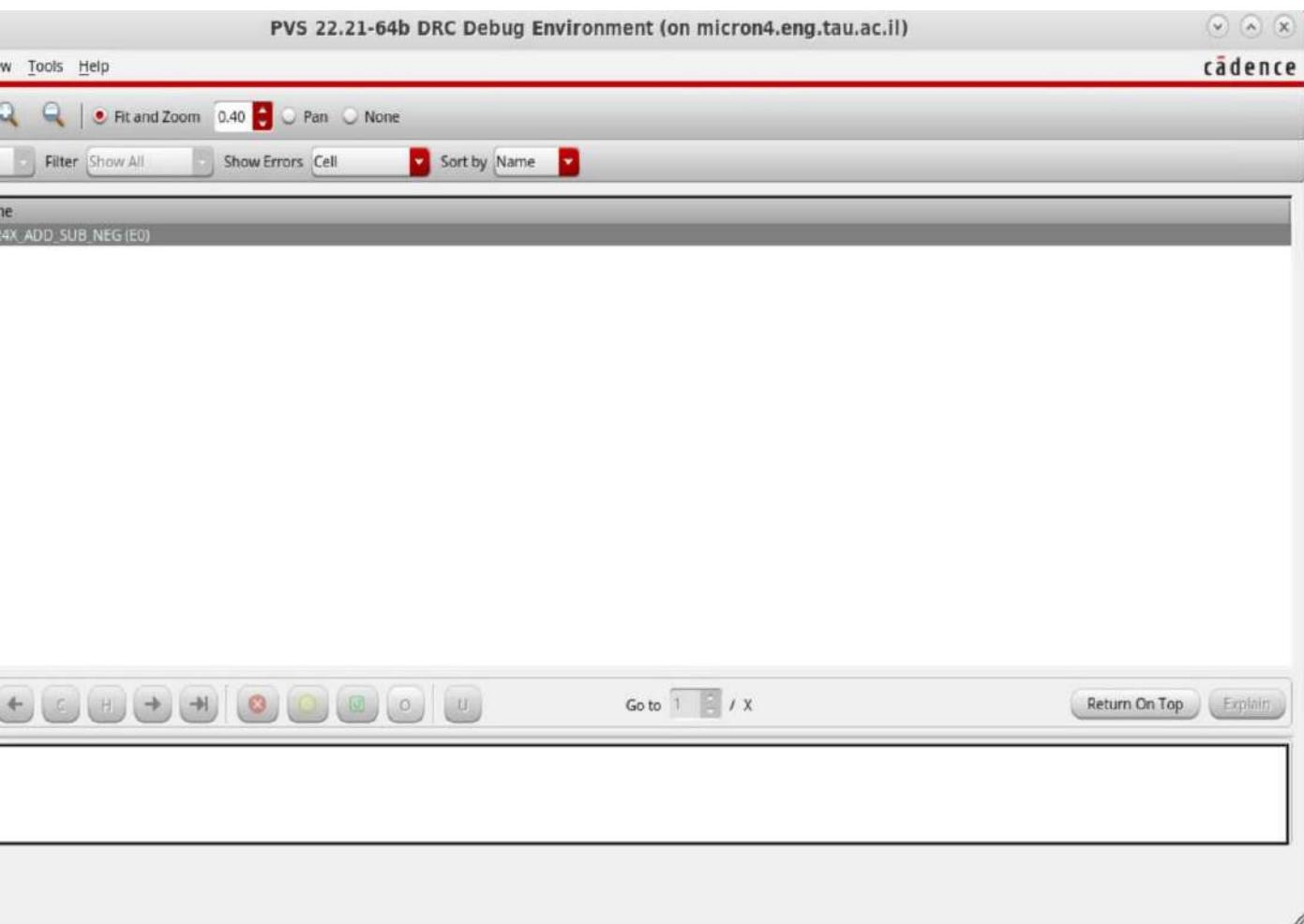
## Quantus Run (on micron4.eng.tau.ac.il) X

The Quantus run "XOR4X\_ADD\_SUB\_NEG" completed successfully  
The output is in :



Library: Emilnassra\_final\_proj  
Cell: XOR4X\_ADD\_SUB\_NEG  
View: smart\_RCC\_typical

**Close**



## ADD/SUB/NEG Block

### Functionality

In this circuit we used the Knowles adder that we built to calculate different functions according to the opcode in the table below:

	operation	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
ADD/ SUB/ NEG	ADD	X	0	0	0	
	SUB	X	1	0	0	
	NEG	X	1	1	1	

#### ADD function:

- The output of the AND(bit1 , bit2) is zero, thus the XORX4 gate; which executes bitwise-XOR of zero and A, passes the value of A as it is; unchanged.
- Bit3 of the opcode is zero, and again the XOR4 gate passes the value of B unchanged, also the ANDX4 gate doesn't change B, because it execute bitwise-AND with '1'.
- Both inputs passes to the Knowles adder with Cin zero and the ADD function is done.

#### SUB function:

- Same as above A passes unchanged.
- Bit3 of the Opcode is '1', so the XORX4 inverts the bits of B, the ANDX4 changes nothing.
- A and the inverted bits of B passes to the Knowles adder and with the Cin='1' and the SUB function is done.

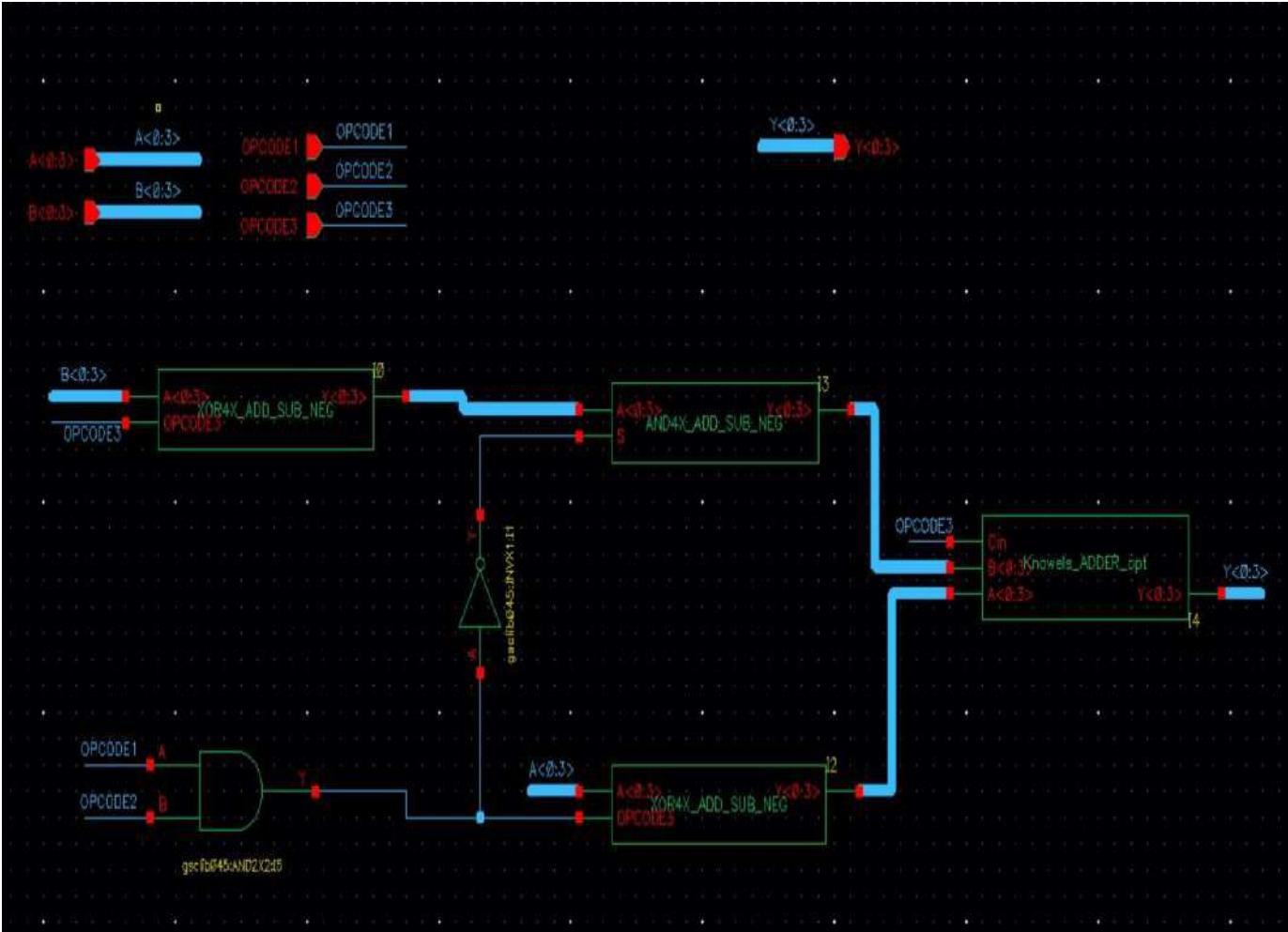
$$A - B = A + [ \text{inv}(B) + 1 ]$$

#### NEG function:

- The output of the AND(bit1 , bit2) is '1', so the XORX4 gate inverts the bits of A.
- The ANDX4 receives the inverted bits of B and Not[ AND(bit1 , bit2) ] which is zero, so the result is zero.
- Only A passes to the Knowles adder, with zero instead of B and Cin='1', and the NEG function is done.

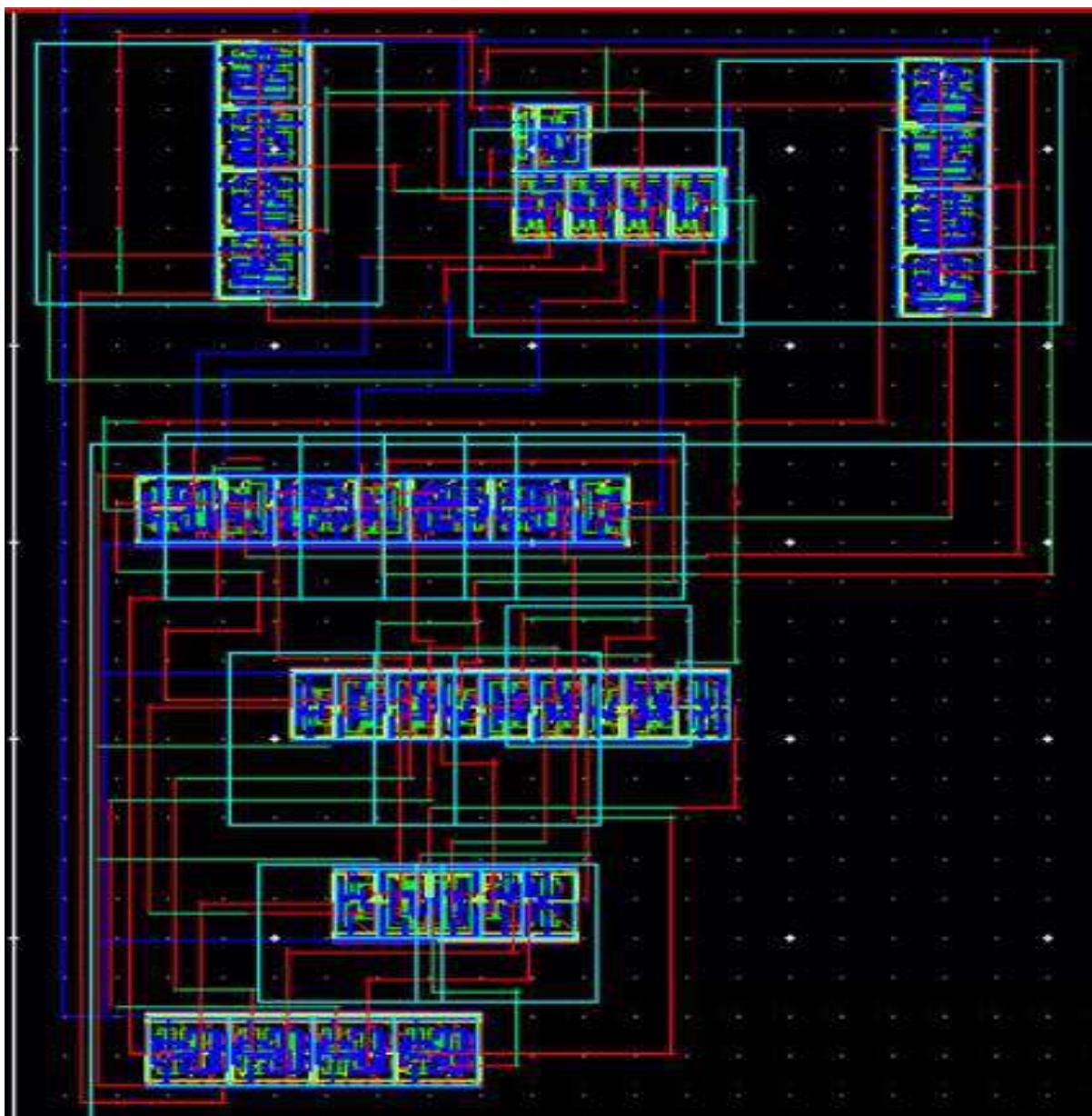
$$- A = \text{inv}(A) + 1$$

## Schematic

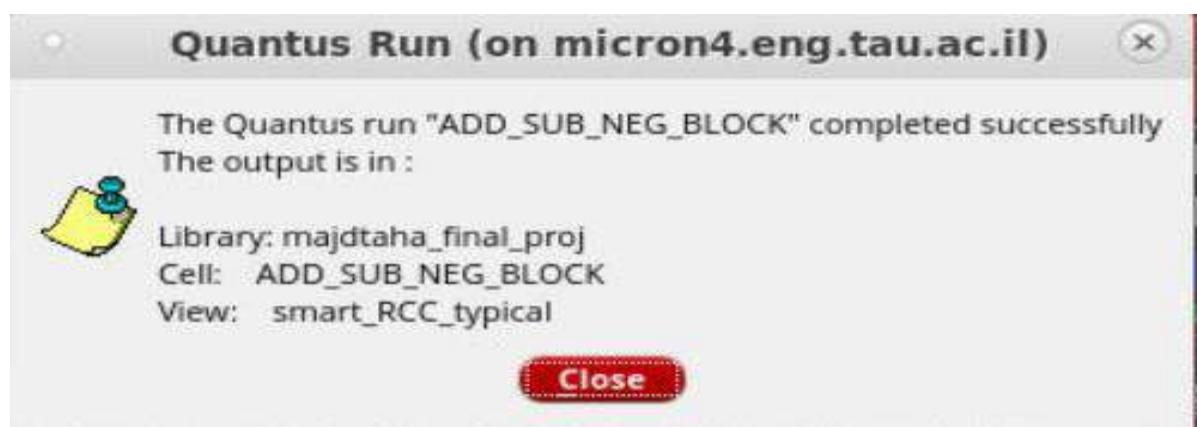


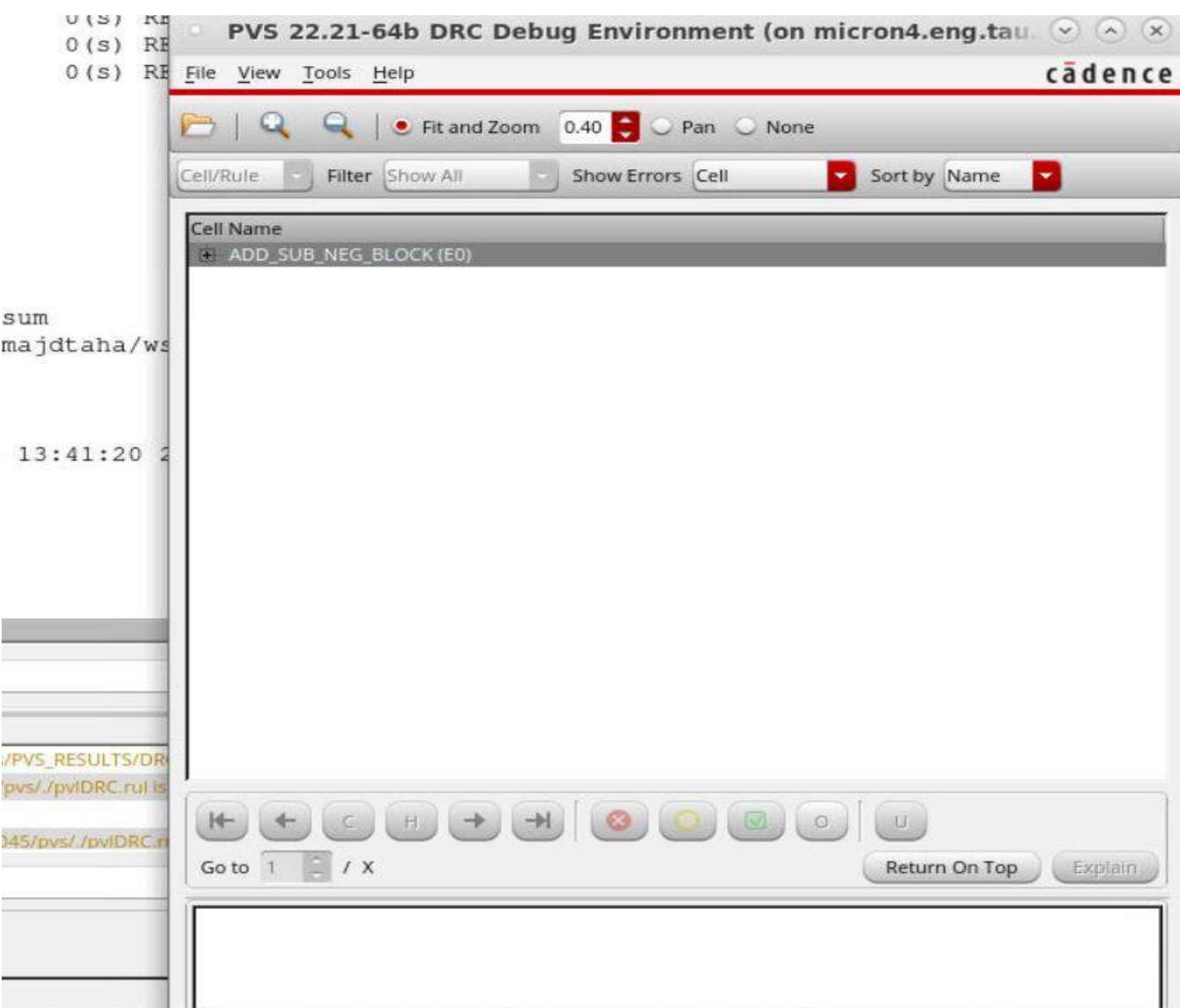
## Symbol & Layout





DRC , LVS , QUANTUS





RESULTS/LVS/ADD\_SUB\_NEG\_BLOCK/svdb/AI

PVS 22.21-64b LVS Run Status ( X )



ERC Results      **Empty**

Extraction Results:      **Clean**

Comparison Results:      **Match**

Do you want to start the LVS DE?

Yes

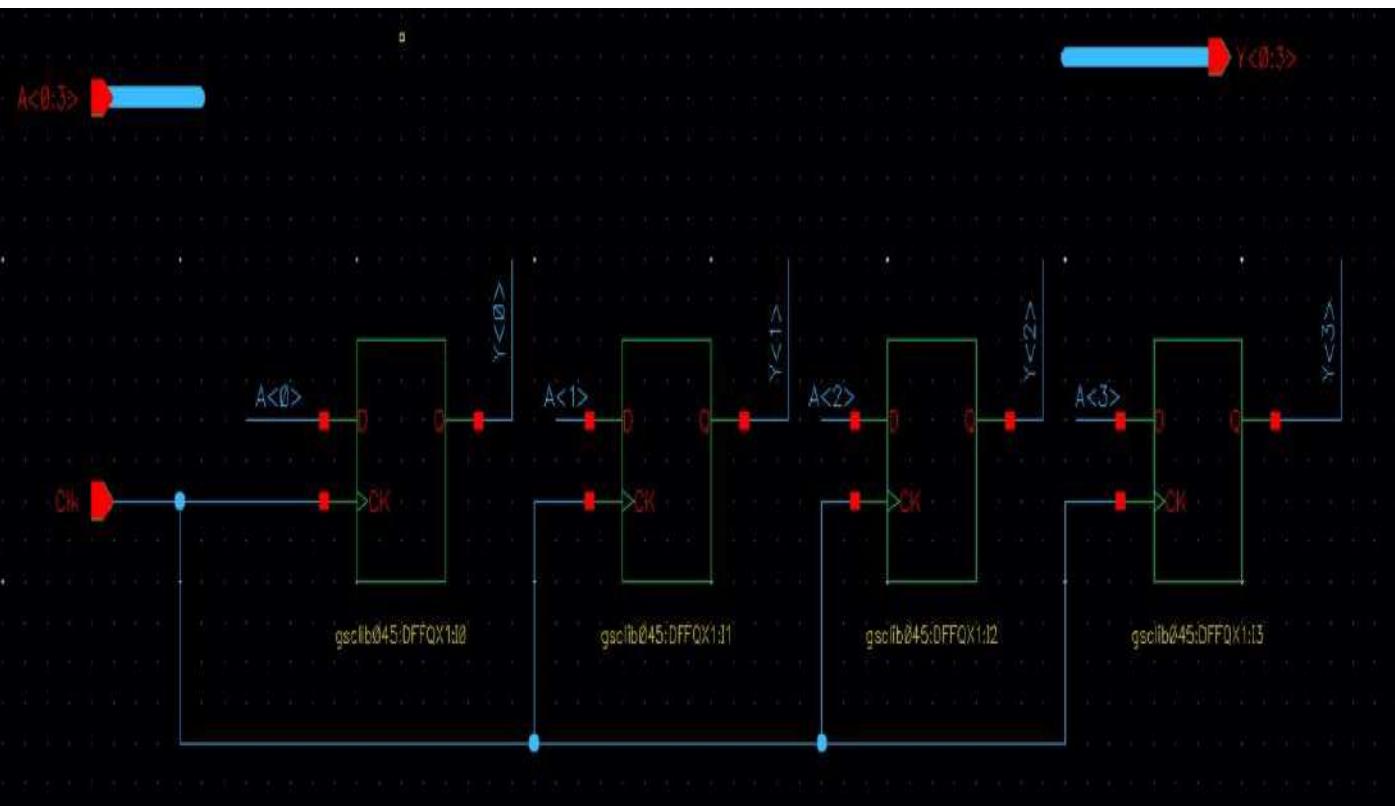
**No**

## Register

### Functionality

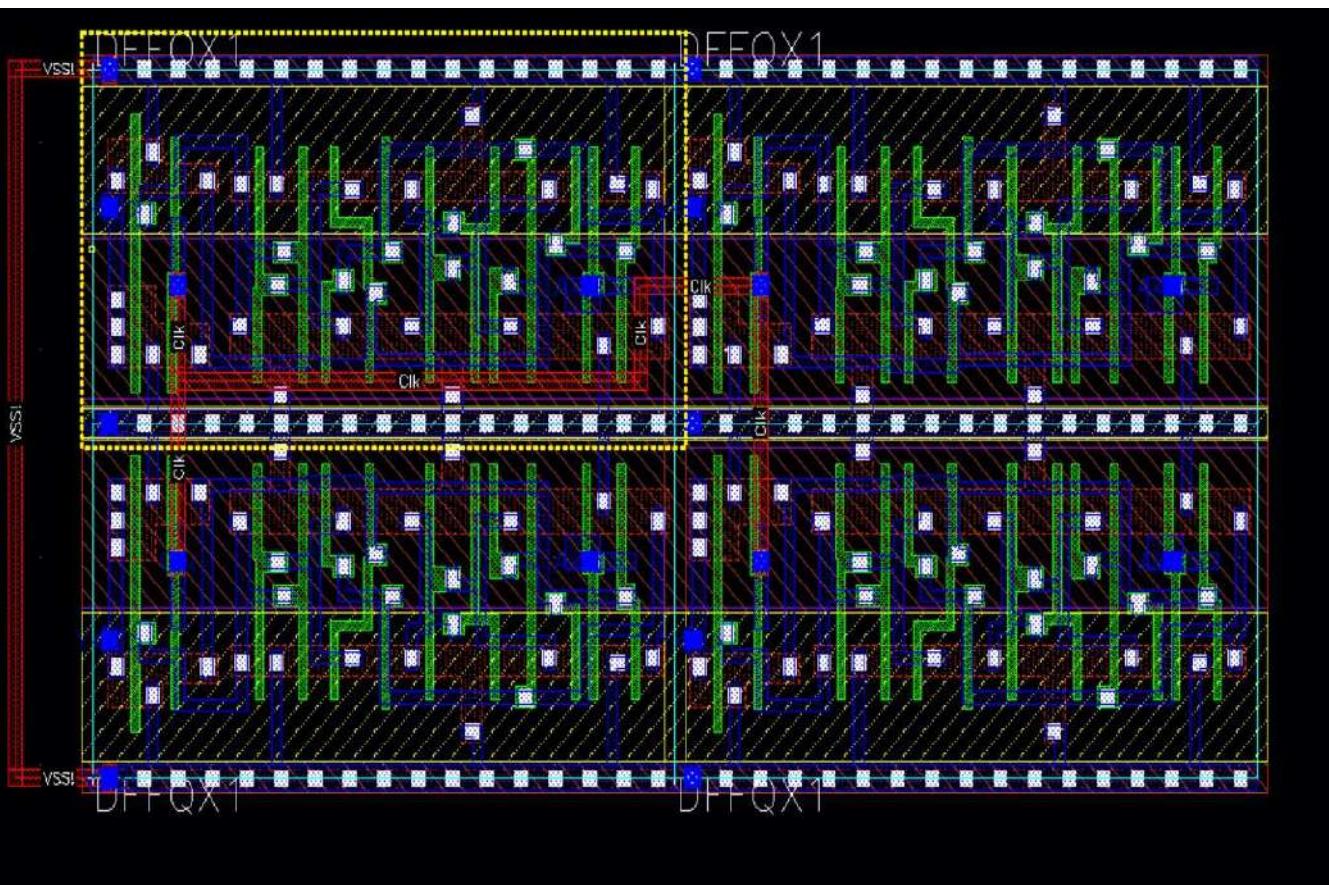
This circuit receives a 4-bit input, when the clock's signal rises the input is stored in the register.

### Schematic



### Symbol & Layout





DRC , LVS , QUANTUS

22.21-64b Reports: Done [DRC] REGISTER... (on micron4.eng.tau.ac.il)

LVS finished X: 4950 Y: 30500 DRC: 1.2202 Cmt:

Time : 2 (s)  
 Used : 19 (M)  
 Total Geometry : 144 (522)  
 RuleChecks : 562  
 Results : 0 (0)  
 0 found in file REGISTER.sum  
 Project database is /project/gdk45/users/emadmazzawi/ws/PVS\_RESULTS/  
 All SoftShare licenses.

Check Finished Normally. Mon Sep 16 20:32:33 2024

Warnings: 0 Info: 573

M400 Output Screen file /project/gdk45/users/emadmazzawi/ws/PVS\_RESULTS/DRC/REGISTER/REGISTER.gch already generated.  
 Path tracing is off. The command line option --PathTracing is disabled. It is set in your file  
 to generate LAYOUT\_PRIMARY "REGISTER".

PVS 22.21-64b DRC Debug Environment (on micron4.eng.tau.ac.il)

File View Tools Help

CellName: Cell Name:

CellList: Filter: Show All Show Errors Cell Sort By: Name

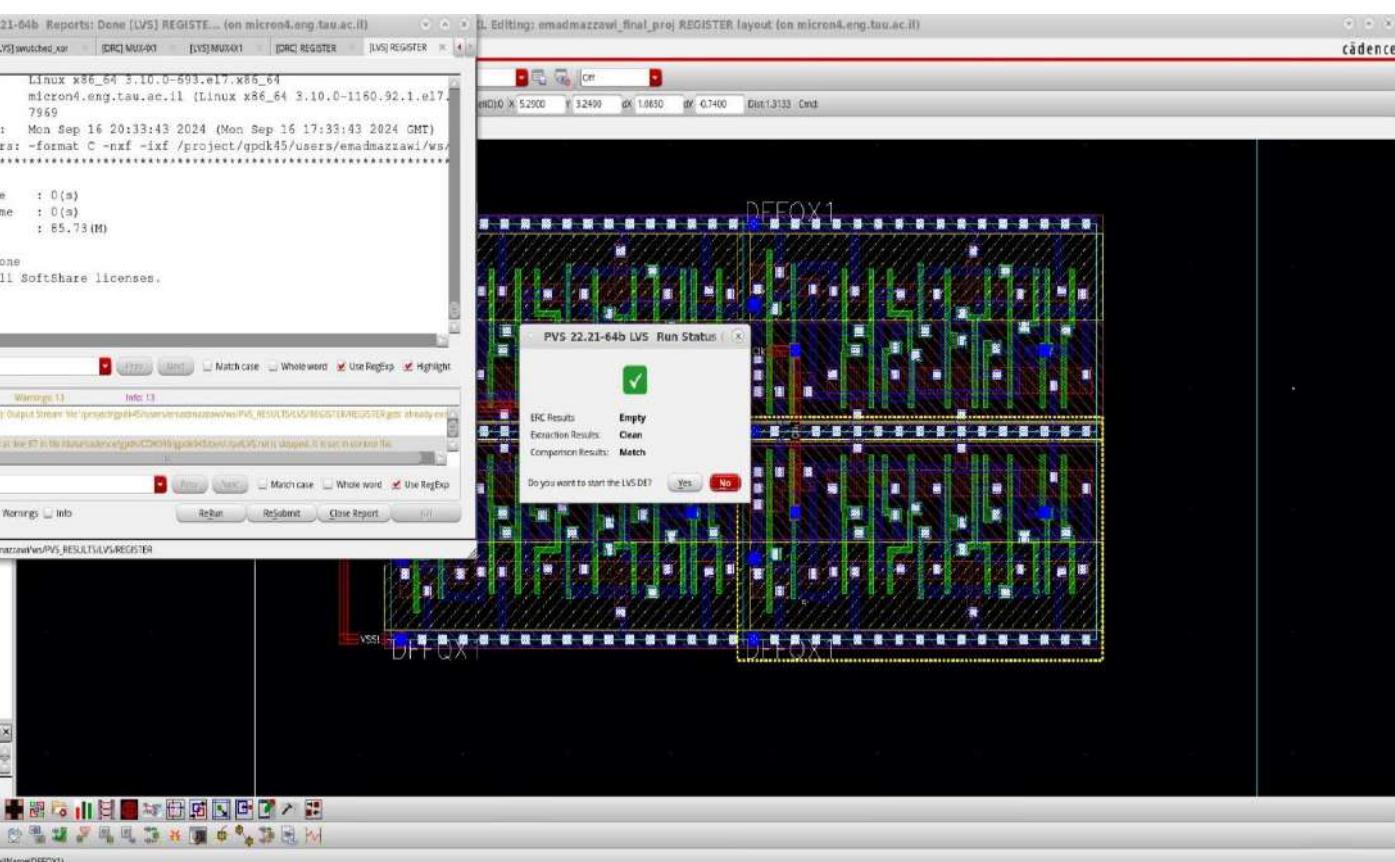
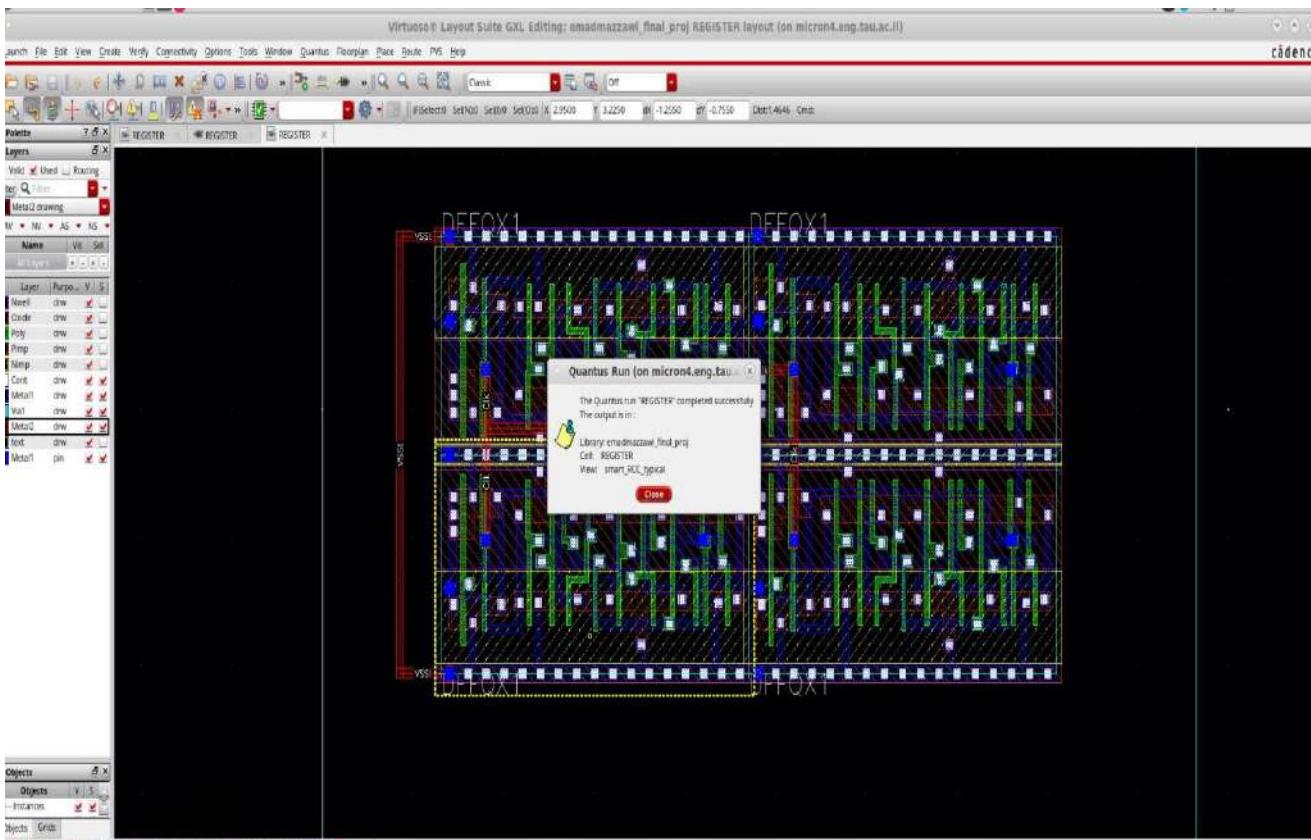
Go to: / X Return On Top Import

cadence

CellName(DFFQX1)

M:/pvsOpen(PVS)onTerm: drc /timeout 30

R: 504 MousePopUp

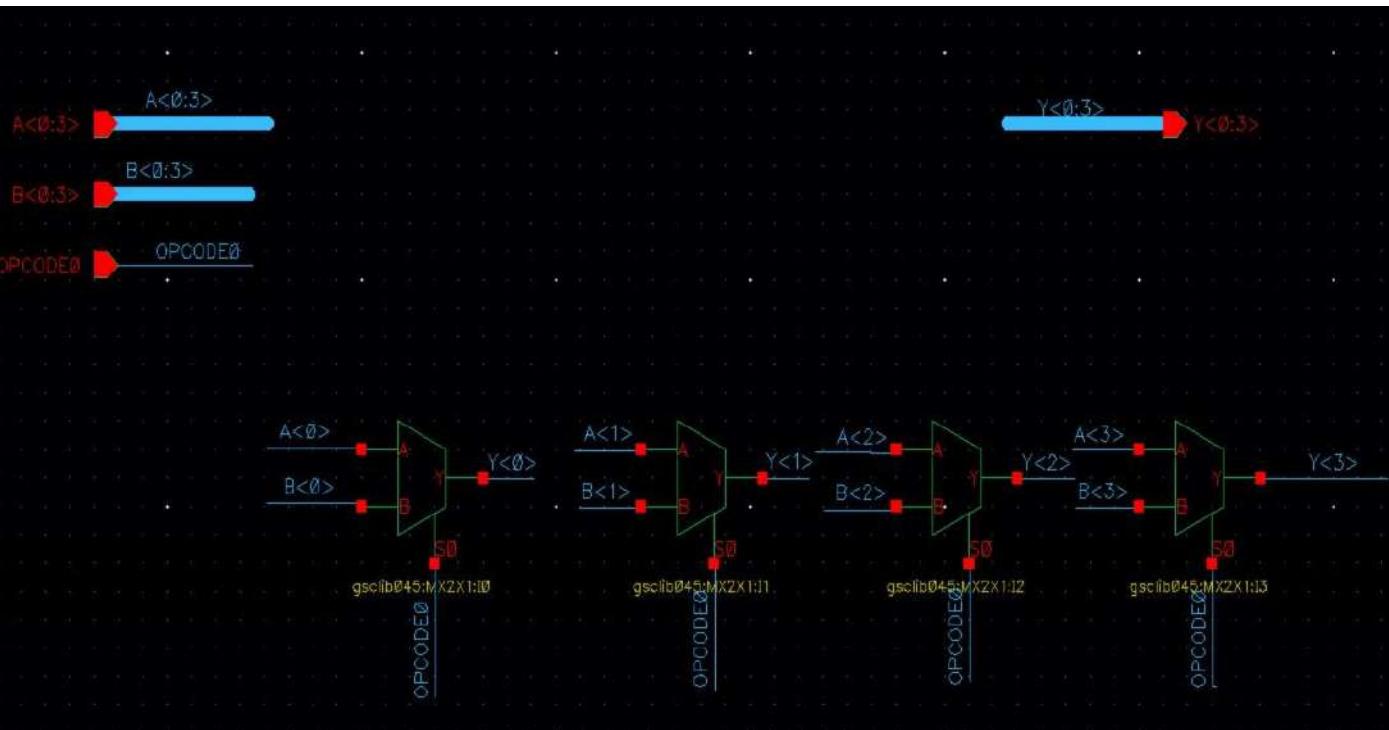


## MUX 2X1

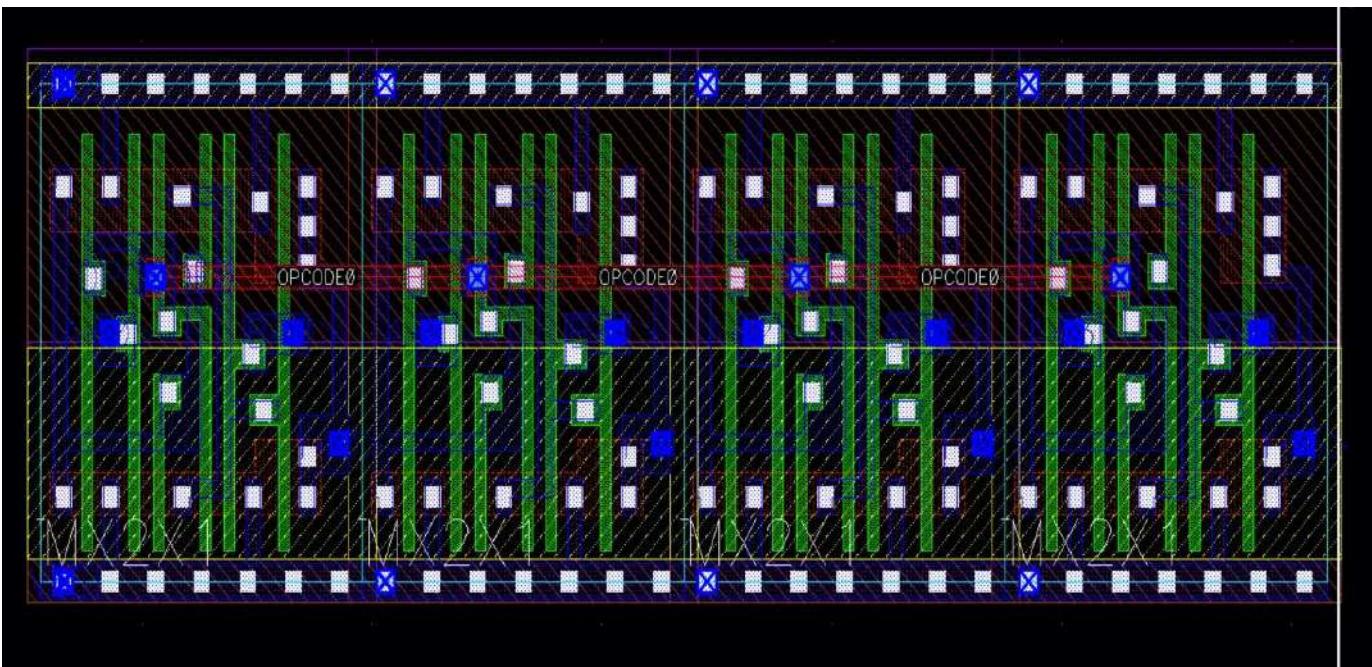
### Functionality

The multiplexer receives 2 data inputs - 4 bit each, and according to the select lines, gives a single output.

### Schematic

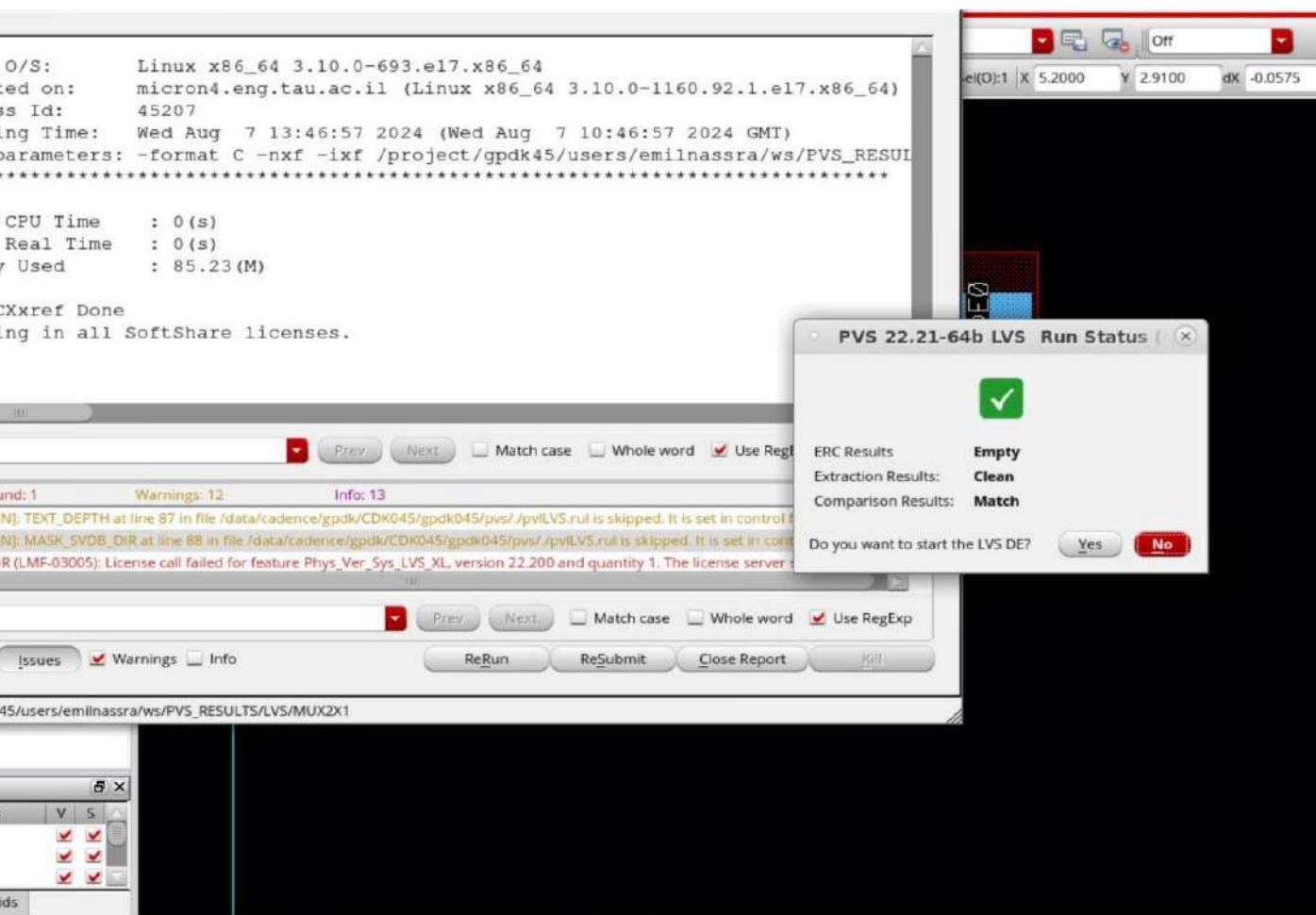


### Symbol & Layout





DRC , LVS , QUANTUS



PVS 22.21-64b DRC Debug Environment (on micron4.eng.tau.ac.il)

cadence

File View Tools Help

Fit and Zoom 0.40 Pan None

Cell/Rule Filter Show All Show Errors Cell Sort by Name

Cell Name

+ MUX2X1 (E0)

Go to 1 / X Return On Top Explain

## Quantus Run (on micron4.eng.tau.)

The Quantus run "MUX2X1" completed successfully  
The output is in :



Library: Emilnassra\_final\_proj  
Cell: MUX2X1  
View: smart\_RCC\_typical

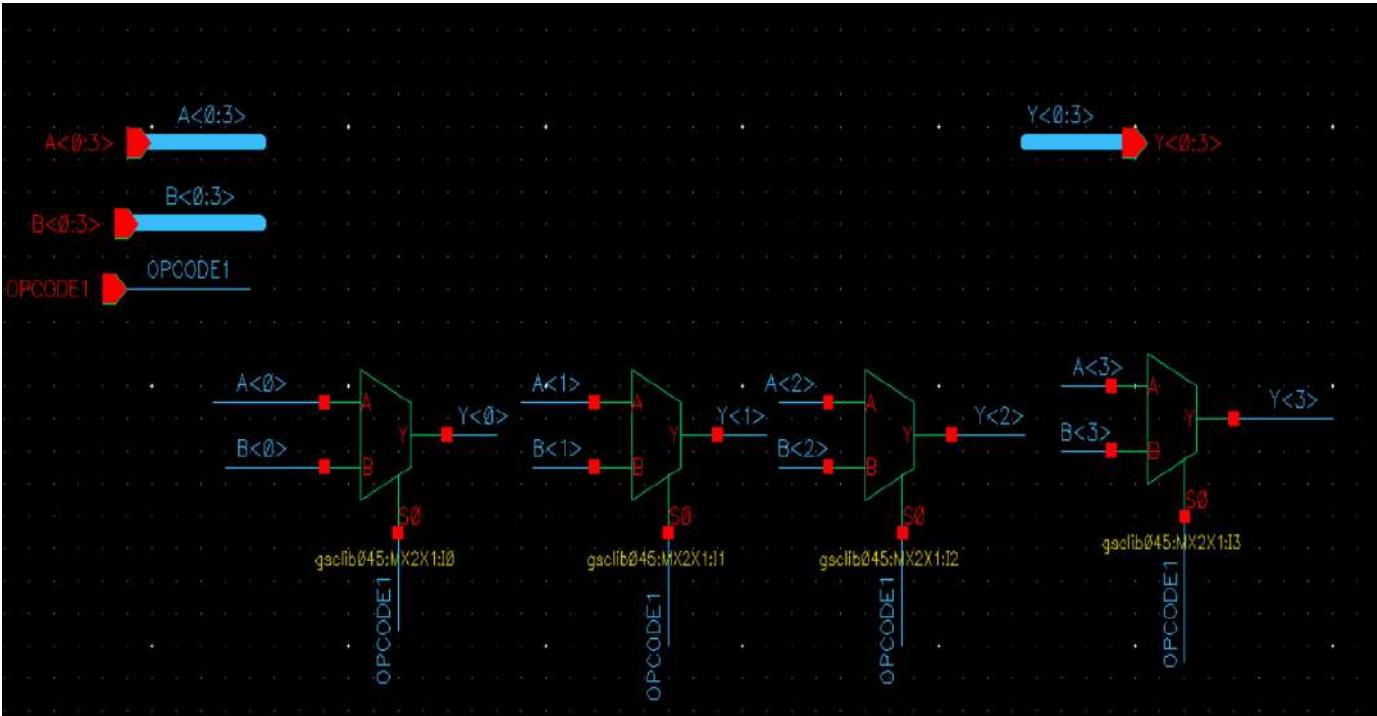
**Close**

## MUX 2X1NEG

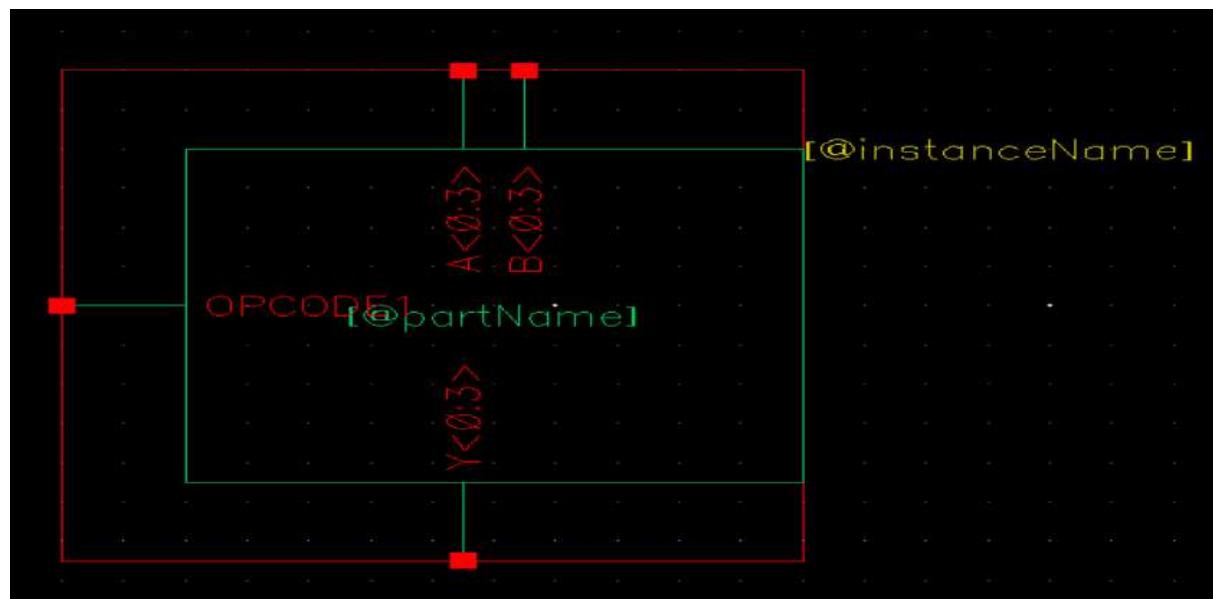
### Functionality

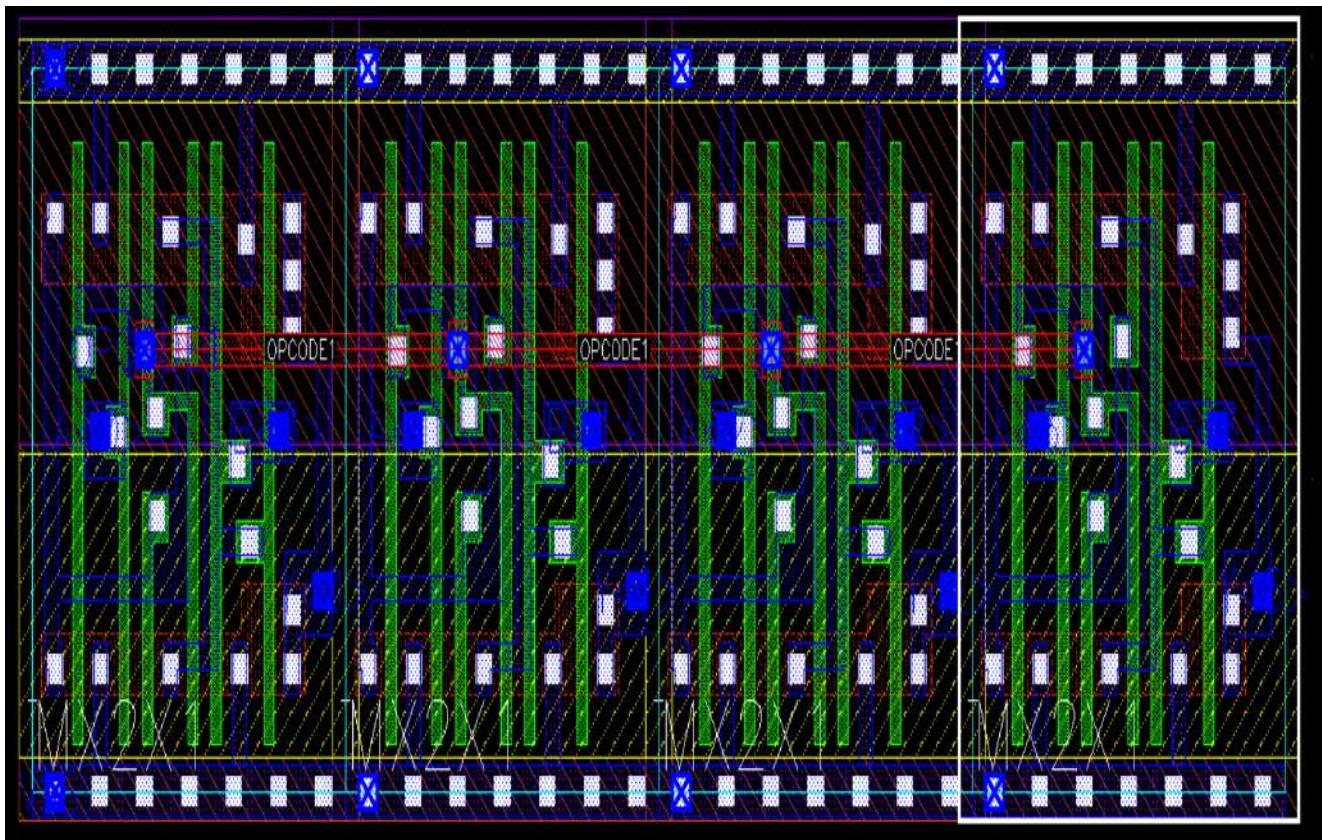
The multiplexer receives 2 data inputs - 4 bit each, and according to the select lines, gives a single output.

### Schematic



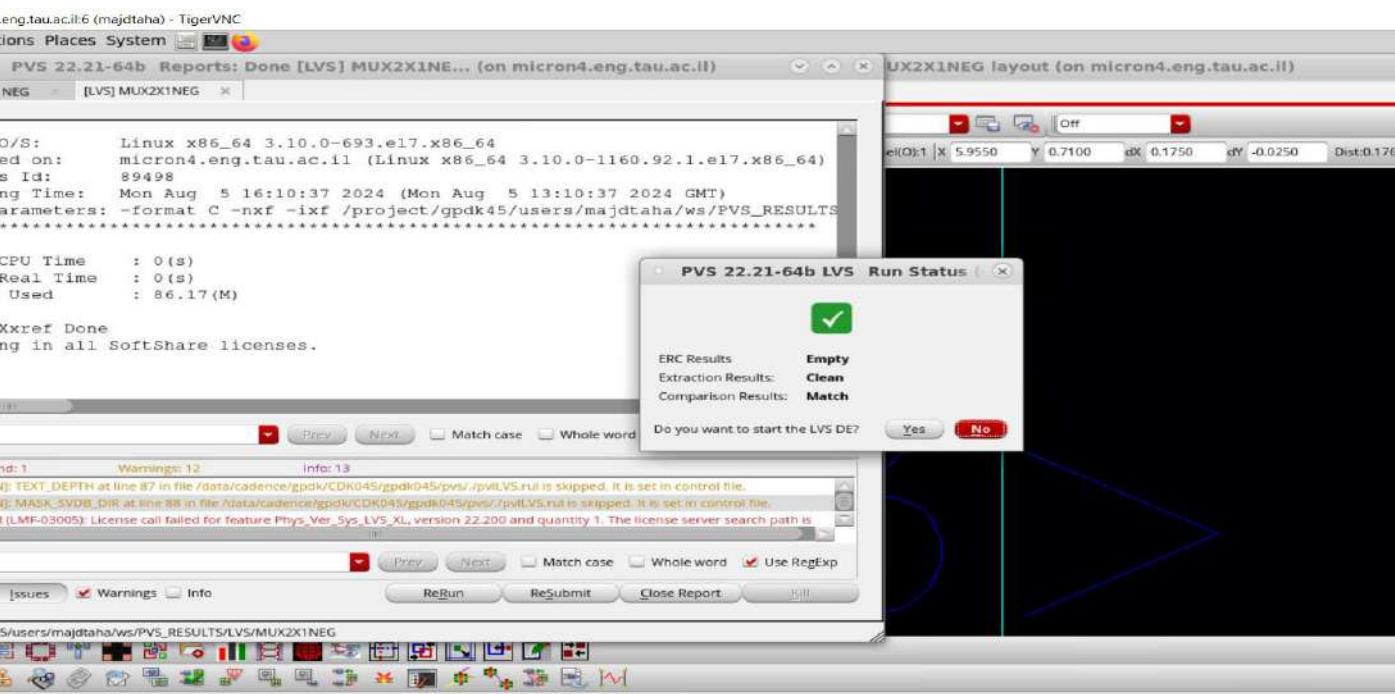
### Symbol & Layout





DRC , LVS , QUANTUS



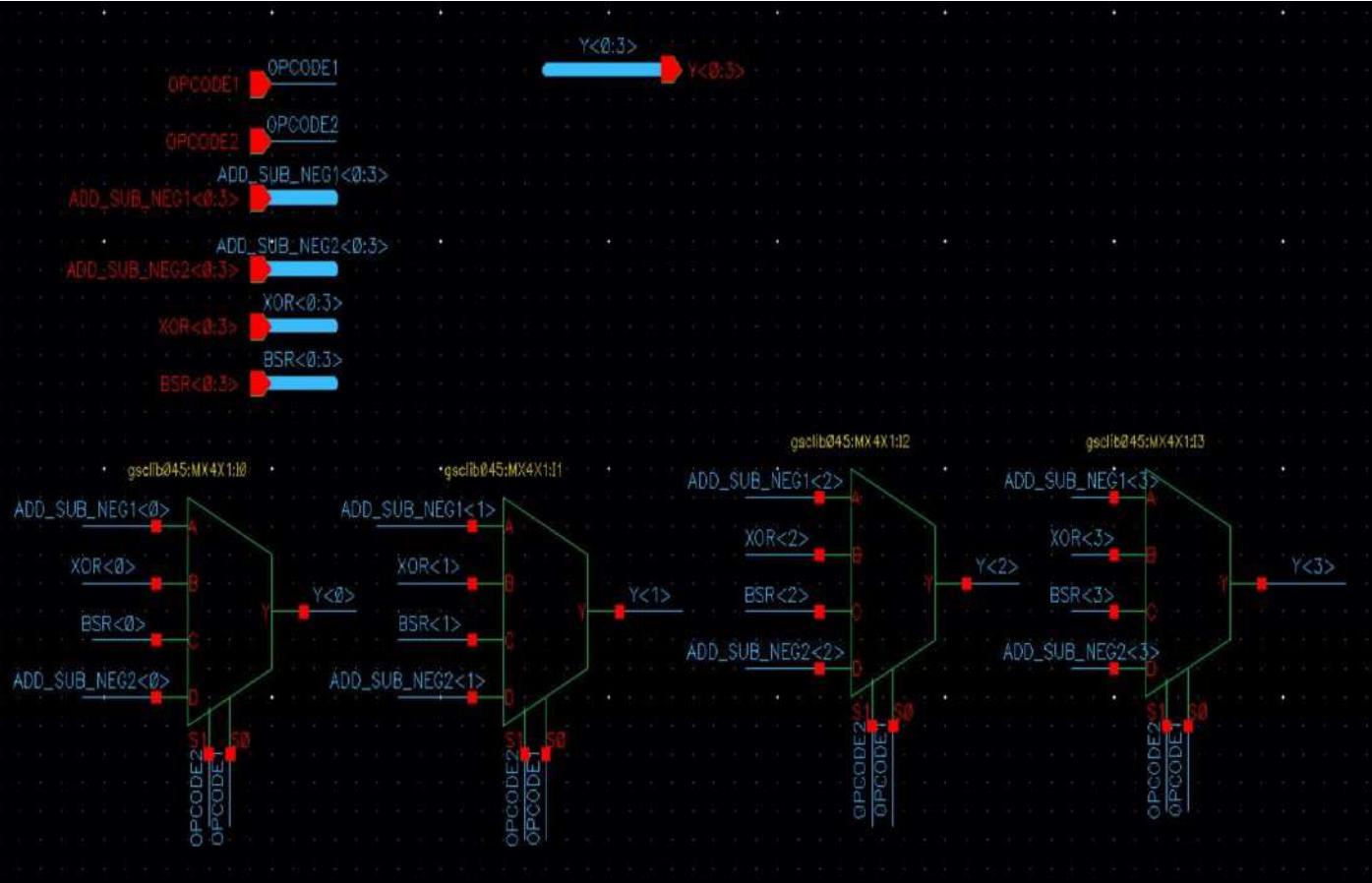


## MUX 4X1

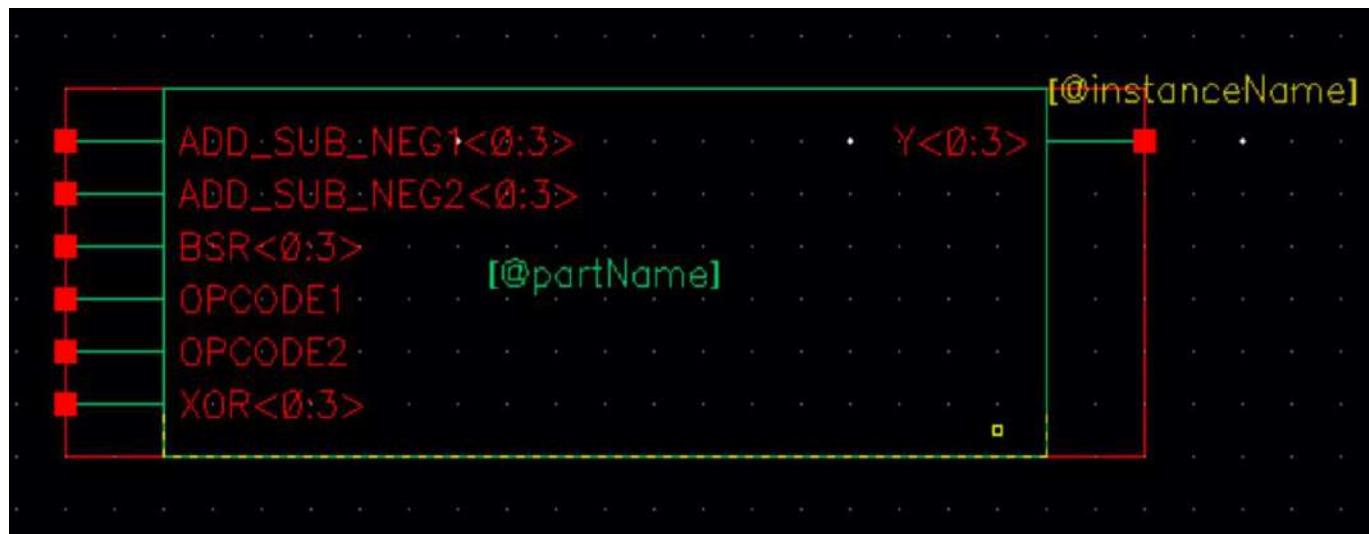
### Functionality

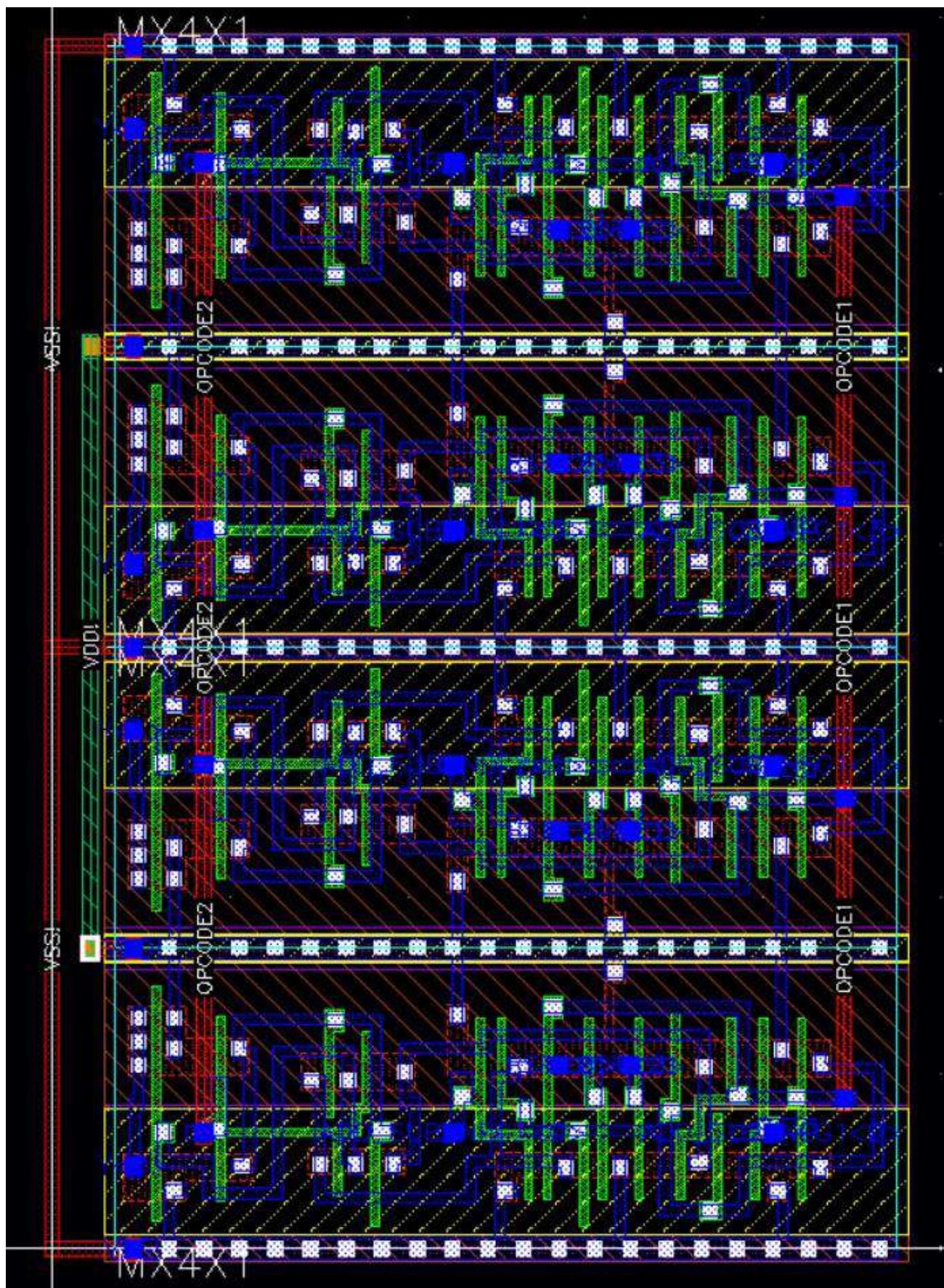
The multiplexer receives 4 data inputs - 4 bit each, and according to the select lines, gives a single output.

### Schematic

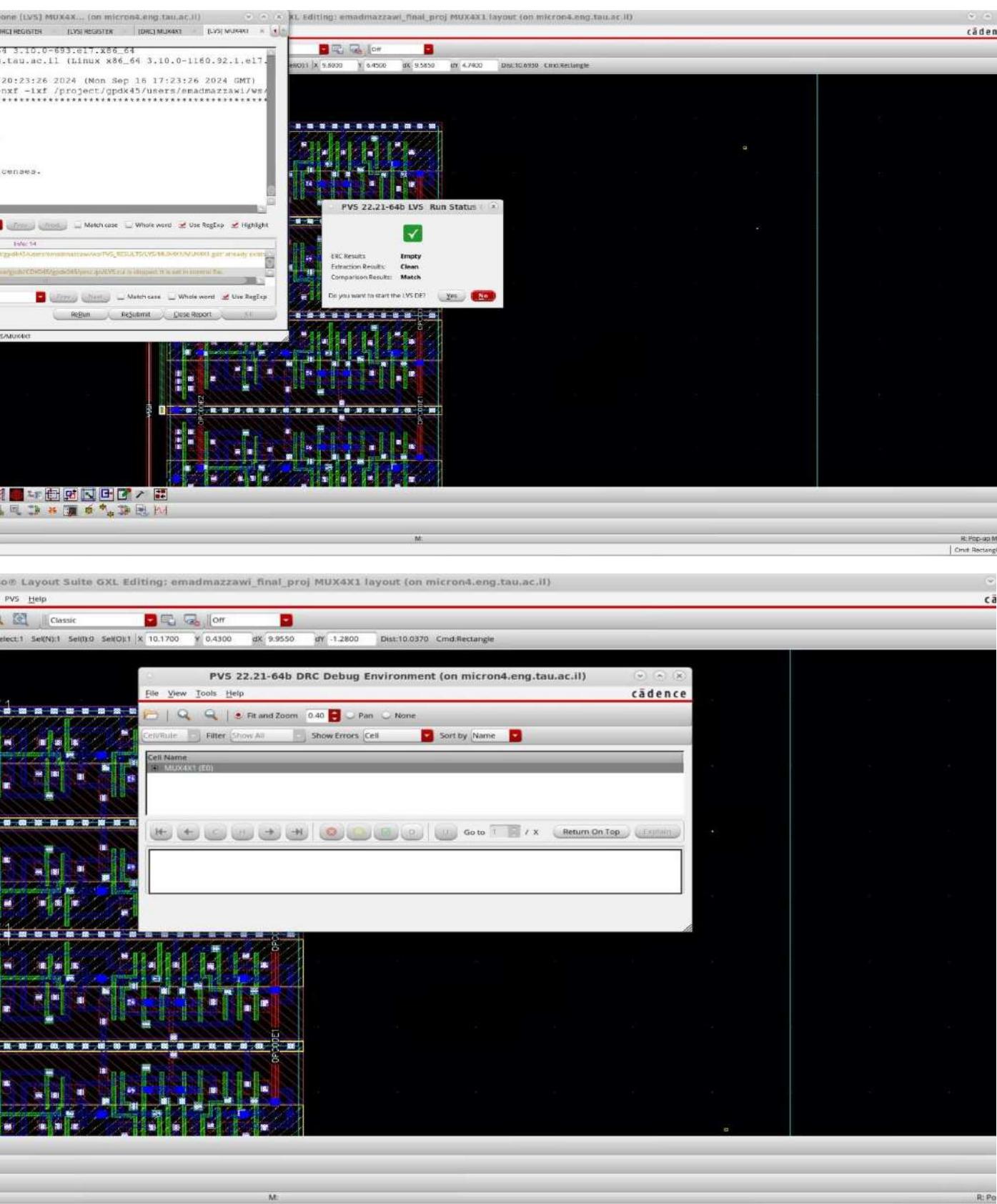


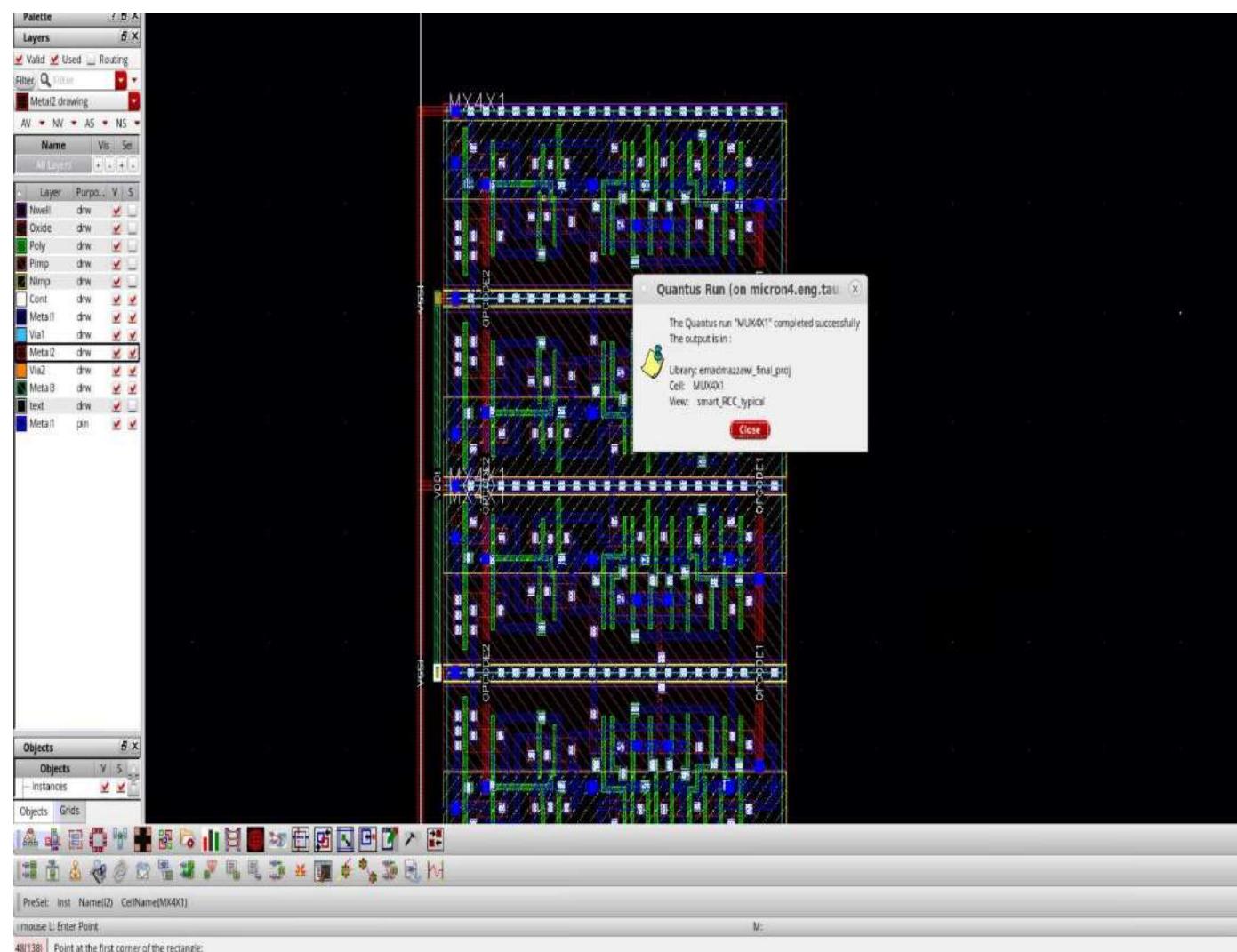
### Symbol & Layout





## DRC , LVS , QUANTUS

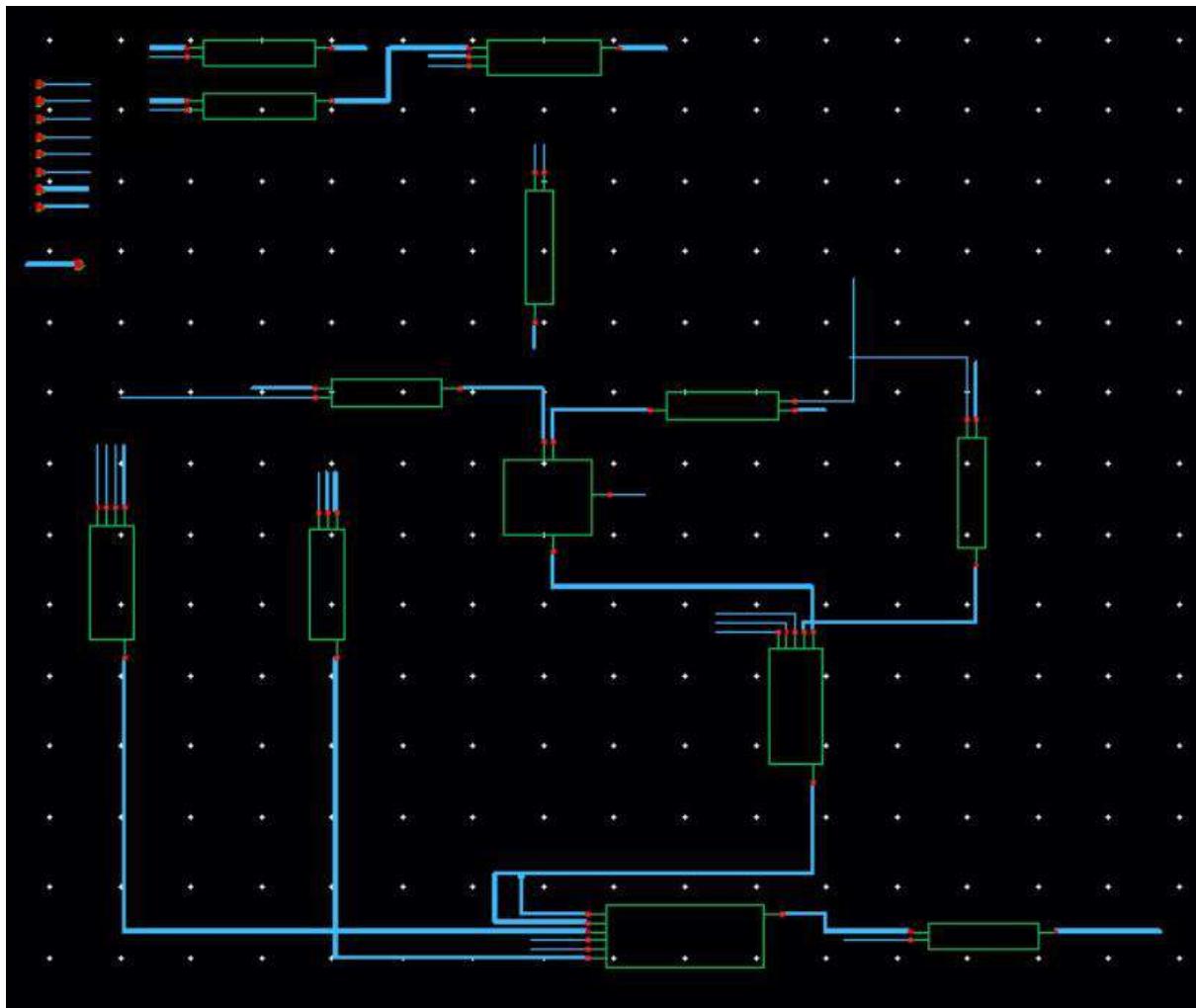




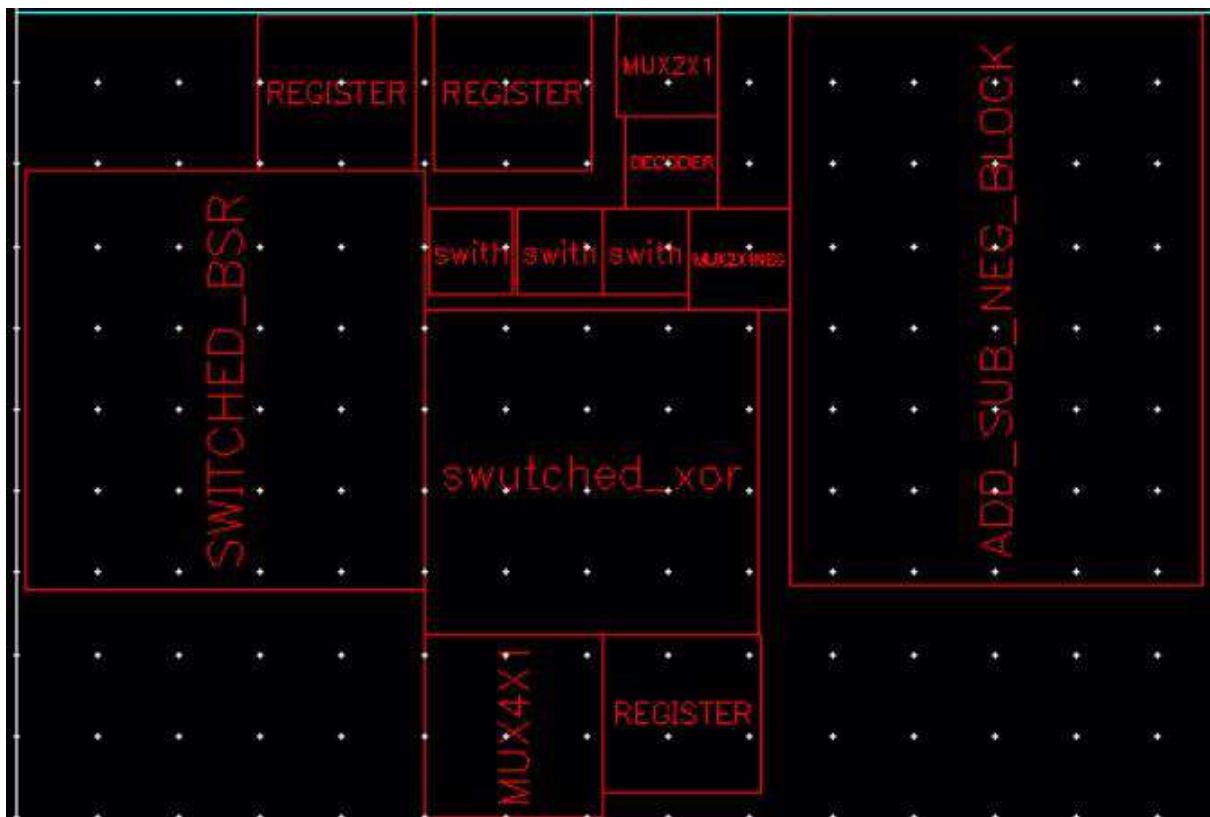
## ALU

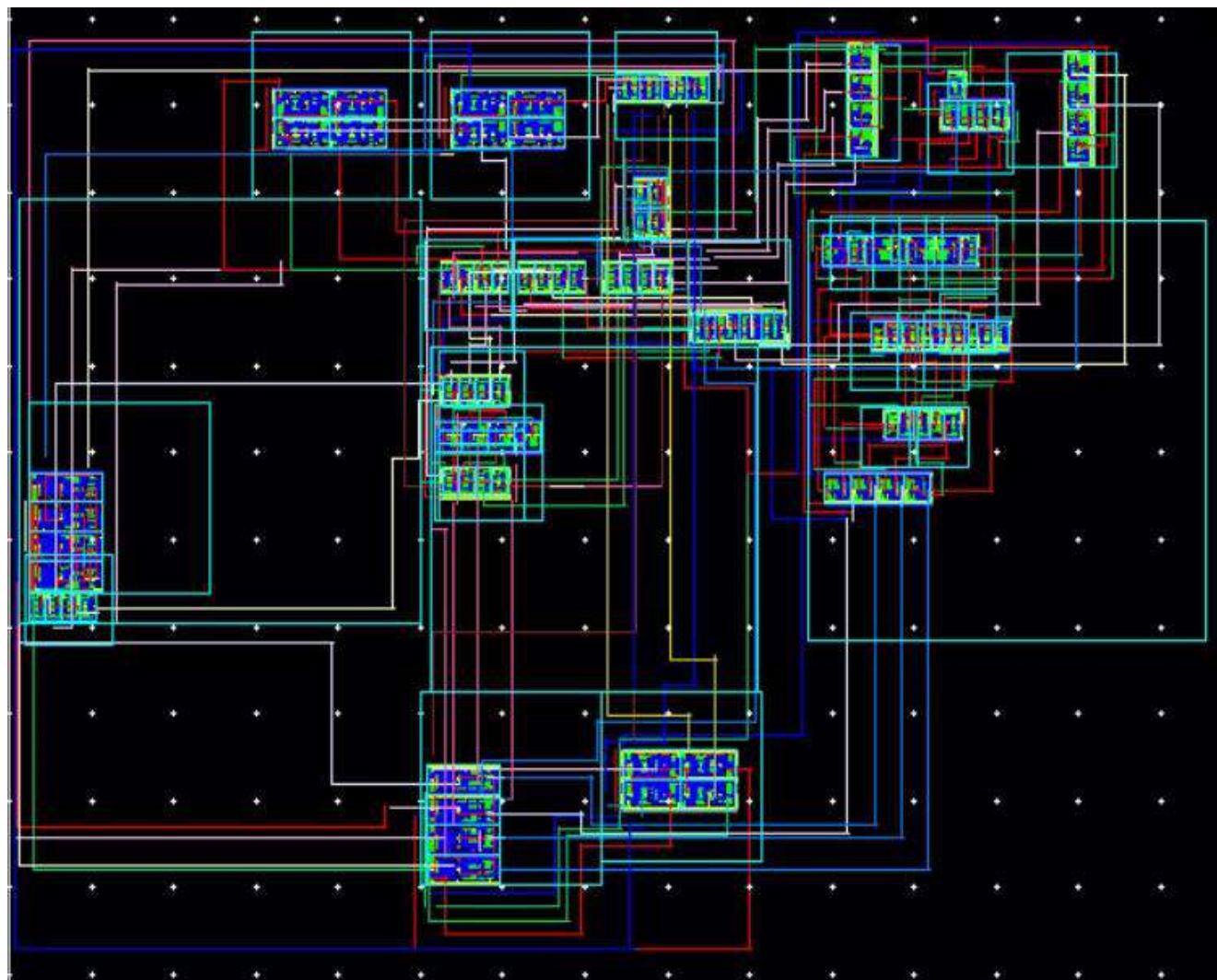
We merged all the components we built before into one big circuit, the ALU.

Schematic



Symbol & Layout & block-diagram





DRC , LVS , QUANTUS



'majdtaha/ws/PVS\_RESULTS/LVS/ALU/svdb/1  
\*\*\*

PVS 22.21-64b LVS Run Status ( X )



ERC Results      **Empty**  
Extraction Results:      **Clean**  
Comparison Results:      **Match**

Do you want to start the LVS DE?

**Yes**

**No**

Quantus Run (on micron4.eng.t) X

The Quantus run "ALU" completed successfully

The output is in :



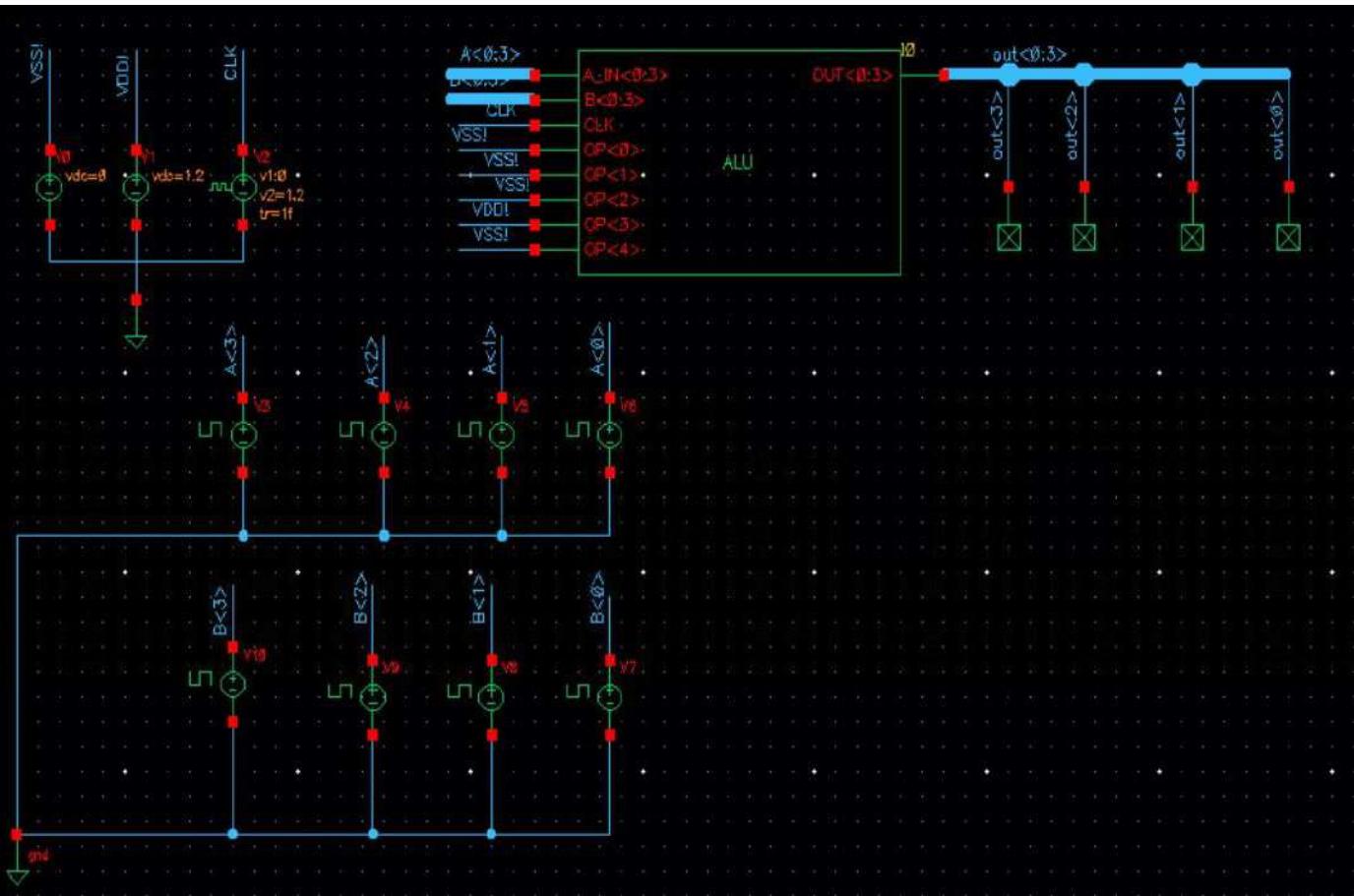
Library: majdtaha\_final\_proj

Cell: ALU

View: smart\_RCC\_typical

**Close**

### ALU test-bench:-

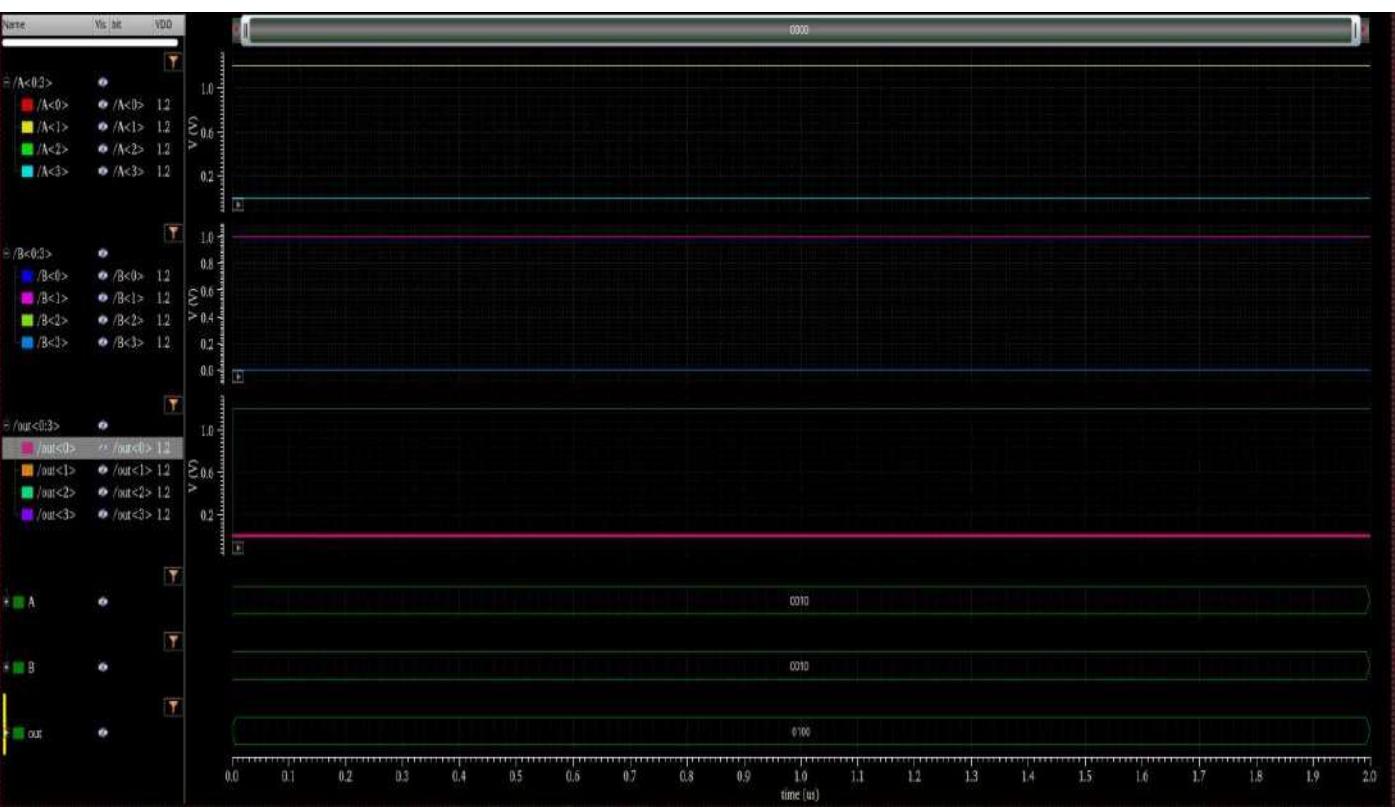


ADD Test - with opcode 00000 :

$$10+5=15$$



2+2=4



SUB Test with opcode 01000

7-3=4 :

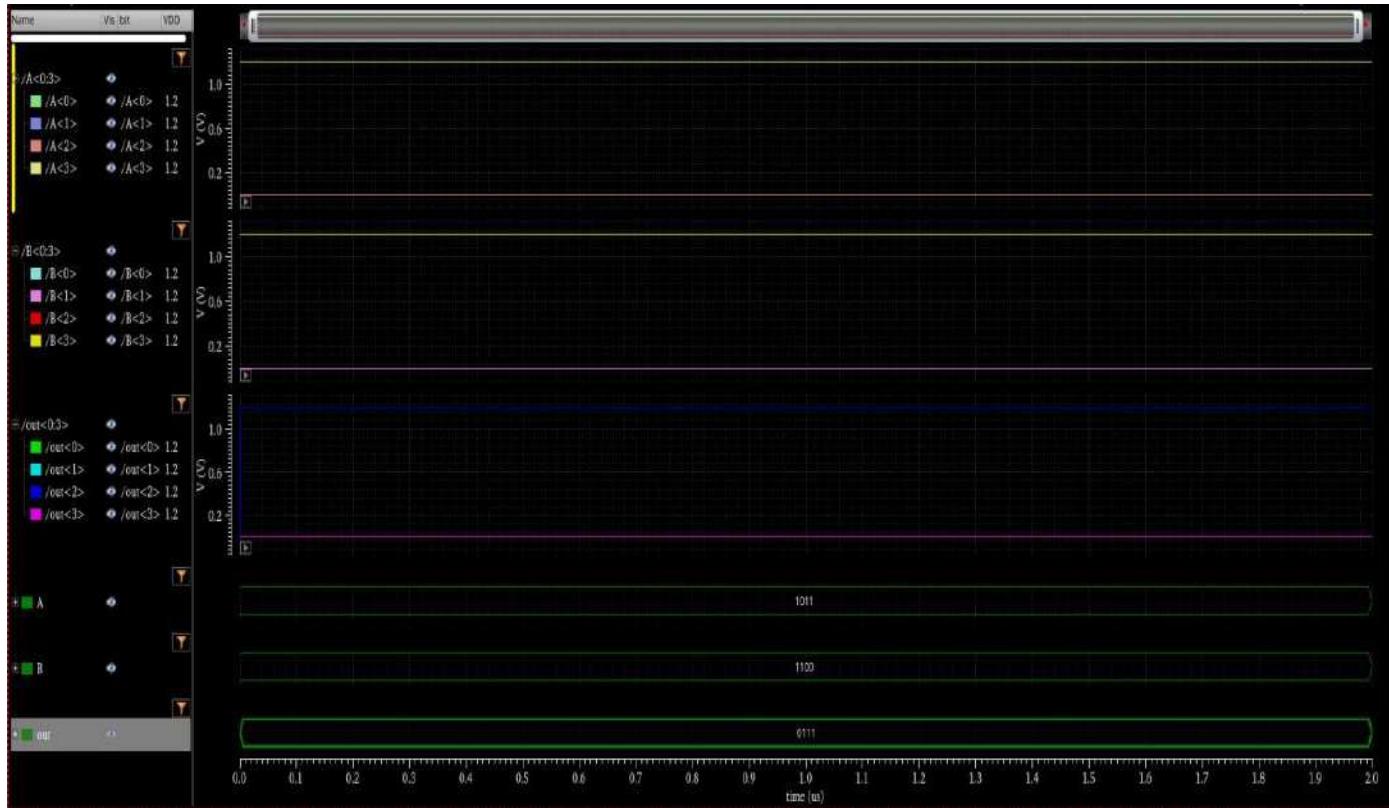


## Neg Test with opcode 01110

Neg(7)=-7 :



XOR Test with opcode 00010 :

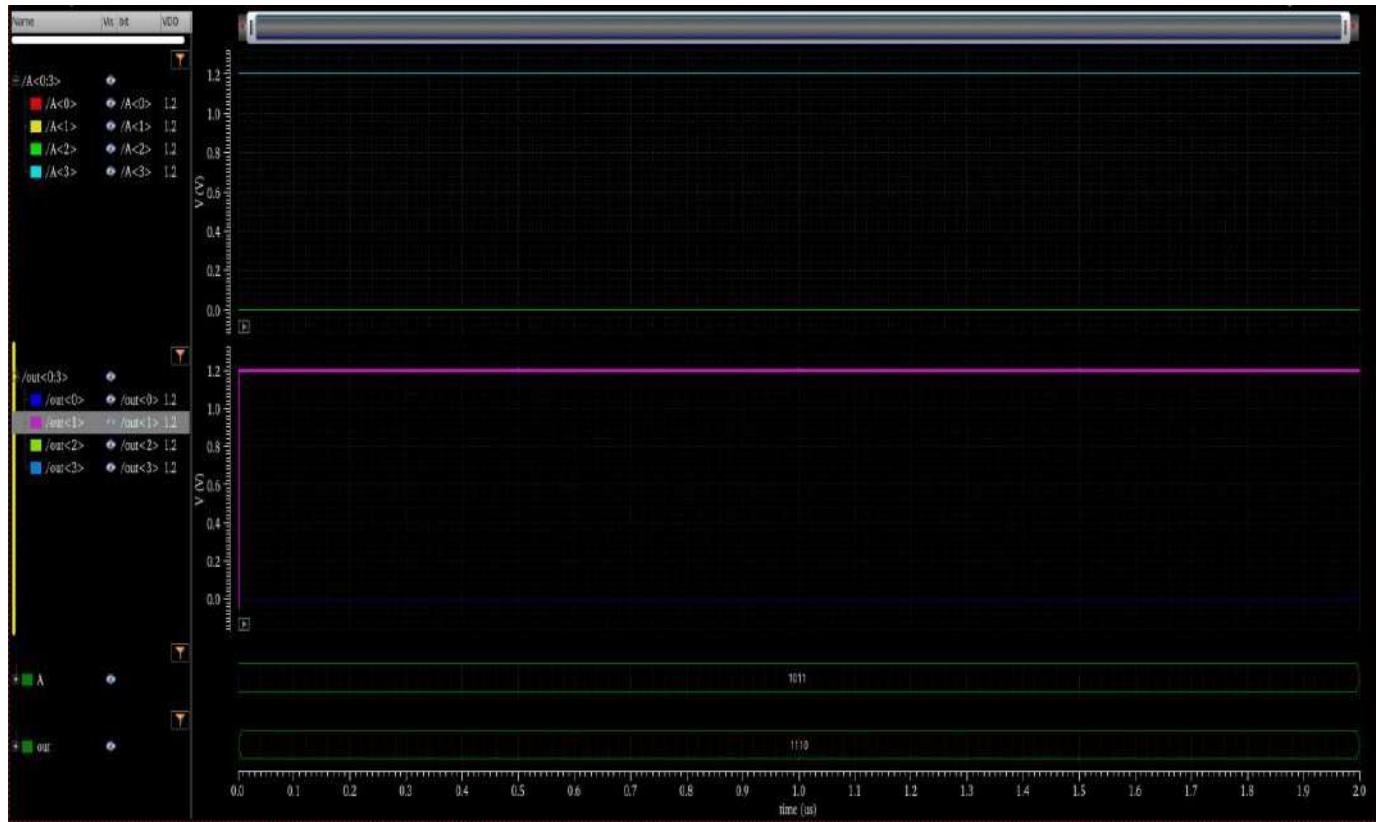


### BSR Test :

1 bit shift with opcode 01100



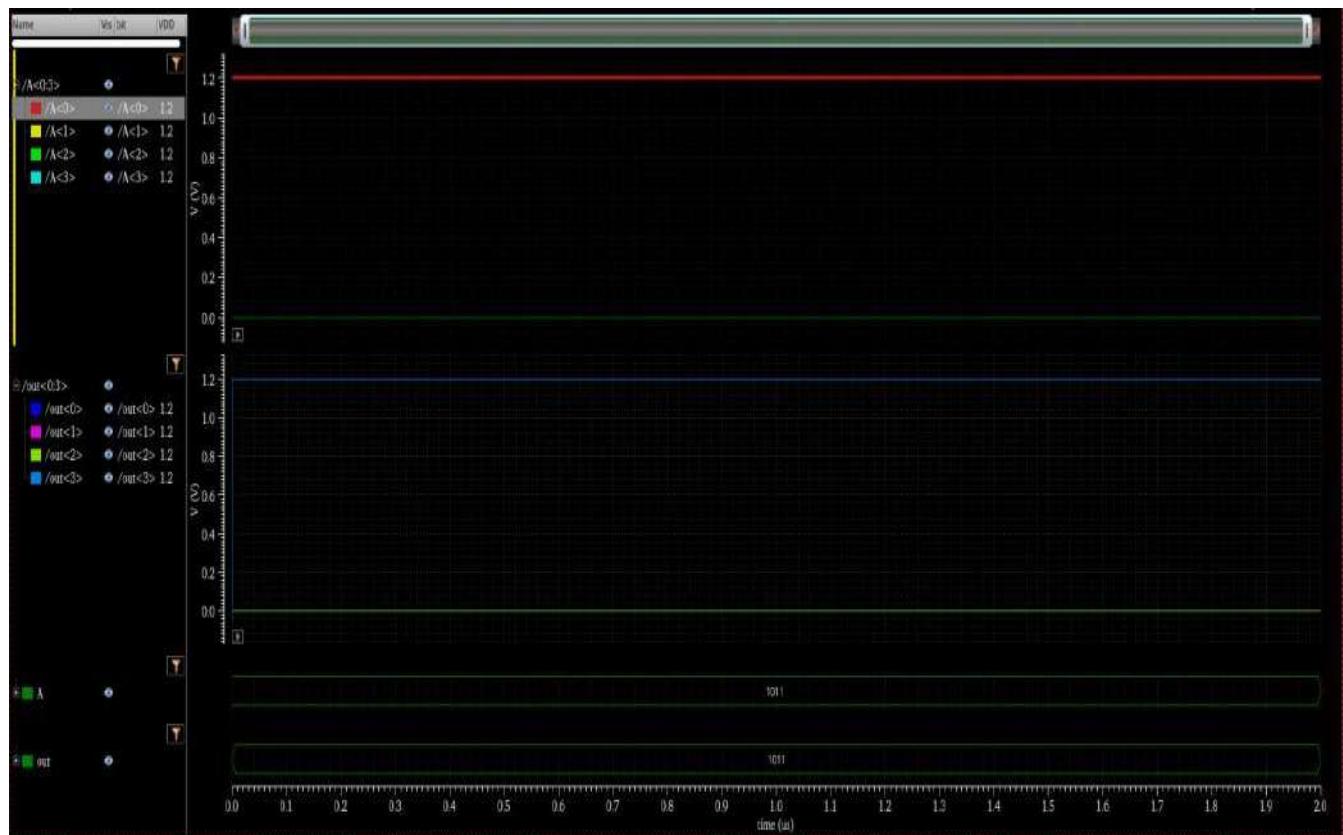
2 bits shift with opcode 10100



3 bits shift with opcode 11100



4 bits shift with opcode 00100



## Delay measurements

The critical path is along the ADD/SUB/NEG function.

We measured the delay for different supply voltages:

1.2 V:  $T_p(VDD=1.2V) = 41.27 \text{ ps}$

Virtuoso® ADE Assembler Editing: majdtaha\_final\_proj ALU\_test maestro (on micron4.eng.tau.ac.il)

File EAD Parasitics/LDE Window Help

Basic

No Sweeps Single Run, Sweeps and Corners Reference

ALU\_test maestro ALU ALU ALU

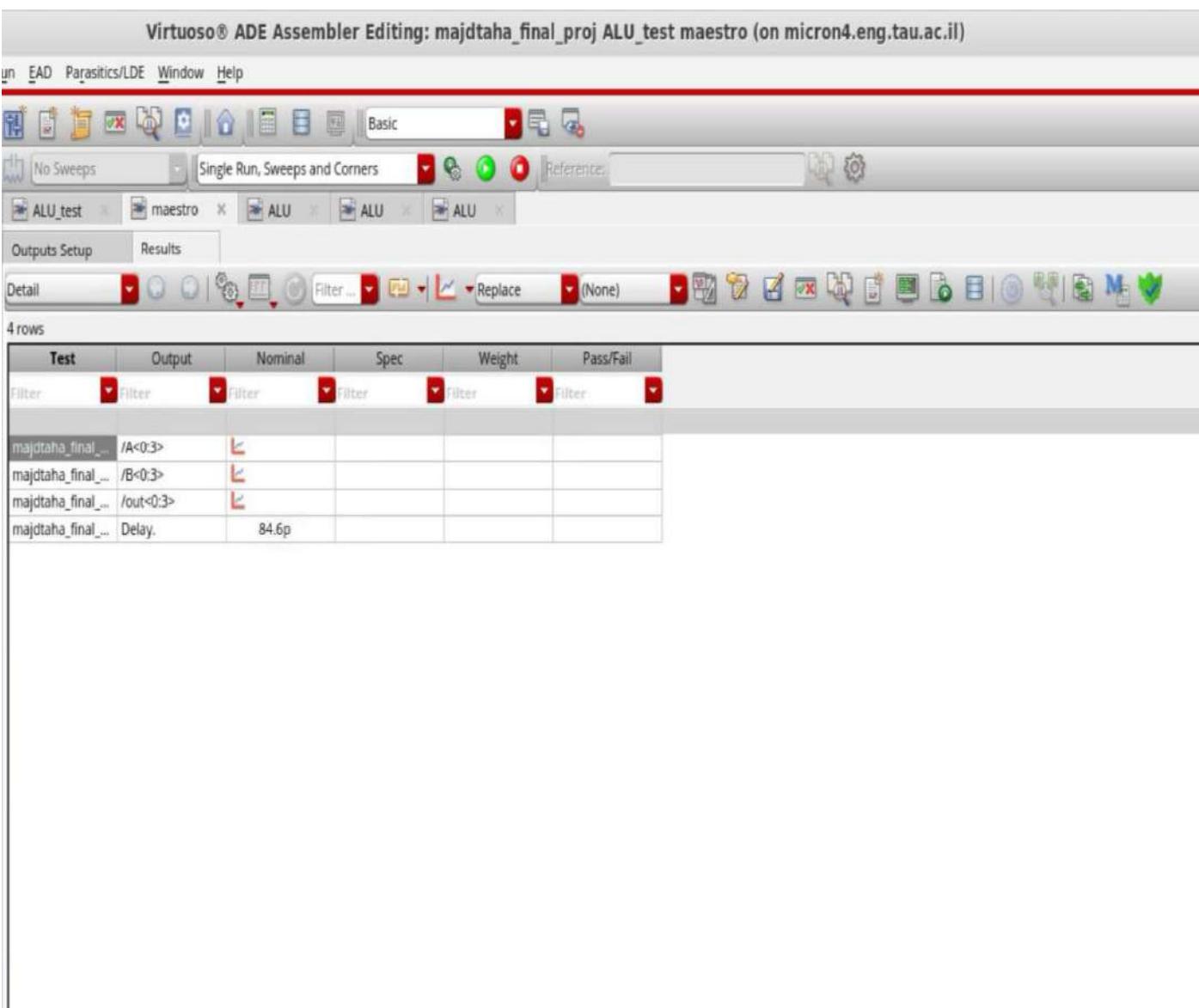
Outputs Setup Results

Detail Filter... Replace (None)

4 rows

Test	Output	Nominal	Spec	Weight	Pass/Fail
majdtaha_final_...	/A<0:3>	<input checked="" type="checkbox"/>			
majdtaha_final_...	/B<0:3>	<input checked="" type="checkbox"/>			
majdtaha_final_...	/out<0:3>	<input checked="" type="checkbox"/>			
majdtaha_final_...	Delay.	41.27p			

900mV:  $T_p(VDD=900mV) = 84.6 \text{ ps}$



We can see clearly that for the higher supply voltage value the delay time is lowest, and a longer delay time as we decrease the value of the supply voltage.

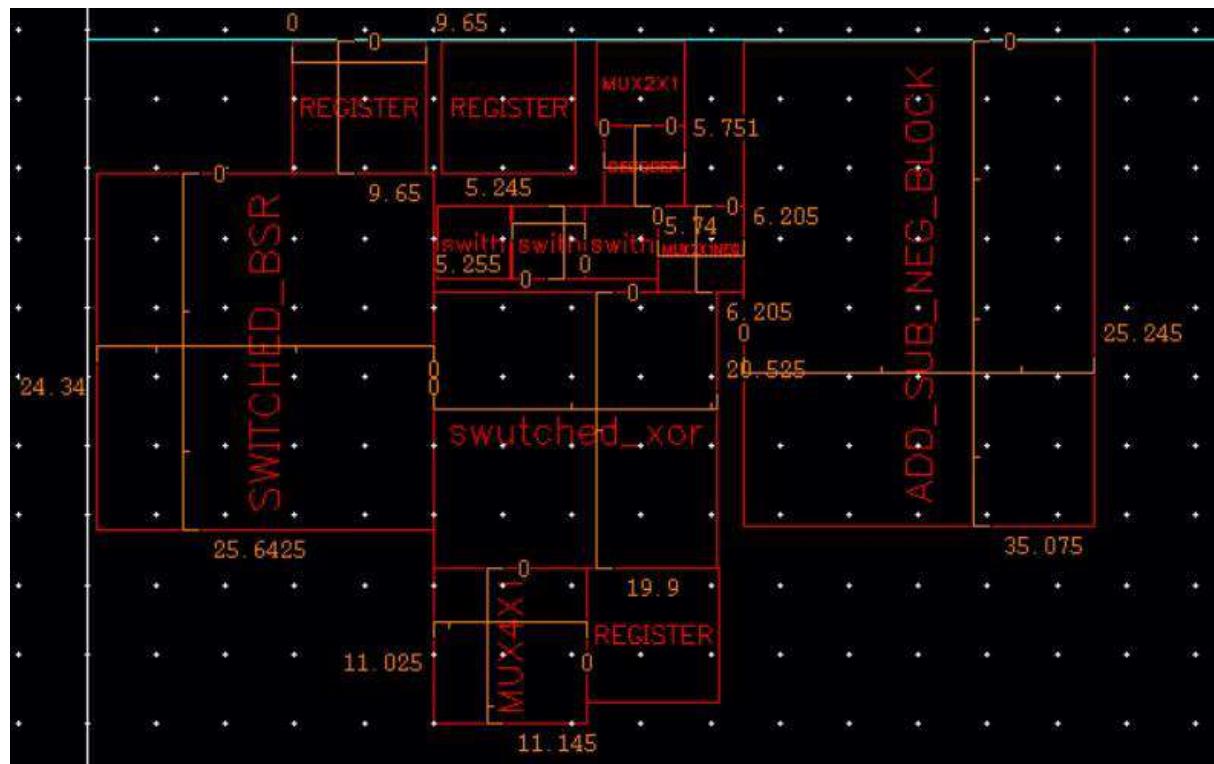
So the supply voltage affects the performance of the ALU, and subsequently the frequency of the clock.

#### Frequency measurements

$$VDD = 900mV \rightarrow f_{max} = 1/T_p(VDD=900mV) = 11.82 \text{ GHz}$$

$$VDD = 1.2V \rightarrow f_{max} = 1/T_p(VDD=1.2V) = 24.23 \text{ GHz}$$

Realization area:-



The area unit is : square(um).

Register- 9.3 , Decoder – 3.3 , MUX2X1 – 3.6 , MUX4X1 – 12.2 , Add/sub/neg - 88.5 , XOR – 40.8 , BSR – 62.3 , Switch – 2.7

Total ALU realization area: 238