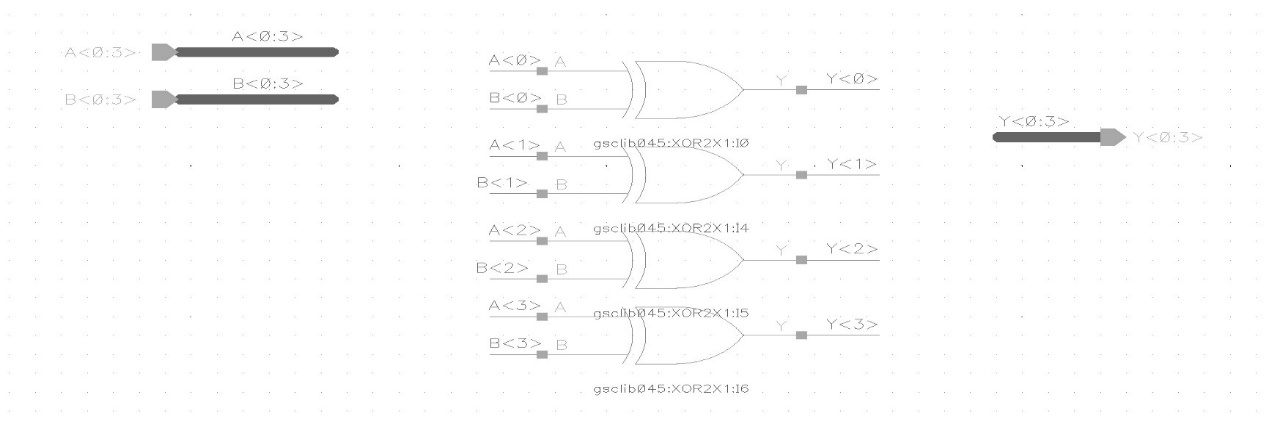
Section B:

**Cells schematic, layout and checks**

4bit-XOR:

A screenshot of a computer screen

Description automatically generated

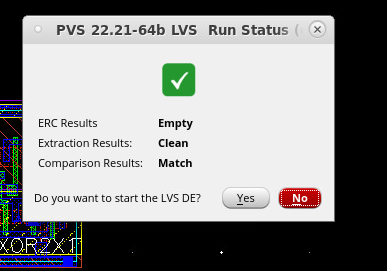
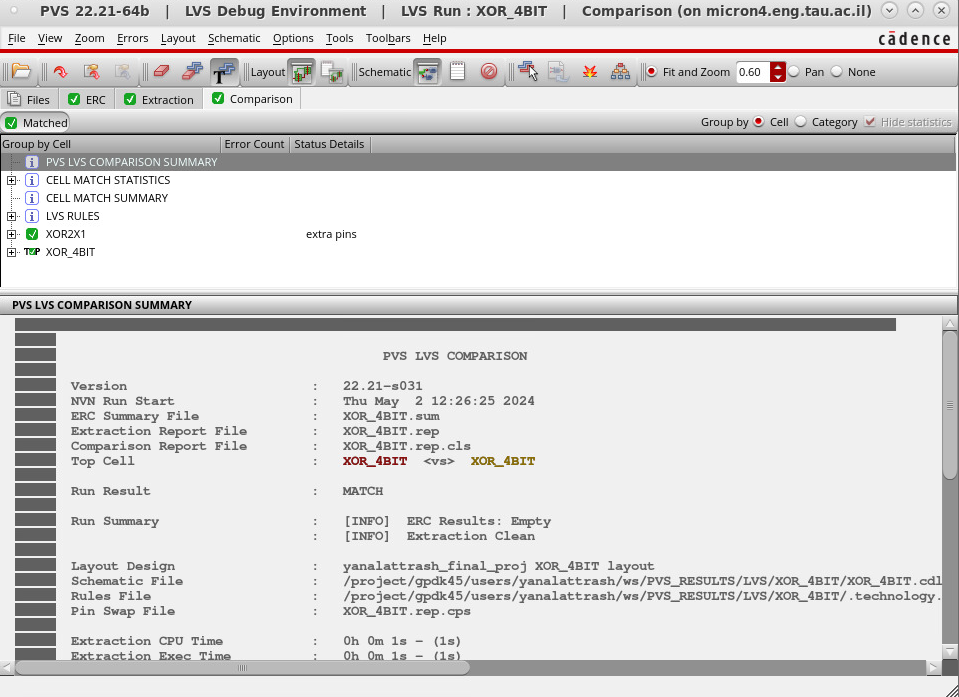
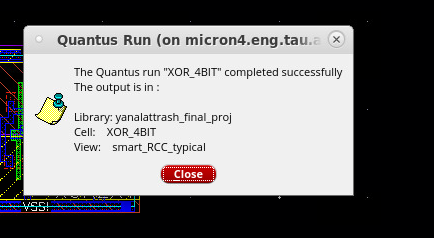


A close-up of a computer screen

Description automatically generated

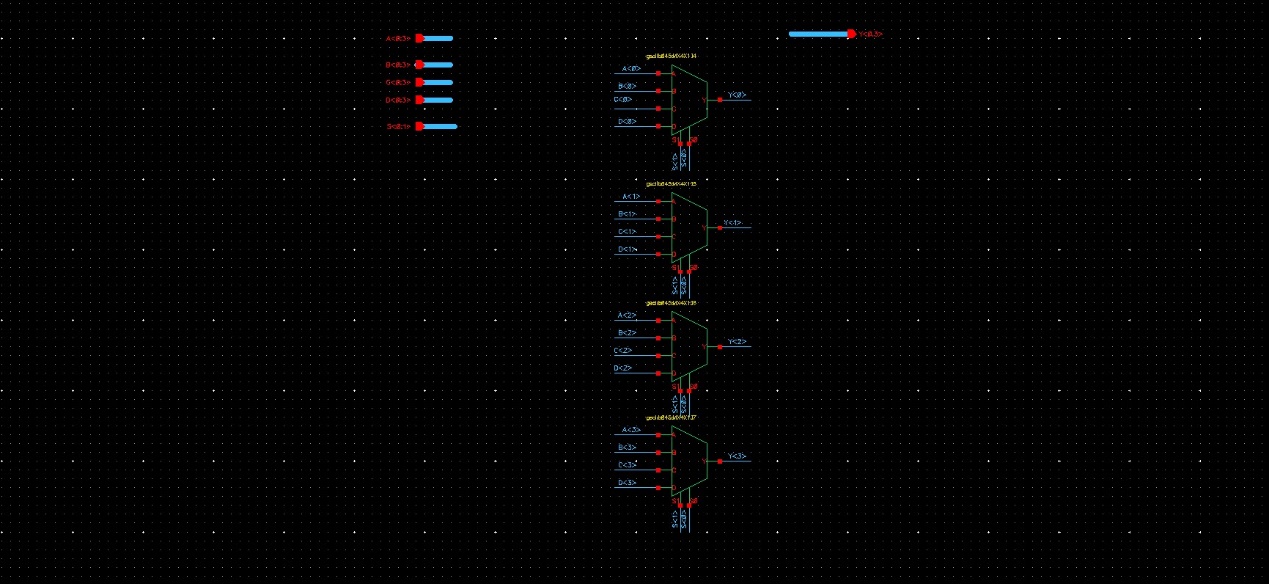
Size=(1.9)\*(7.8)=14.82

Checks:

A screenshot of a computer

Description automatically generated

MUX4-1- 4BITS



A white paper with black lines

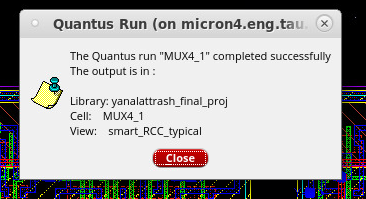
Description automatically generated

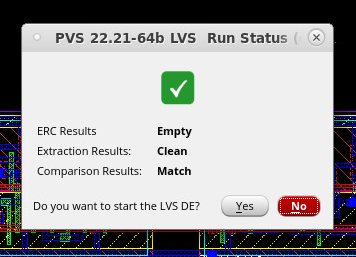
A close-up of a circuit board

Description automatically generated

Size=(9.49)\*(3.66)=34.737

Checks:

A screenshot of a computer

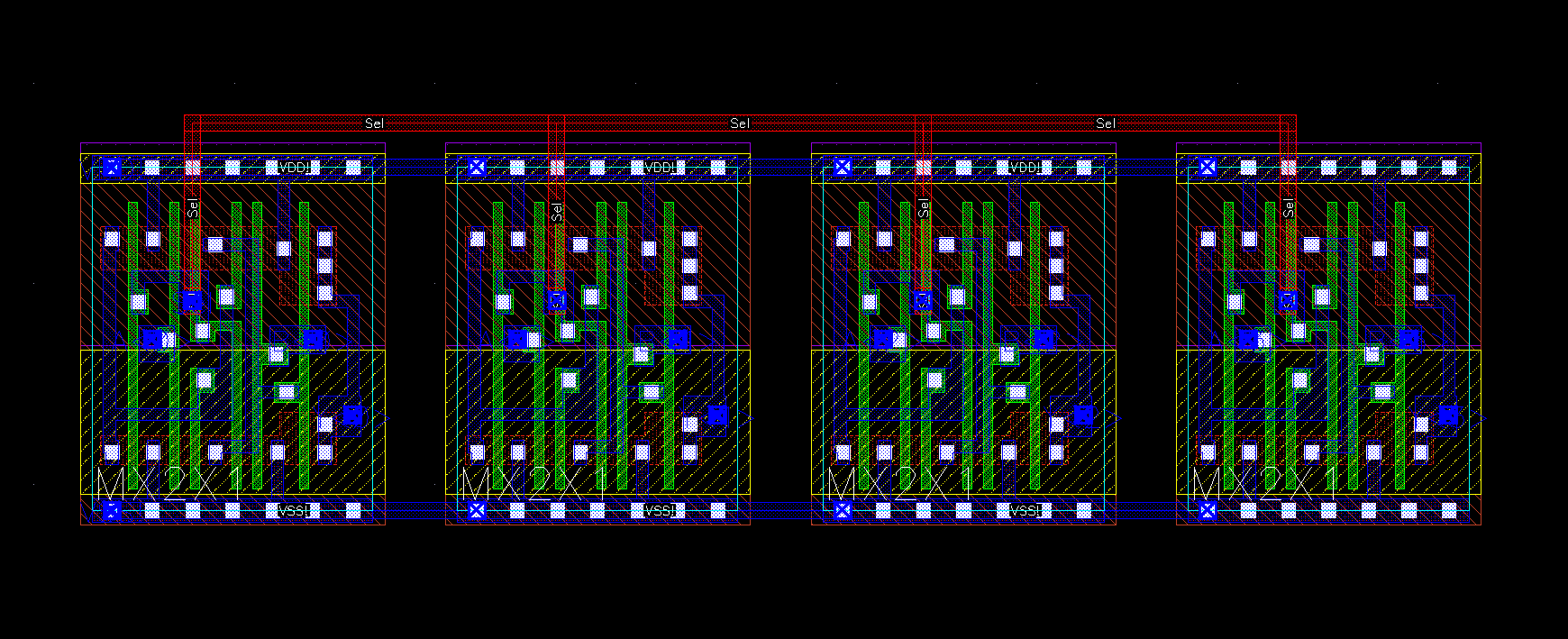
Description automatically generatedA screenshot of a computer

Description automatically generated

Mux2 1 -4bitA close-up of a diagram

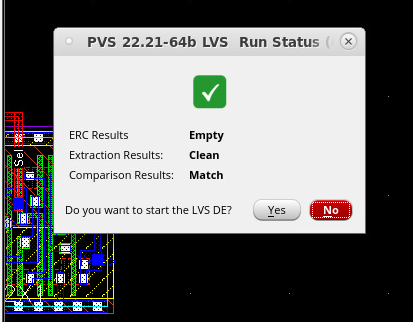
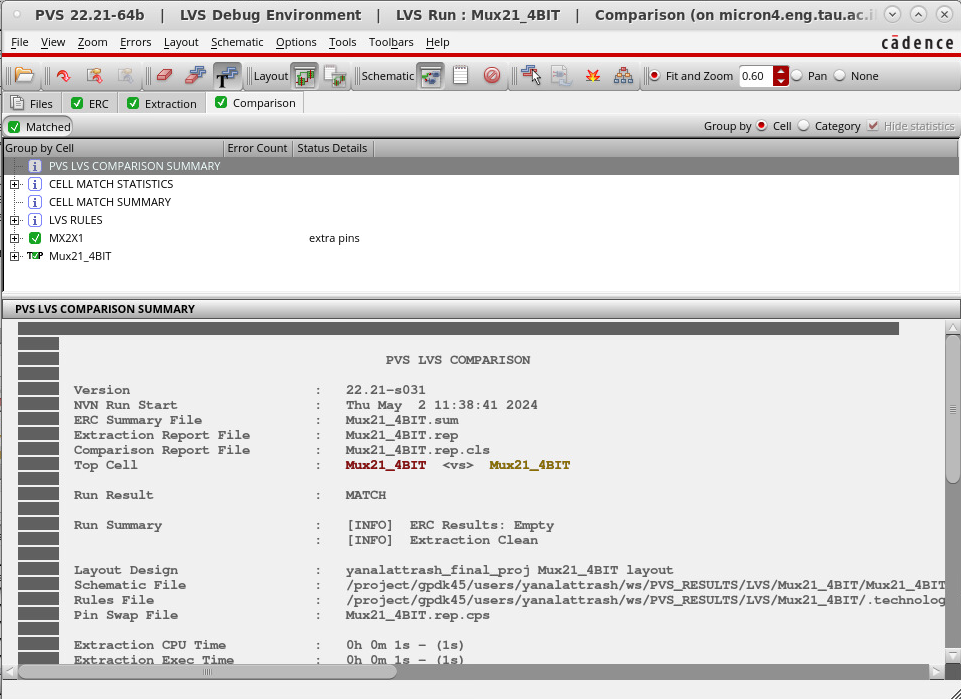
Description automatically generatedA screenshot of a computer program

Description automatically generated



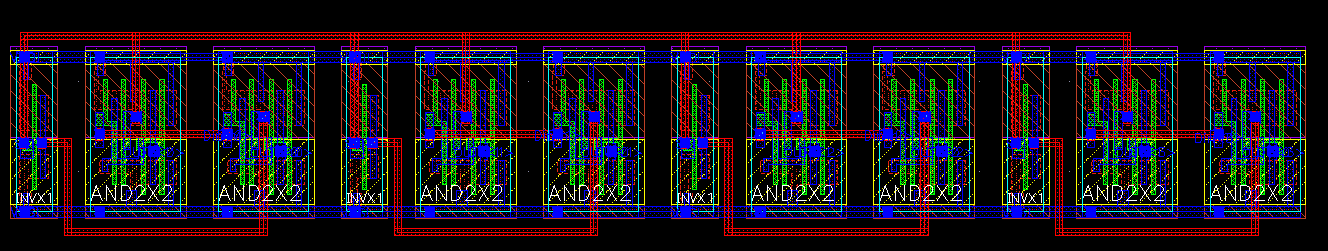
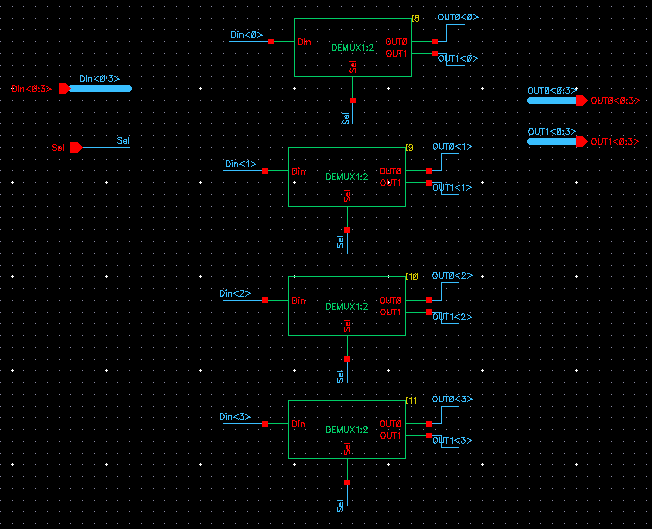
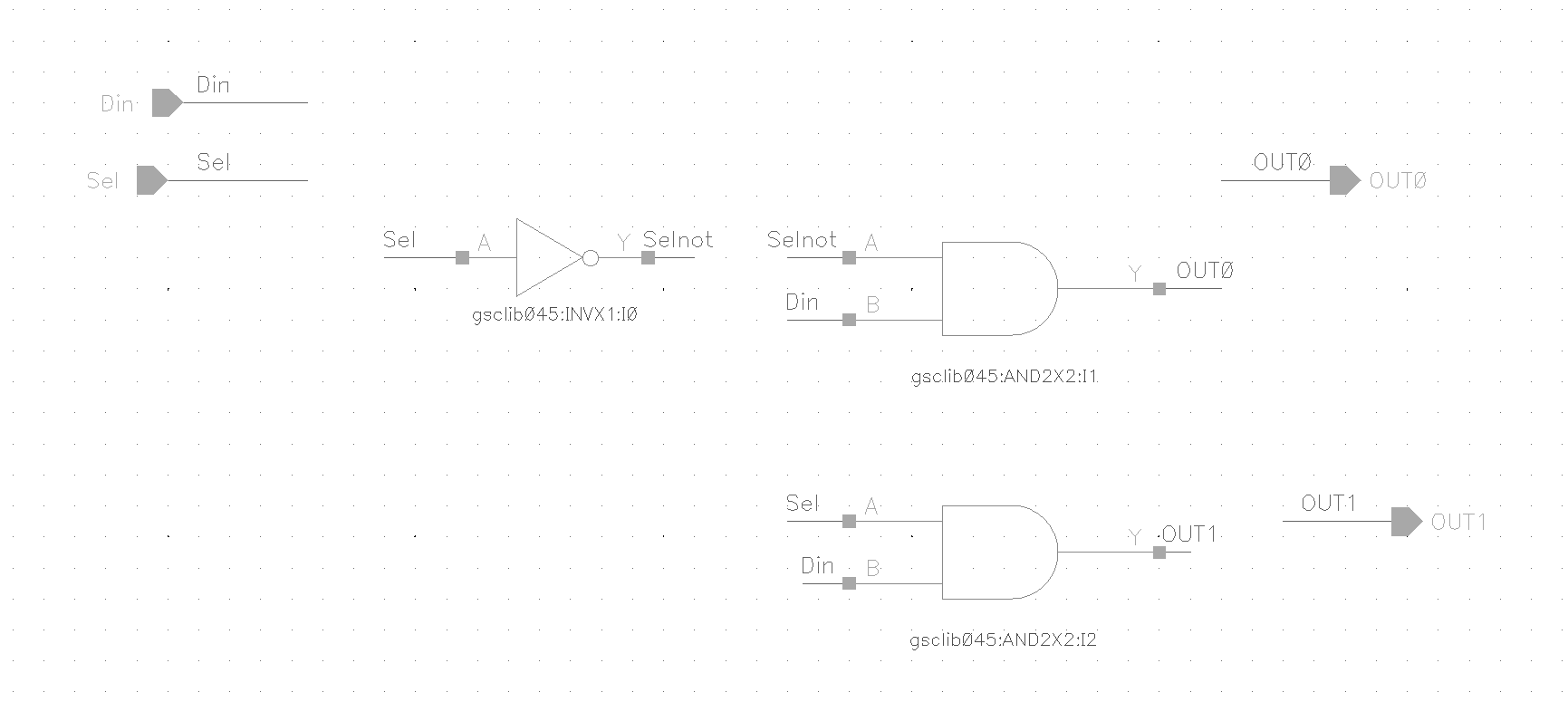
Size=(6.985)\*(2.04)=14.2494

Checks:

A screenshot of a computer error message

Description automatically generated

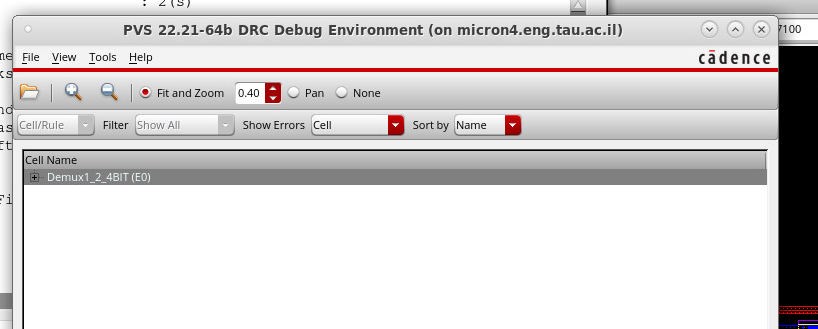
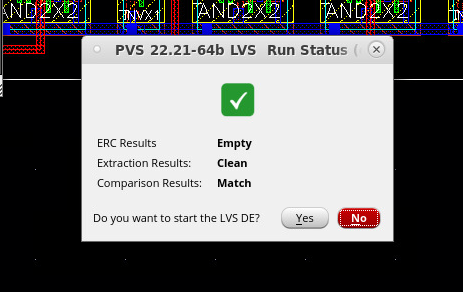
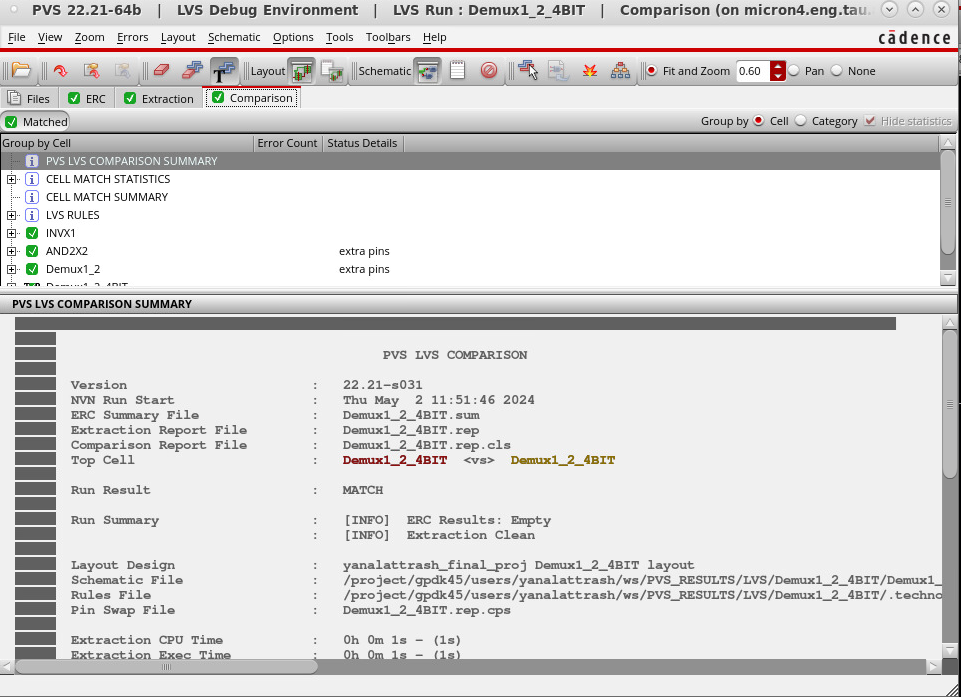
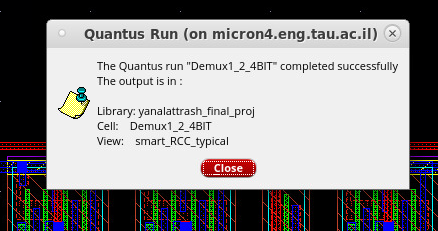
+ Demux1:2 Demux1:24bit



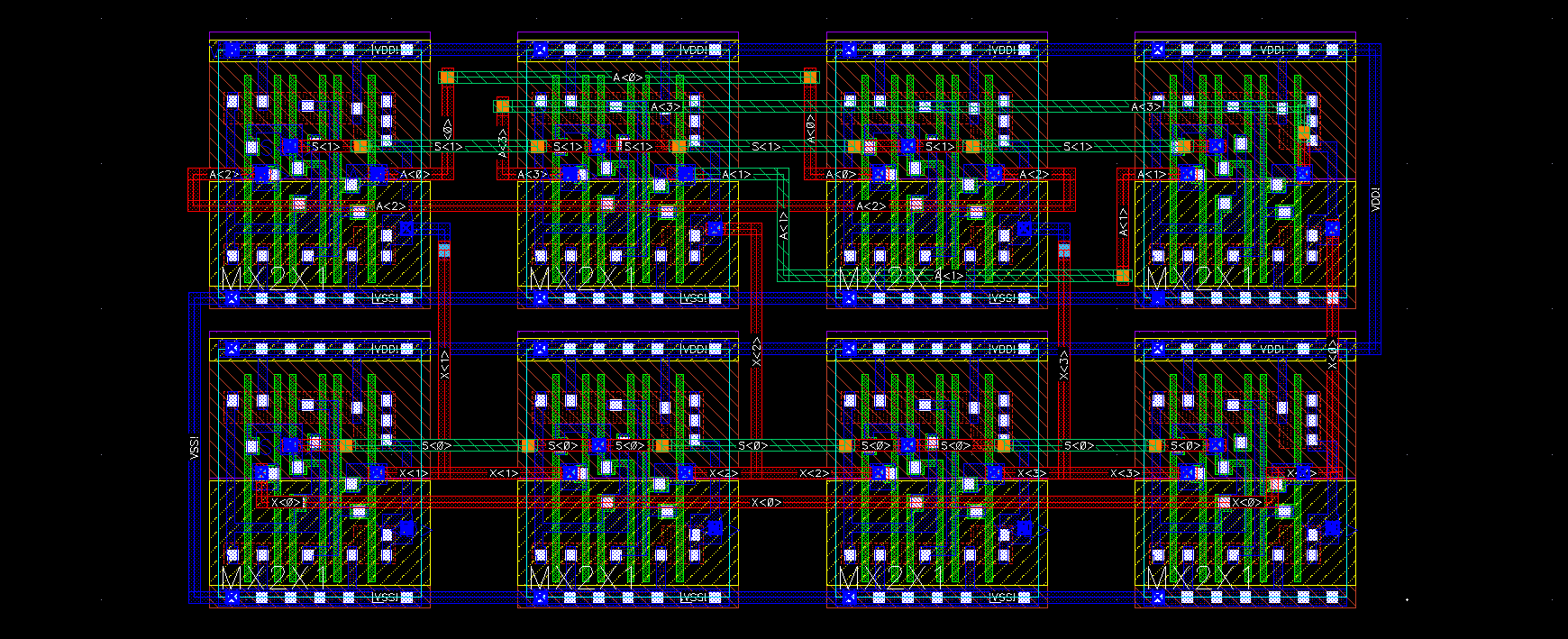
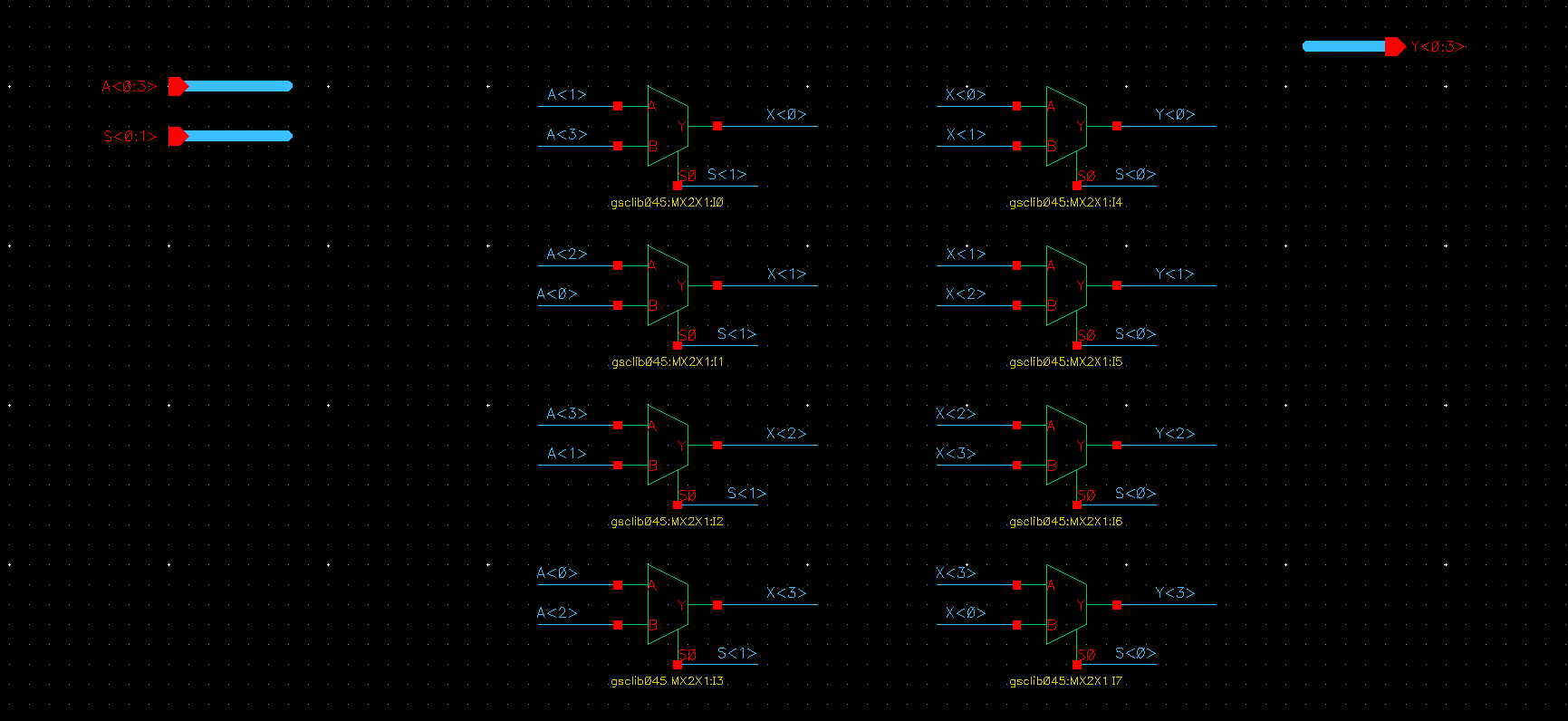
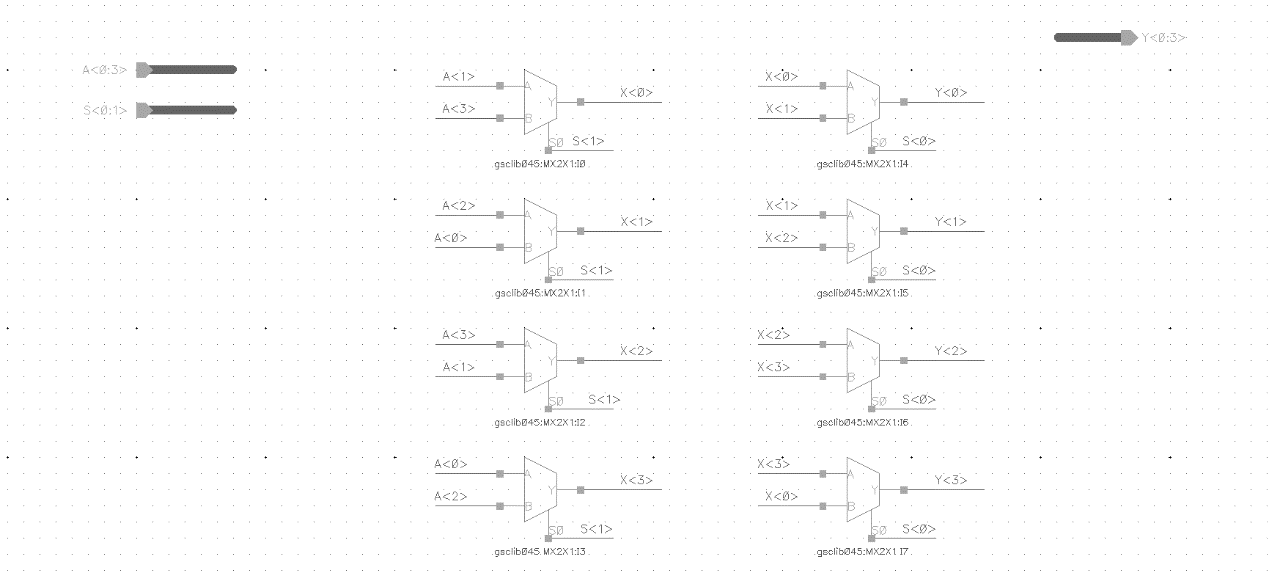
Demux1:2-4bit Size=(14.37)\*(2.25)=32.3325

Demux1:2 Size =(3.365)\*(2.25)=7.57125

Check:

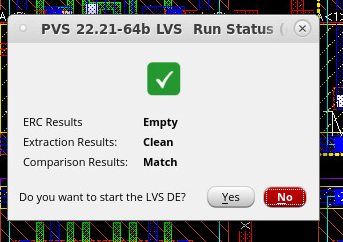
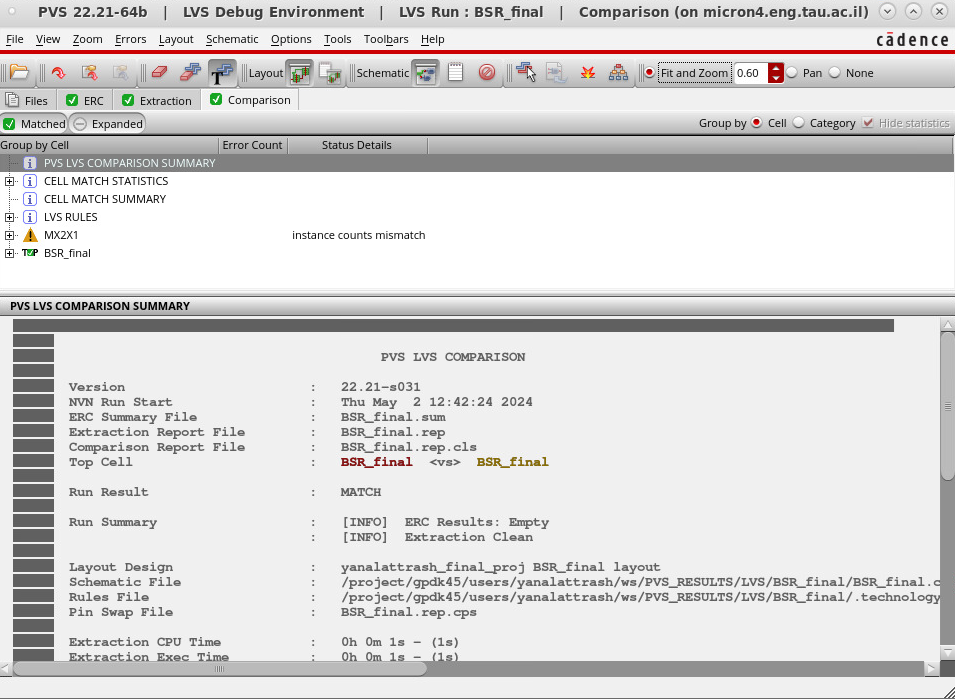
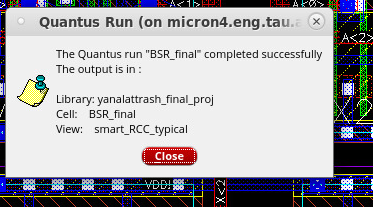


BSR



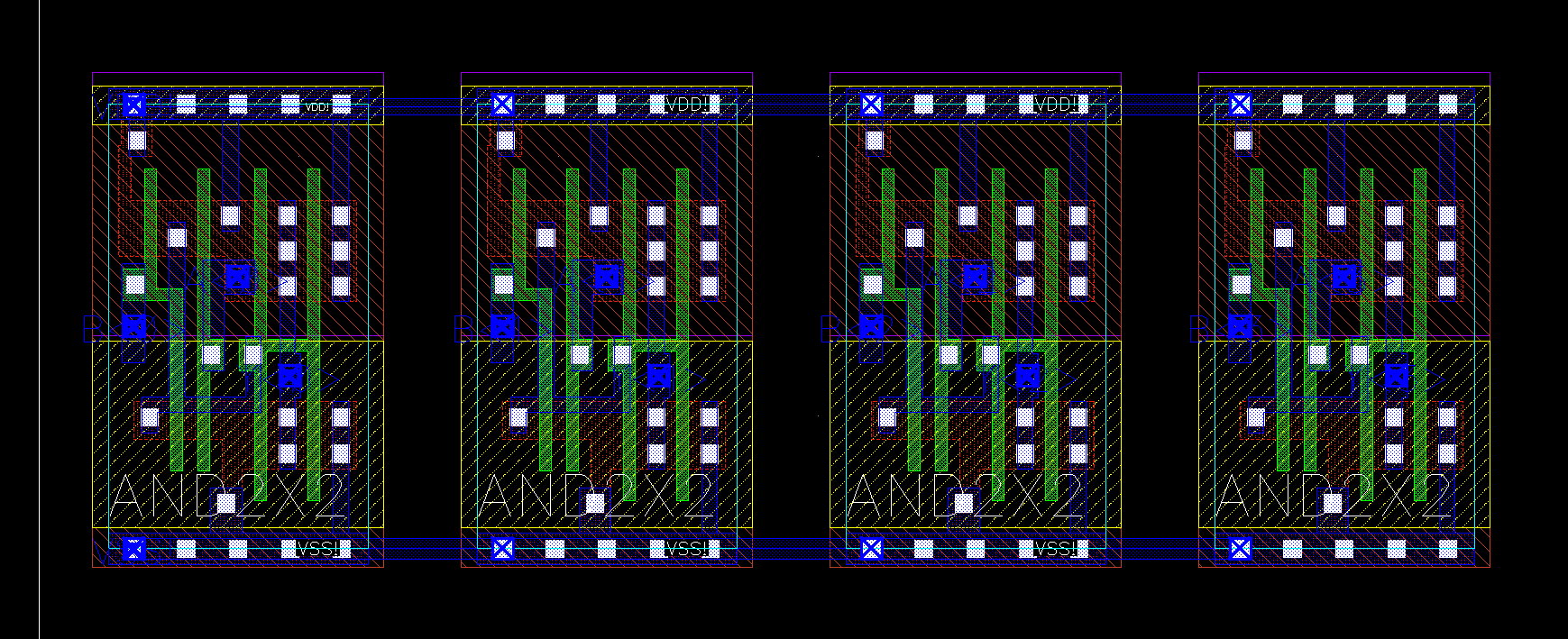
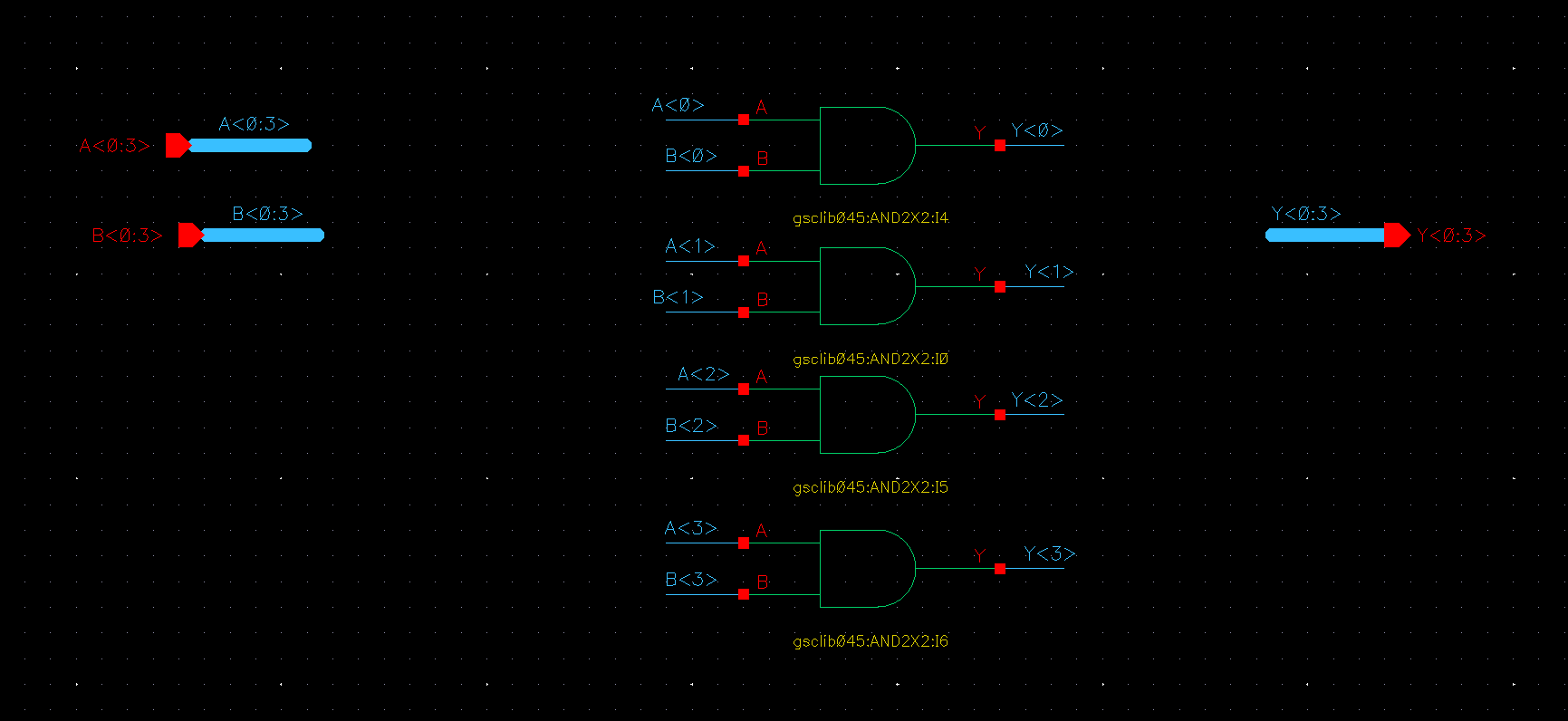
Size=(8.37)\*(4.225)=35.6144

Checks:

A screenshot of a computer error message

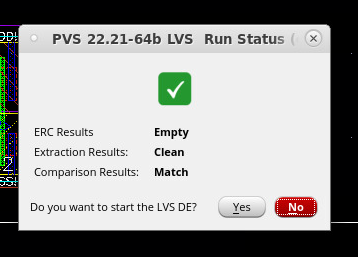
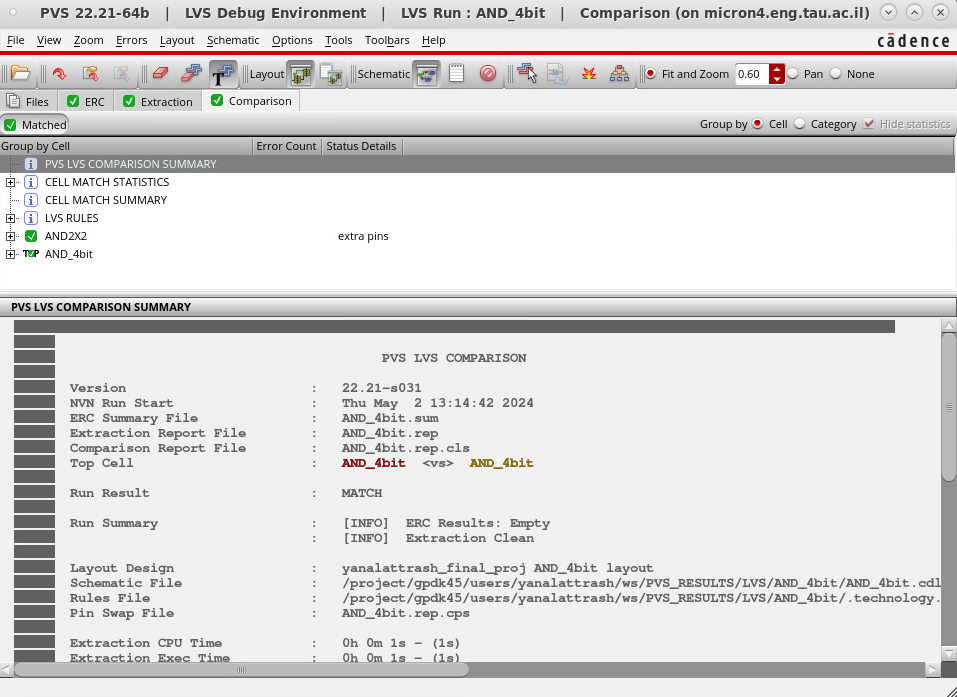
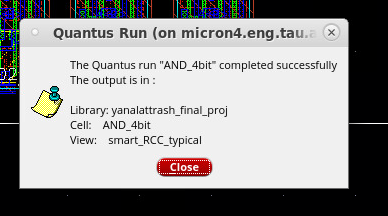
Description automatically generated

And\_4bit



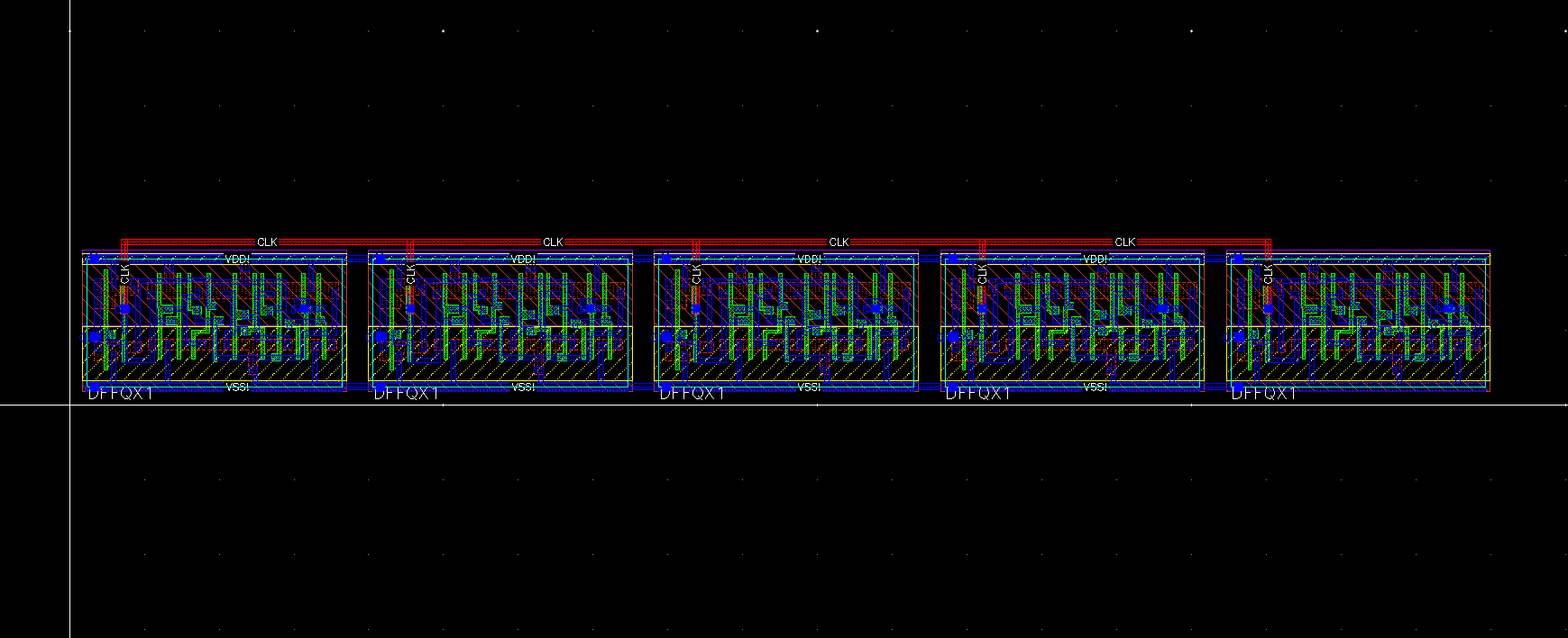
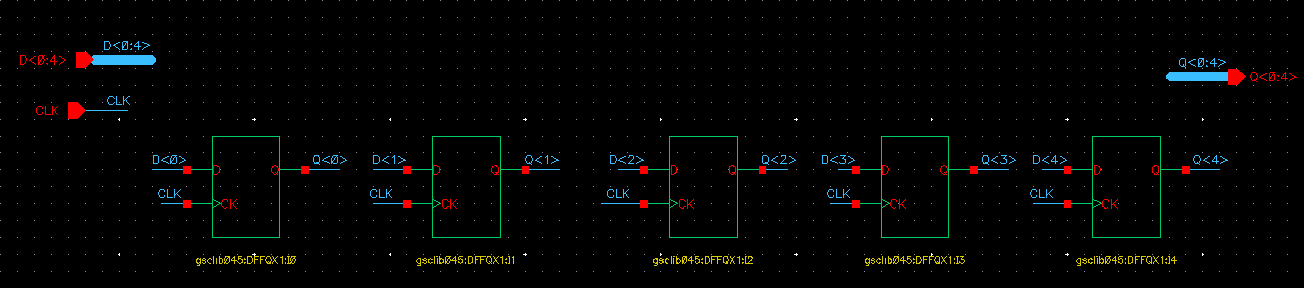
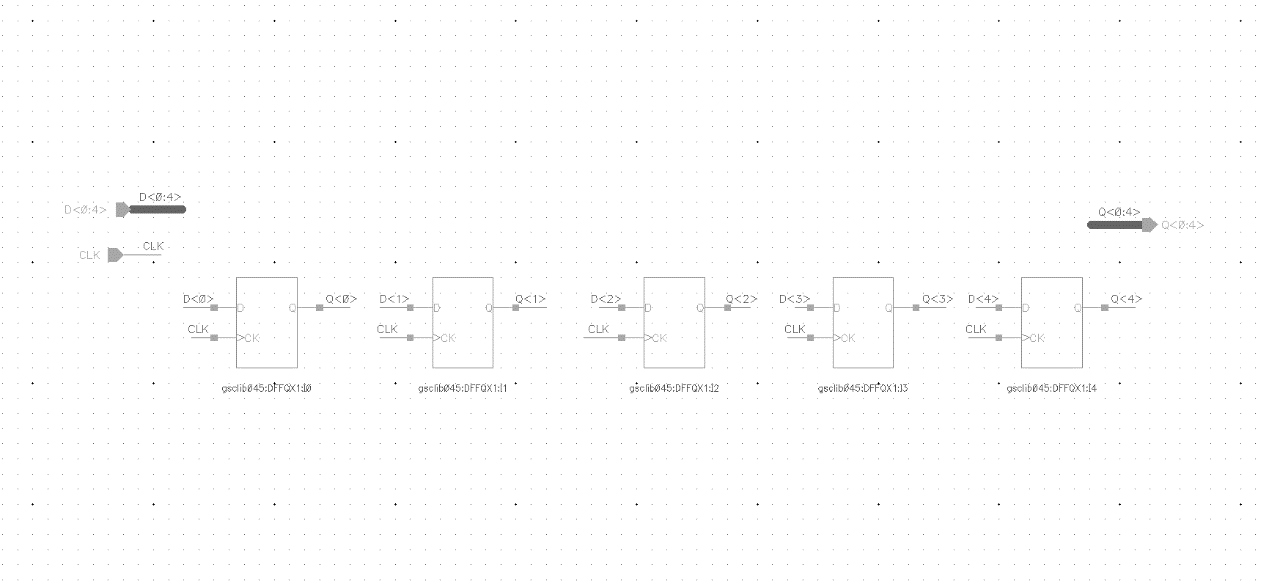
Size =(1.9)\*(5.38) =10.222

Checks:

A screenshot of a computer

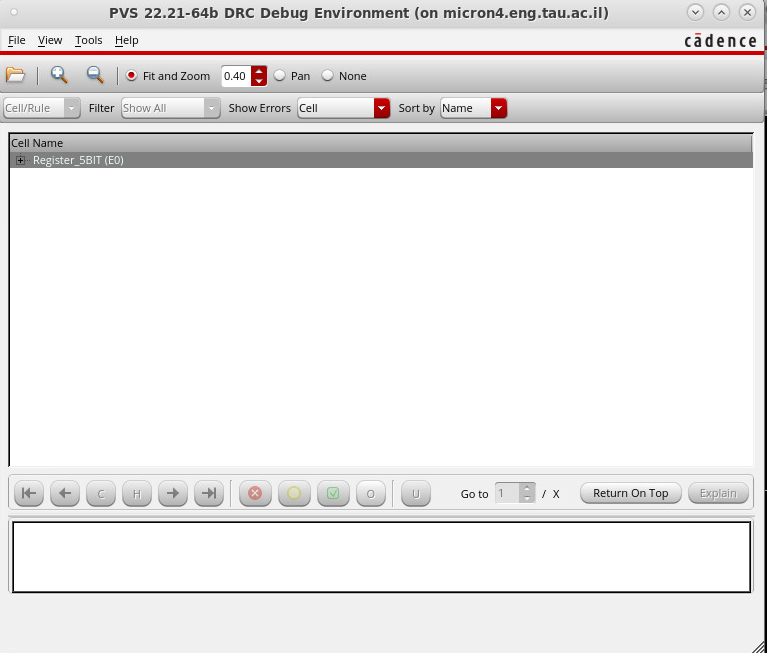
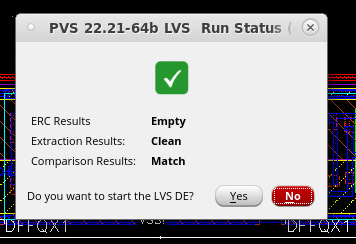
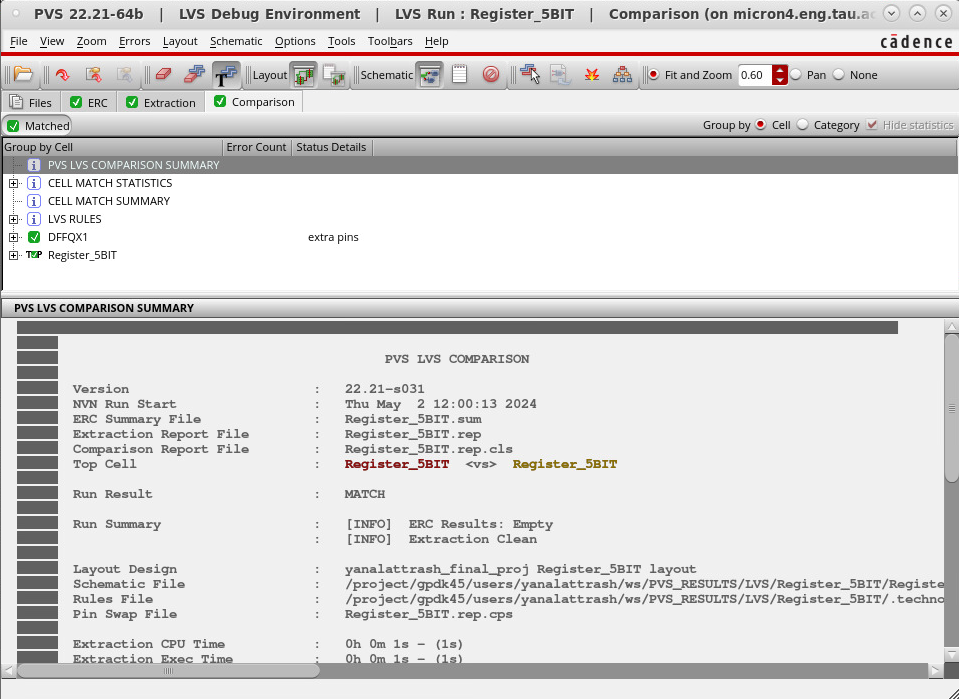
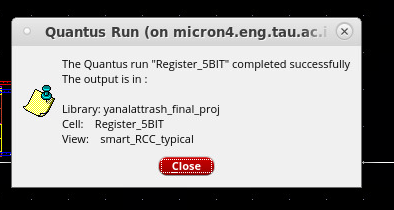
Description automatically generated

5bit-reg

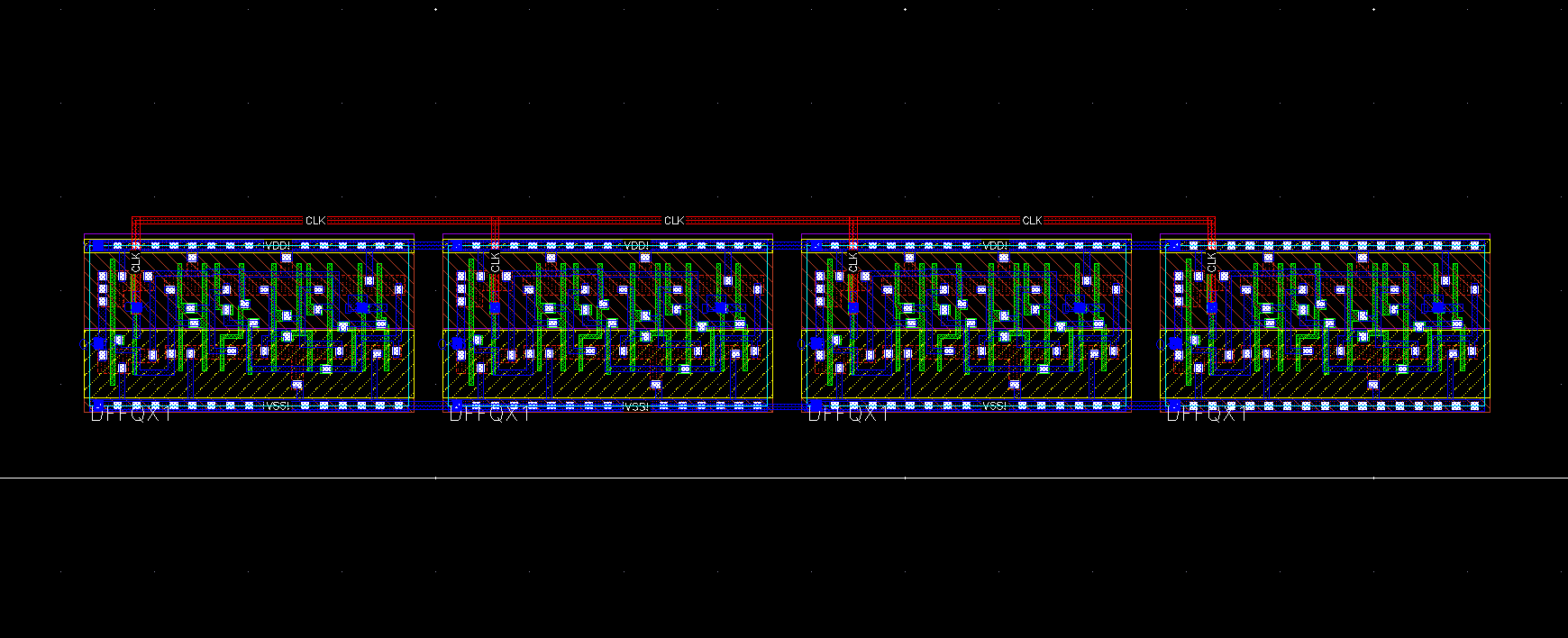
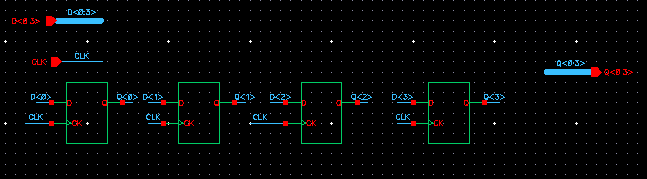
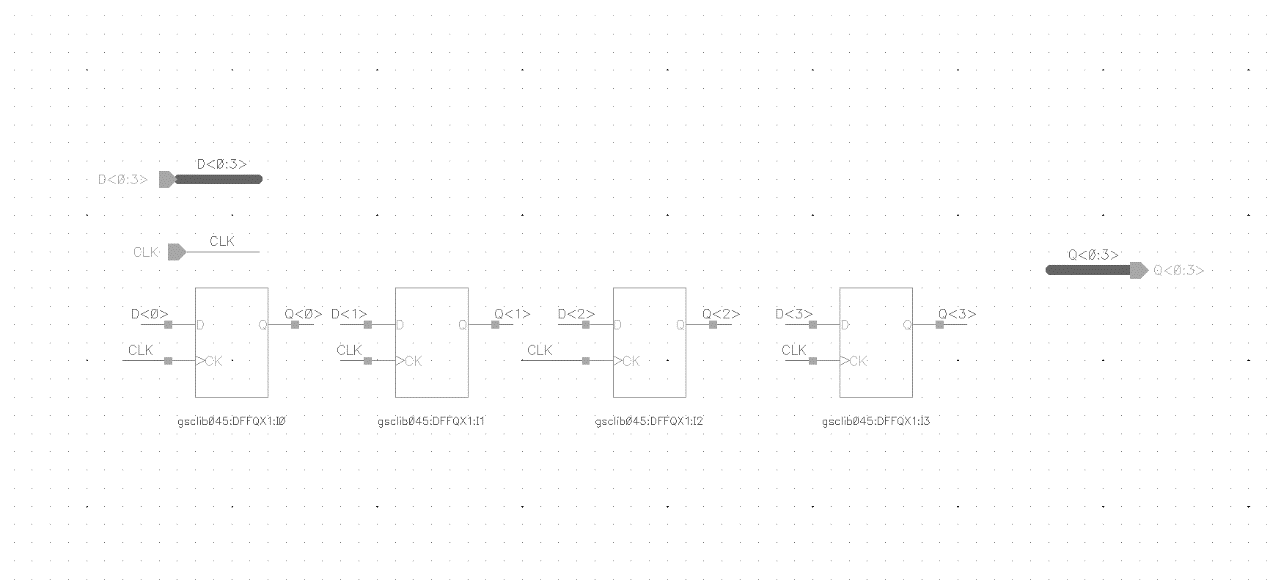


Size =(2.155)\*(18.805) = 40.53

Checks:

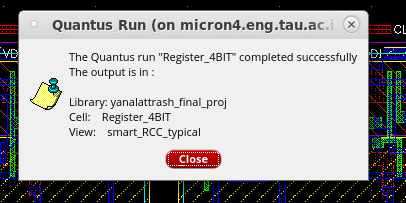


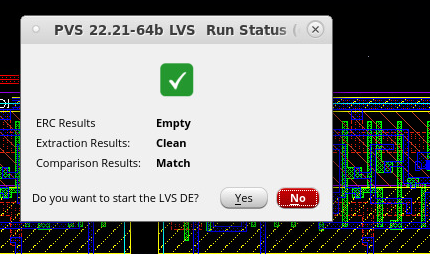
4-bit reg



Size= (1.9) \*(14.98) = 28.462

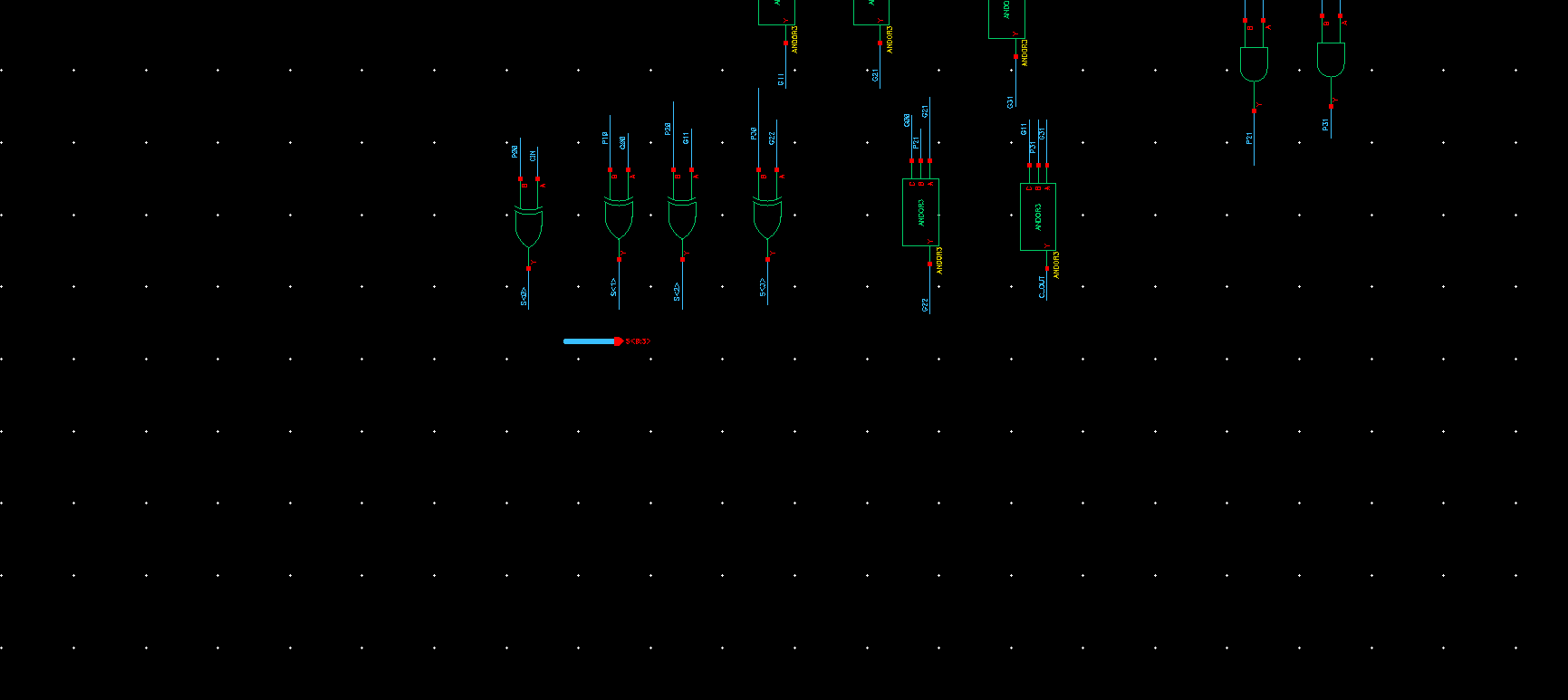
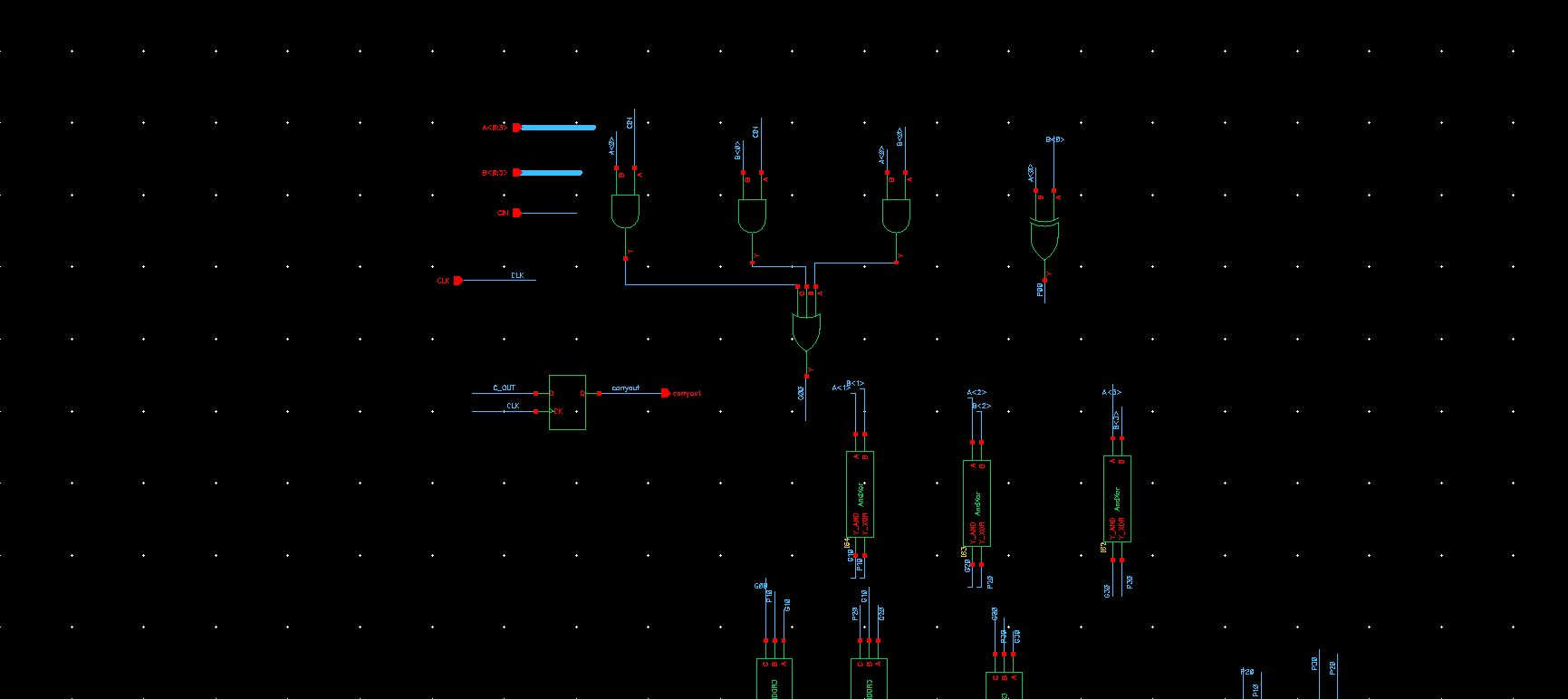
Checks:

A screenshot of a computer

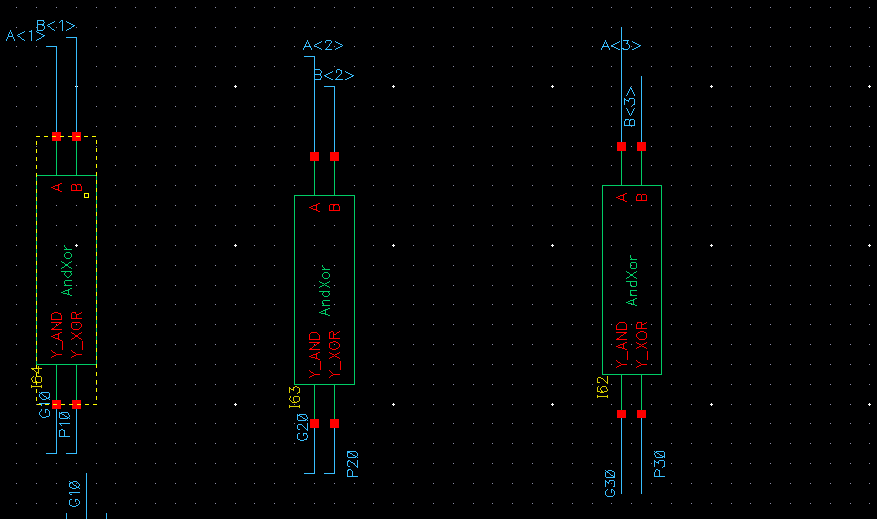
Description automatically generatedA screenshot of a computer

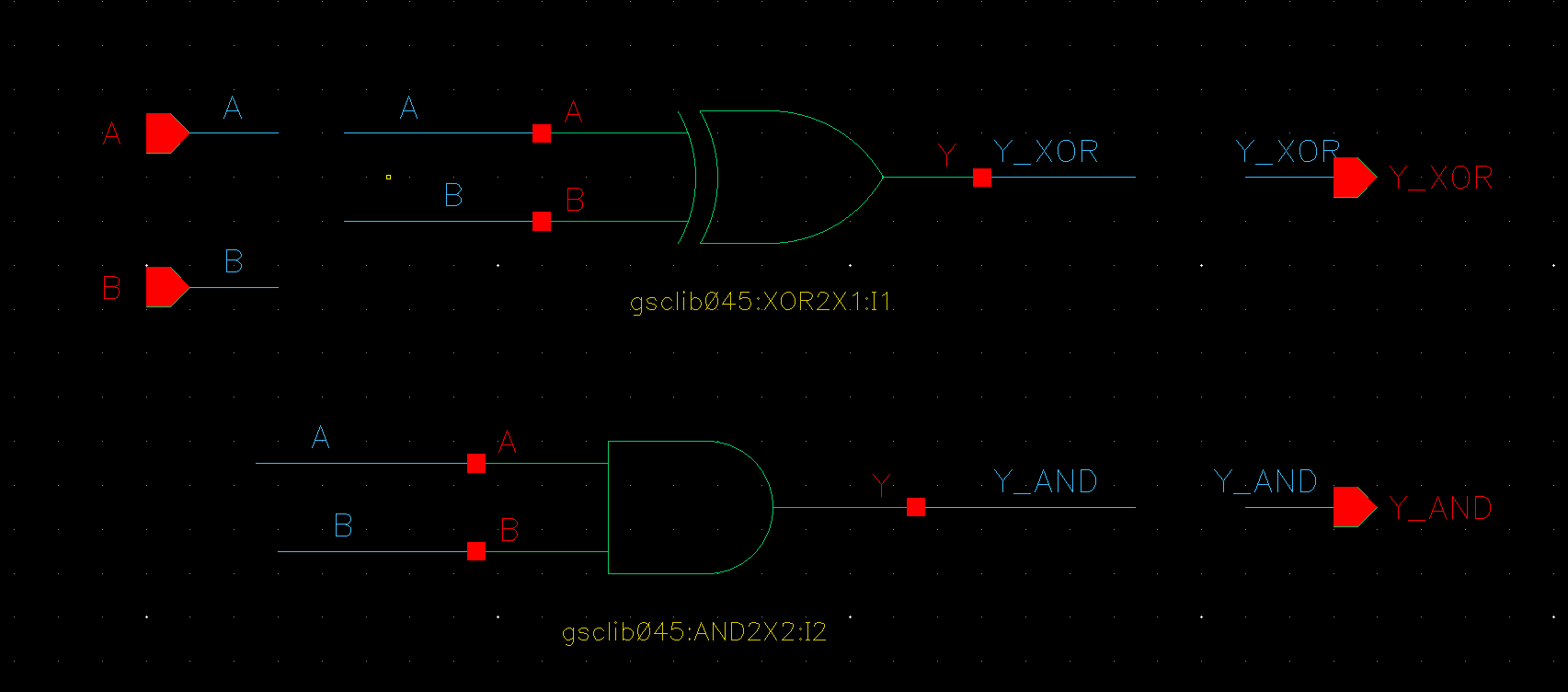
Description automatically generated

King stone Adder Schematic:

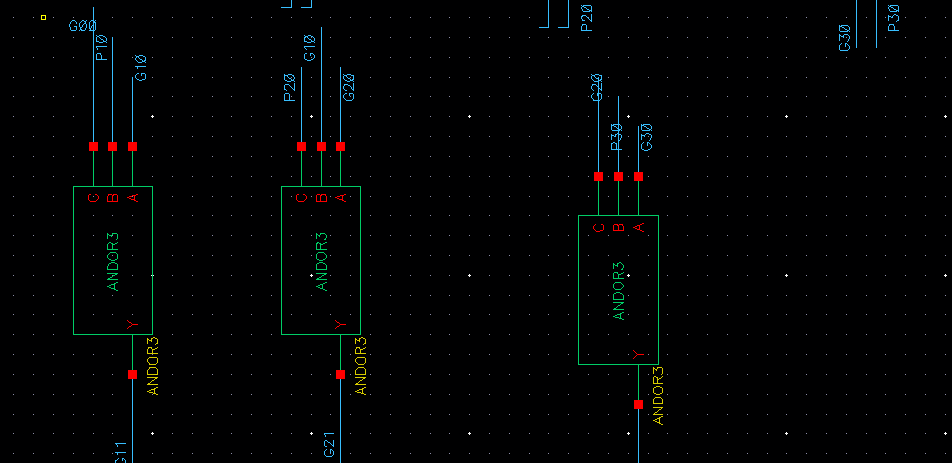


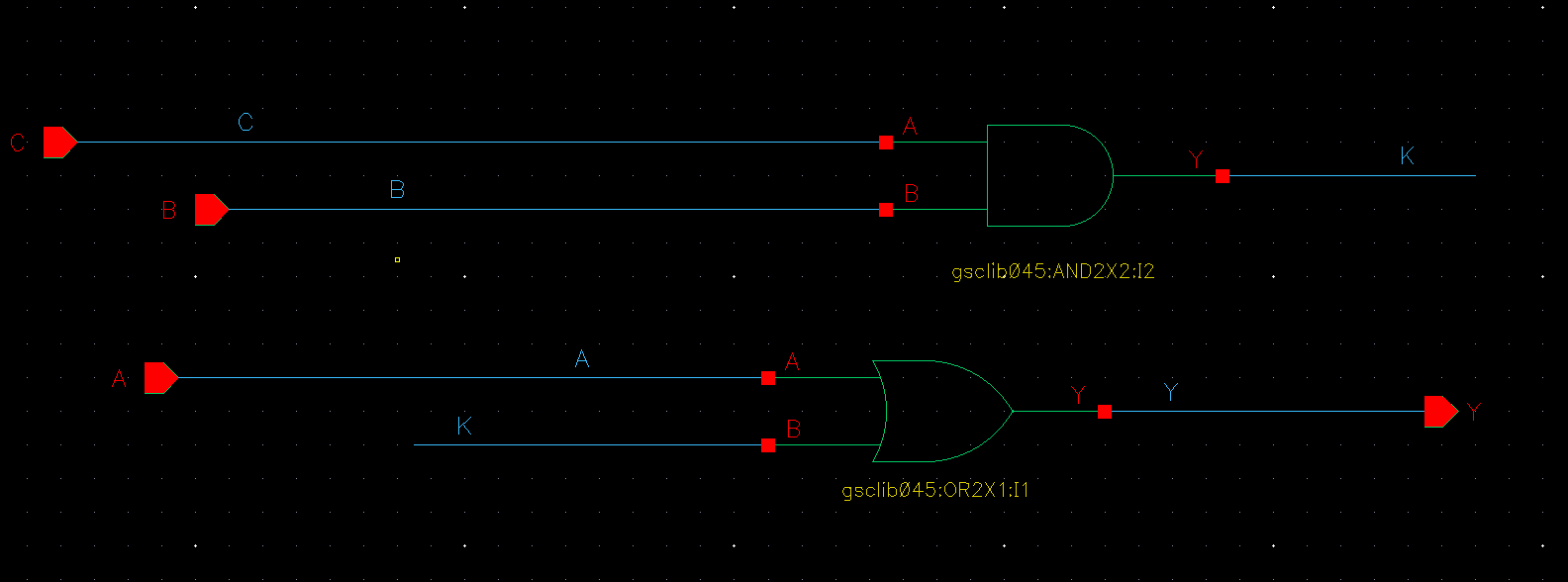
Add\_Xor block Schematic:

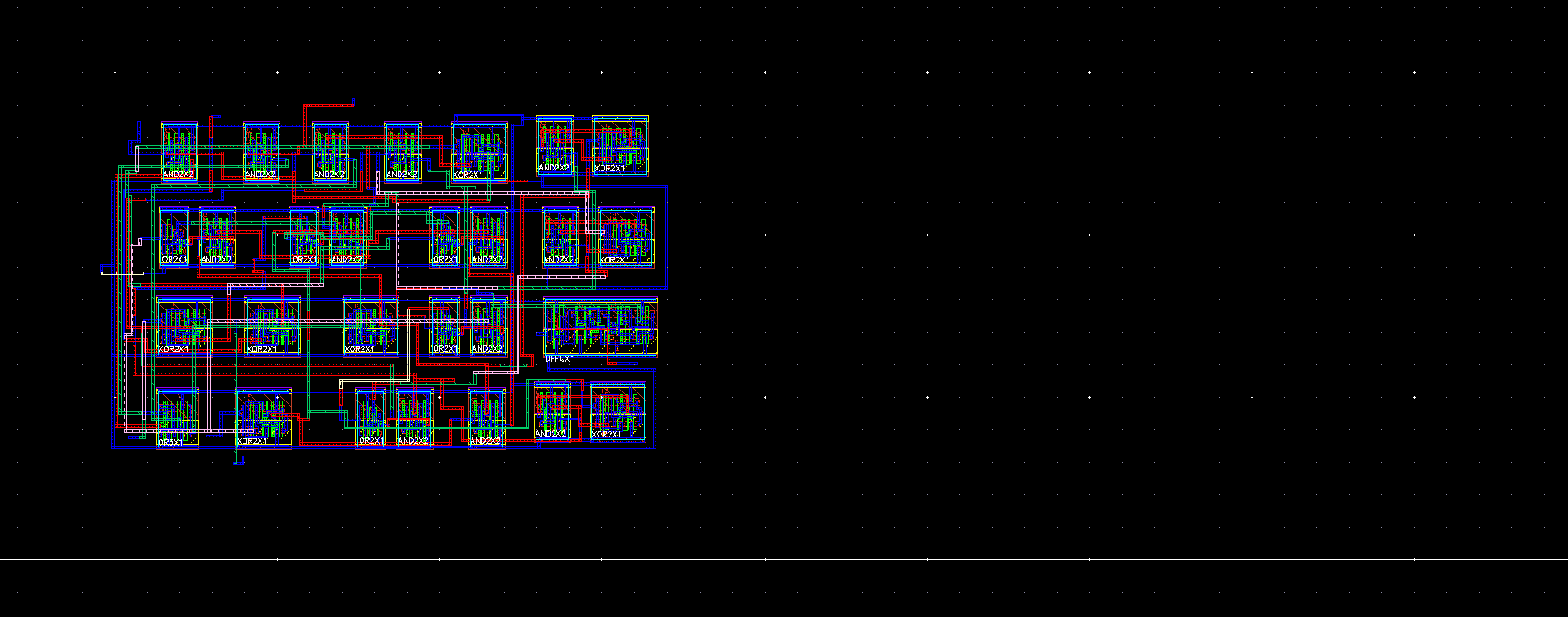




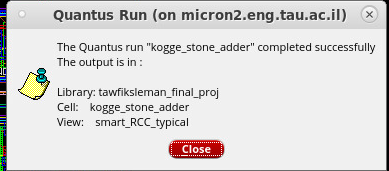
Andor3 gate Schematic:



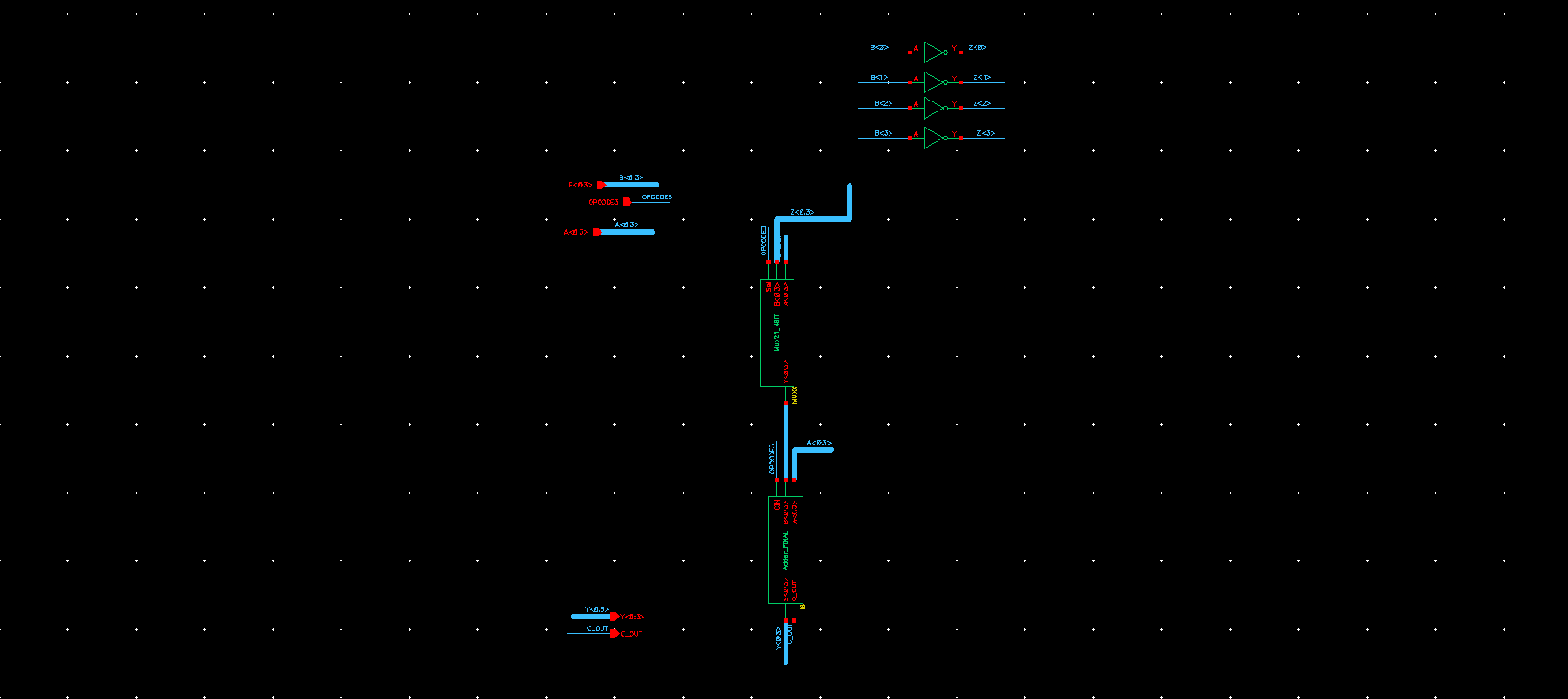


Kogge Stone adder layout :Size=(10.315)\*(17.12)=176.6

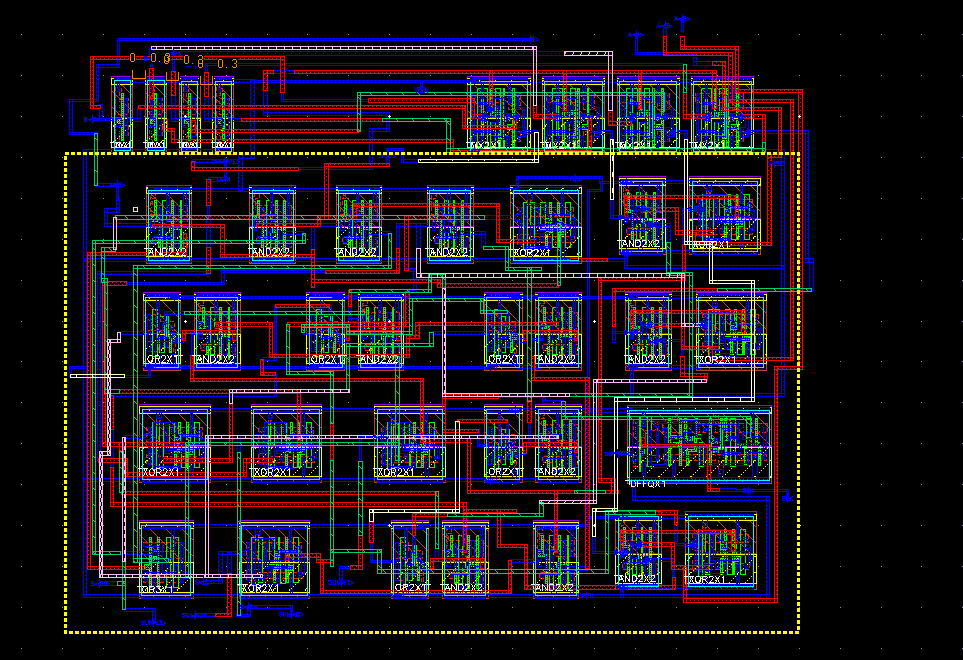
Check: 



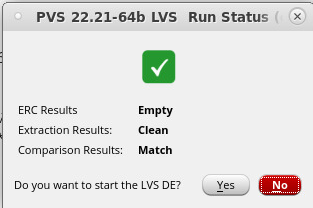
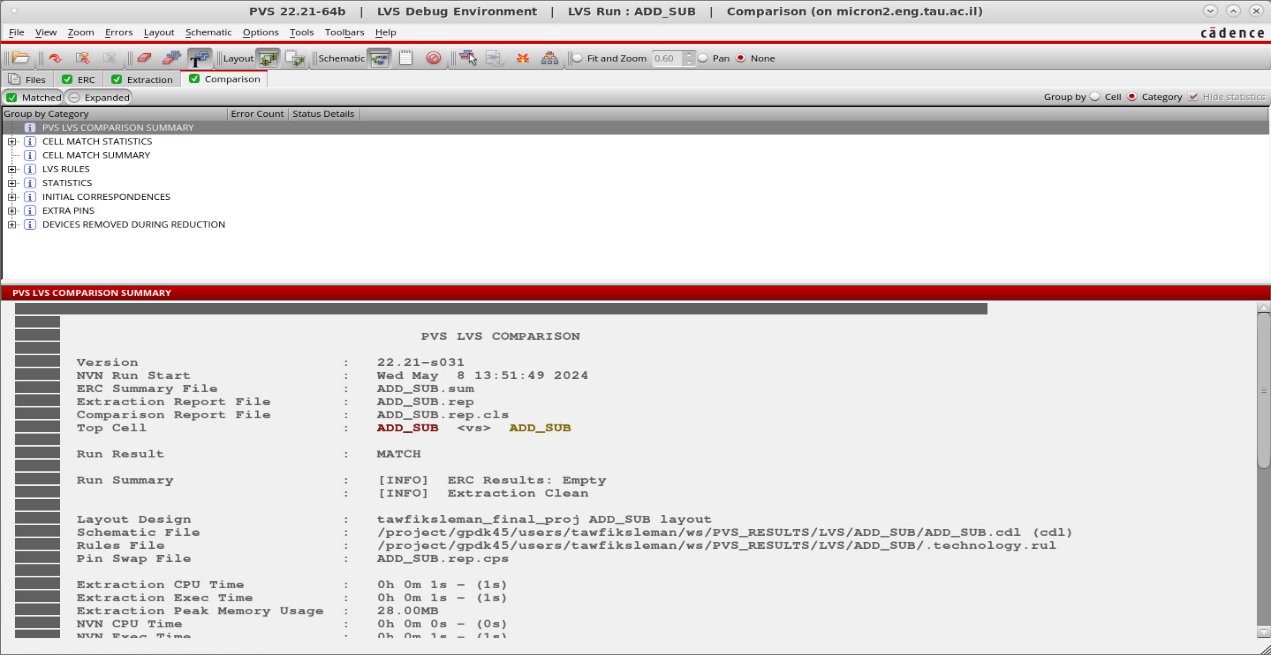
Adder-sub block schematic:

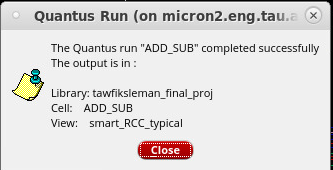


Adder-sub block layout:

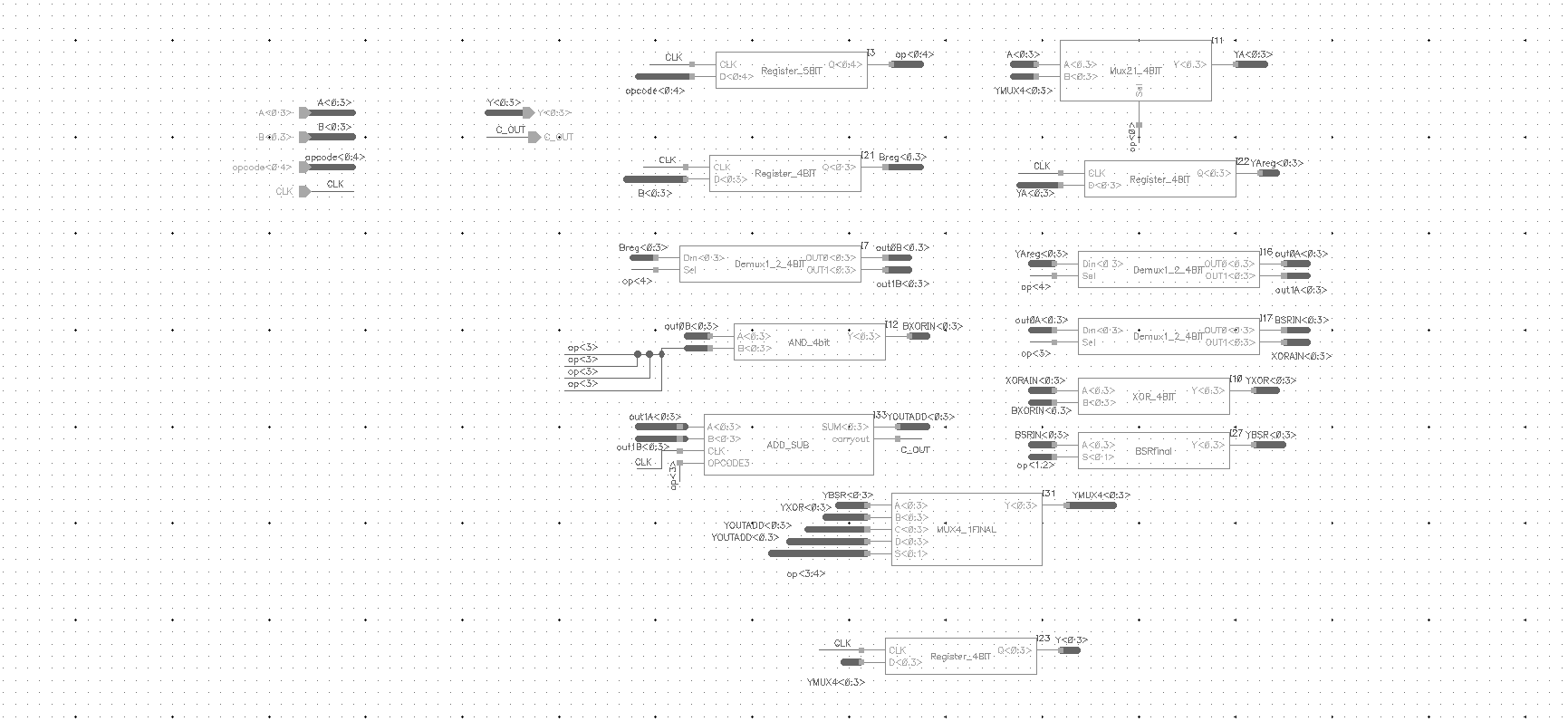


Size = (18.31)\*(15.04) =275.4

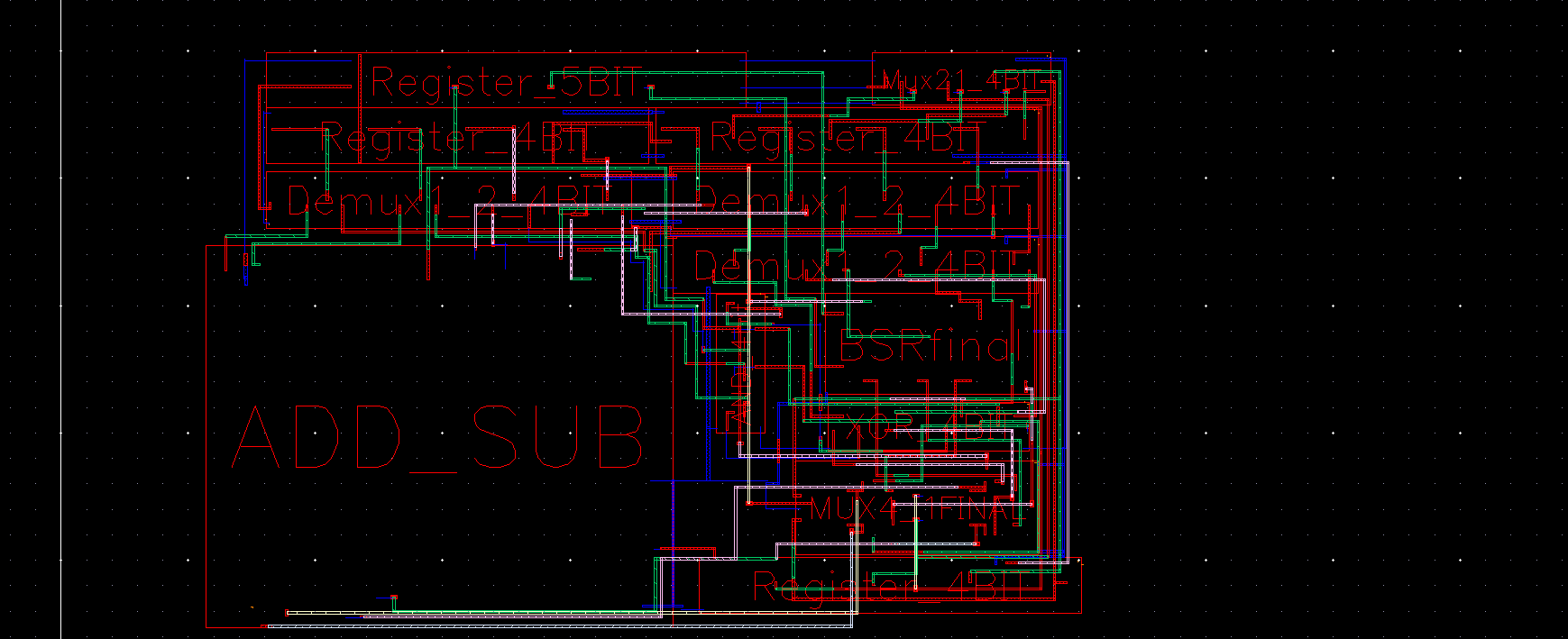
Check: 



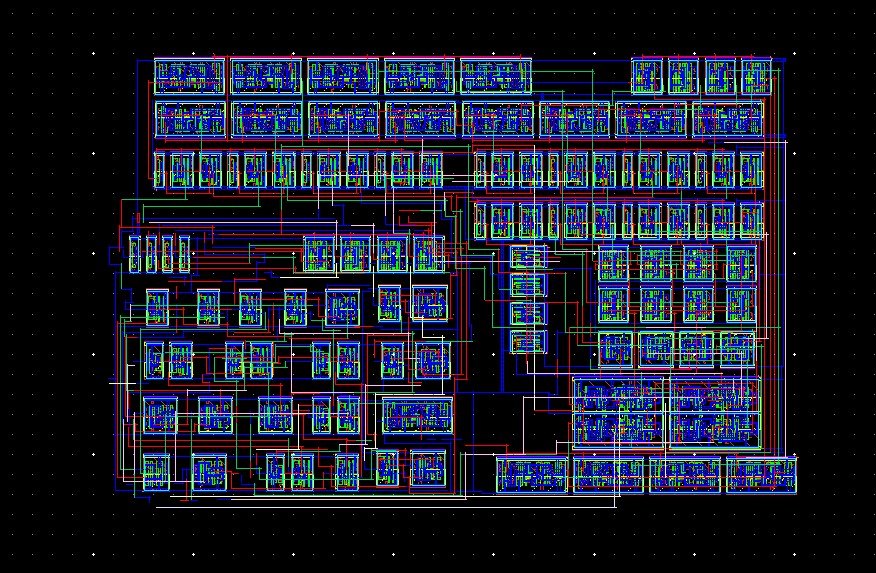
ALU Schematic:



ALU Floorplan:

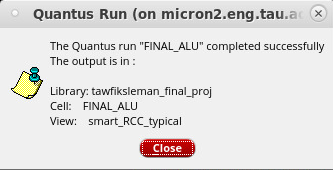
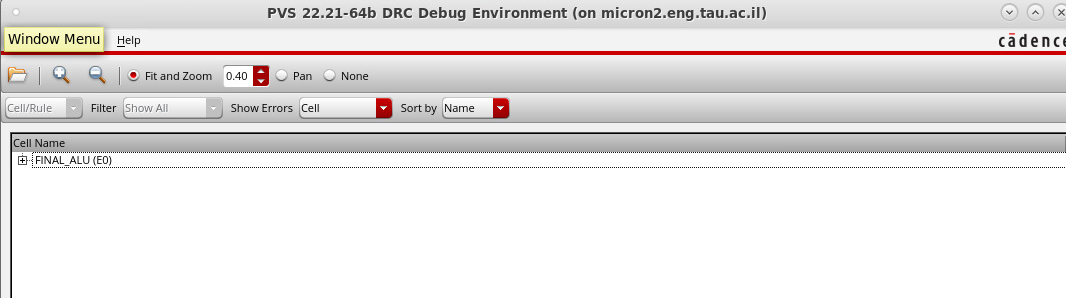
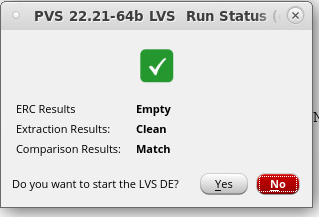
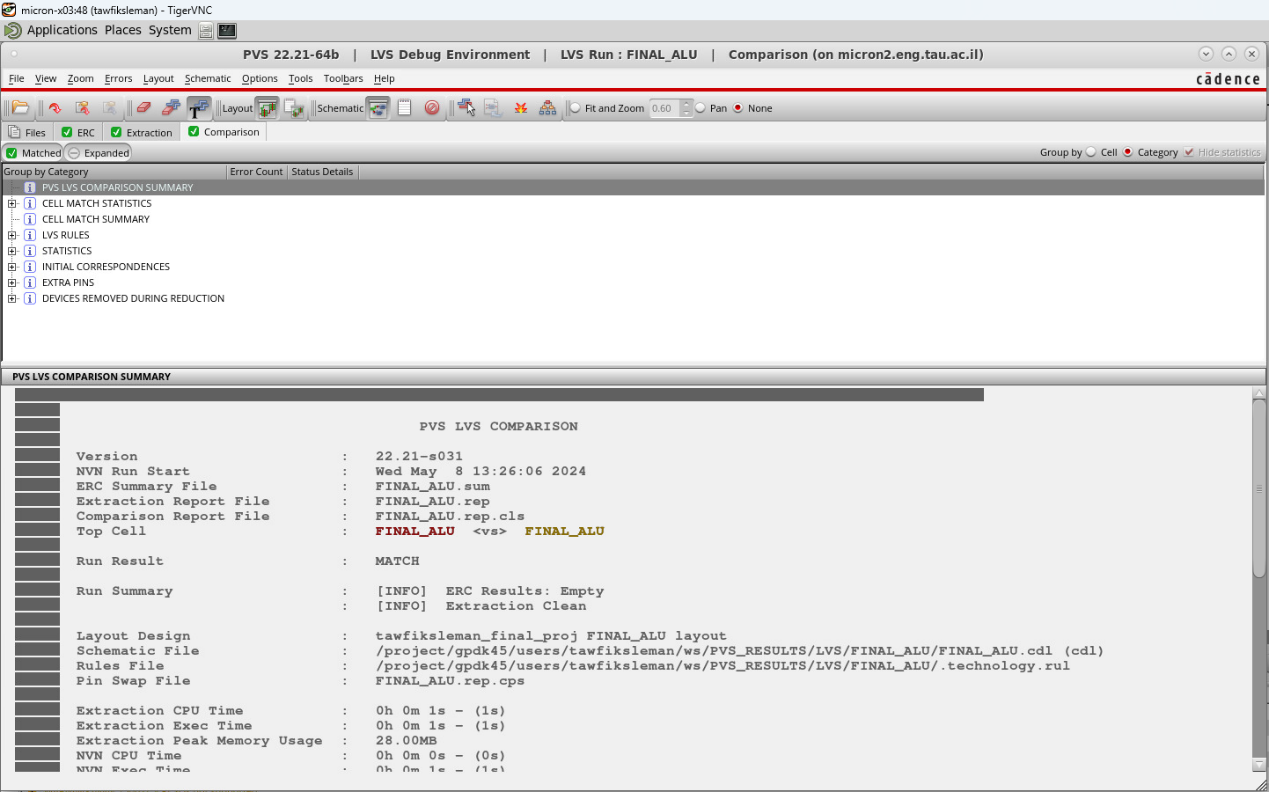


ALU layout:



Size= (33.555)\*(22.56) =757

ALU check:



Total cells size:

|  |  |
| --- | --- |
| cell size µm2 | cell |
| 40.53 | 5-bits register |
| 28.462 | 4-bits registerB |
| 32.3325 | Demux 1:2 (4bits) |
| 10.22 | And (4bits) |
| 34.74 | MUX 2:1(4bits) |
| 28.462 | 4 bit A reg |
| 32.3325 | Demux 1:2(4bits) |
| 32.3325 | Demux 1:2(4bits) |
| 14.82 | xor |
| 275.4 | Add-sub |
| 34.74 | MUX 4:1(4bits) |
| 35.614 | BSR |
| 28.462 | 4 bit Y reg |
| 628.46 | TOTAL SIZE |

we required that visualization area less than 1.5 of the cells total size. 1.5 of Total cells size = 1.5\*628.46=942.7, we see that its grater that ALU size which is 757.

**ALU Simulations:**

**Adder test:**

****

In the graph, we see that A=0111 in 2's complement (equivalent to 7 in decimal) and b=1000 in 2's complement (equivalent to -8 in decimal) was inputted into the ALU with opcode 10000, representing the "add" command. The resulting output is 1111, which translates to -1 in decimal. Upon adding 7 and -8, we observe a carry-out (cout) of 0, indicating the absence of overflow.

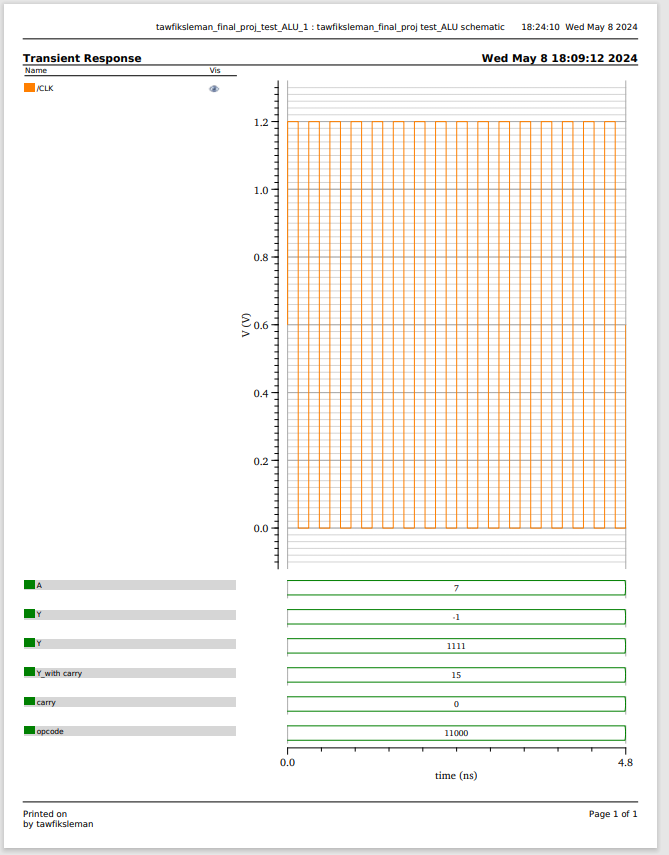
NOTICE that in this case we don’t have an overflow because XOR (CarryIN-4bit, Carryout-4bit)≠1

**Adding with feedback test:**

****

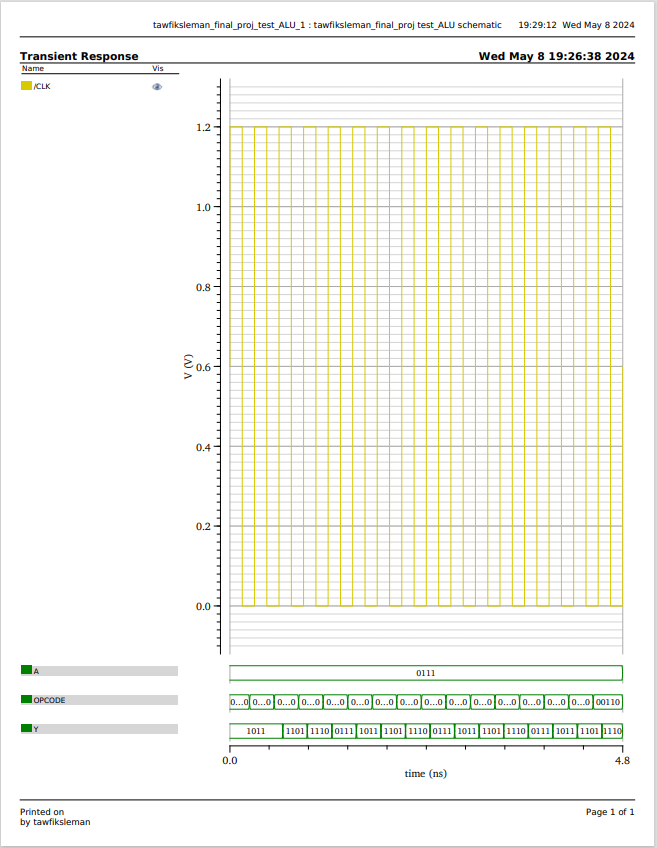
In the graph, we see that when Y=1111 in 2's complement (equivalent to -1 in decimal) and b=1000 in 2's complement (equivalent to -8 in decimal) was inputted into the ALU with opcode 10001, representing the "add with feedback" command. The resulting output is -1(in decimal) with overflow when mean -9 (in decimal) considering carry out, and when Y=0111 in 2's complement (equivalent to 7 in decimal) and b=1000 in 2's complement (equivalent to -8 in decimal) the output is -1 with no overflow.

**Subtract test:**

****

In the graph, we see that A=0111 in 2's complement (equivalent to 7 in decimal) and b=1000 in 2's complement (equivalent to -8 in decimal) was inputted into the ALU with opcode 11000, representing the "subtract" command. The resulting output is 01111 (considering carryout bit), which translates to 15 in decimal. Upon subtracting 7 and -8, we observe that XOR(CarryIN-4bit, Carryout-4bit) =1 indicating of overflow.

**BSR TEST:**

**** In the graph, we see that A=0111 in 2's complement (equivalent to 7 in decimal was inputted into the ALU with opcodes 00000 which mean 1 right shift, 00001 which mean 2 right shift, 00010 which mean 3 right shift, 00011 which mean 4 right shift, when opcode is 00000 the output is 1011 and for opcode 00001 the output is 1101 and for opcode 00010 the output is 1110 and for 00011 opcode, the output is 0111 which 7 In decimal as we expect.

**XOR TEST:**

****In the graph, we see that A=0111 in 2's complement (equivalent to 7 in decimal) and b=1000 in 2's complement (equivalent to -8 in decimal) was inputted into the ALU with opcode 01XX0, representing the "XOR" command. The resulting output is 1111 as we expected a bitwise between A and B