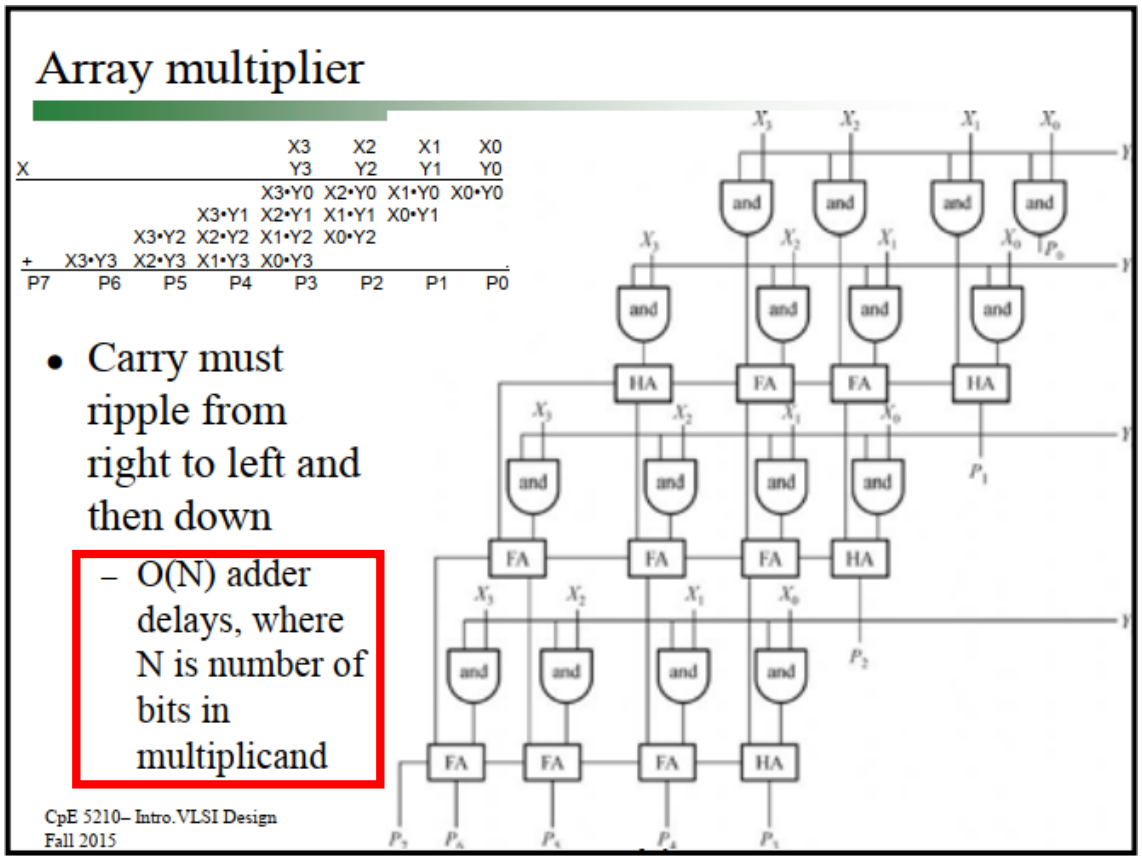
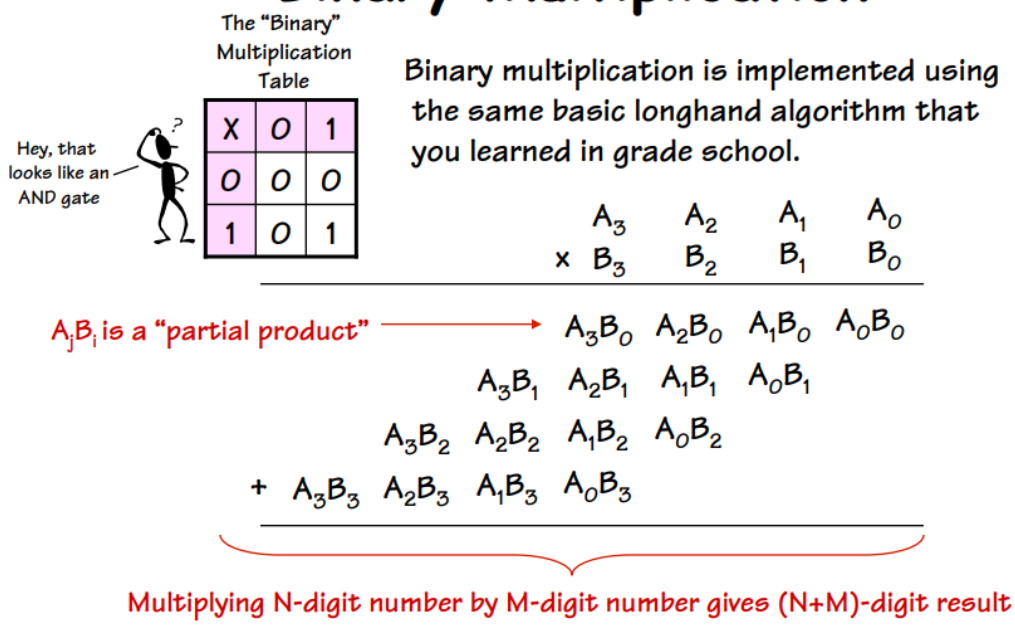
IVLSI 6 Bit Wallace Tree Multiplier

Contents:

- Background Theory (Wallace tree Multiplier delay compared to Array Multiplier, reduction of stages)
- Basic components: Schematic/layout(should be able to recognise comp/drv/lvs/graphs of each component)
- Complete circuit -schematic/layout/drc/lvs/waveform, specs:area,delay, total number of components
- Questions to be answered: how did you optimize area?
- Basic procedure: symbol/tb/layout link

Background Theory:





Easy part: forming partial products (just an AND gate since B_i is either 0 or 1)
Hard part: adding M, N-bit partial products

http://www.csbio.unc.edu/mcmillan/Comp411F15/Lecture12.pdf

Simple Combinational Multiplier

$t_{PD} = 10 * t_{PD}$

not 16

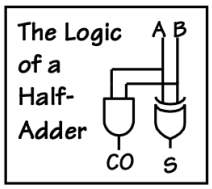
$t_{PD} = (2*(N-1) + N) * t_{PD}$

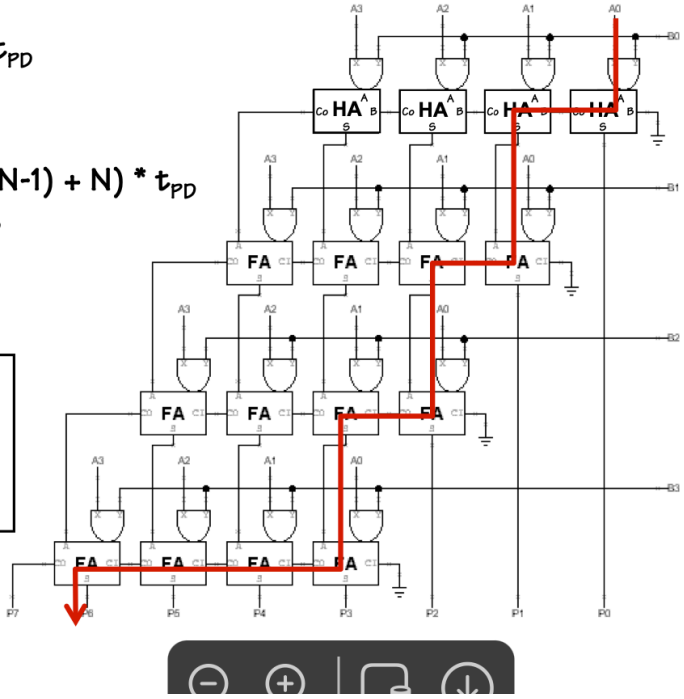
Components

N * HA

N(N-1) * FA

The Logic of a Half-Adder





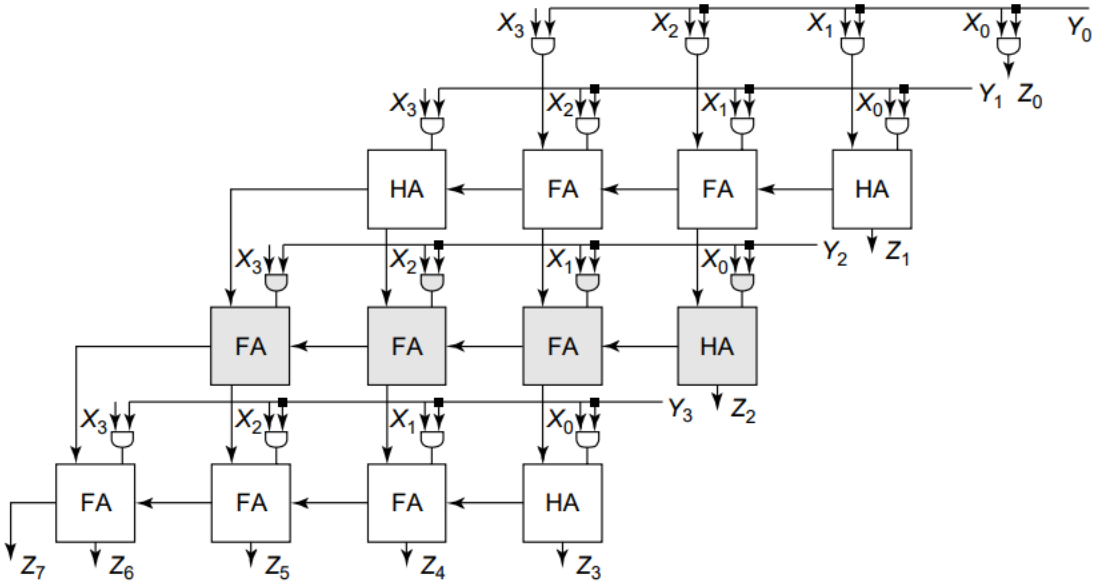
Is this faster than our assembly code?

To determine the timing specification of a composite combinational circuit we find the worst -case path for every output to any input.

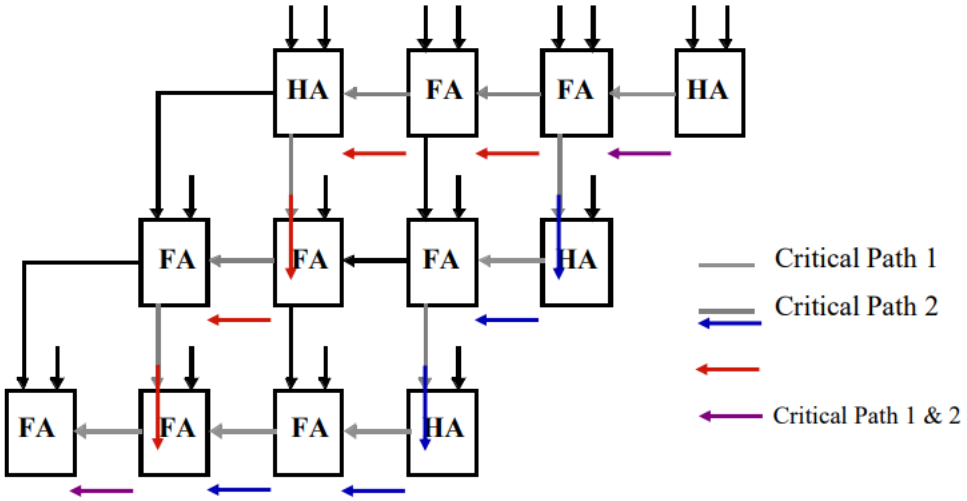
NB: this circuit only works for nonnegative operands

http://www.micro.deis.unibo.it/~baccaran/Rabaey/chapter11.pdf

The Array Multiplier



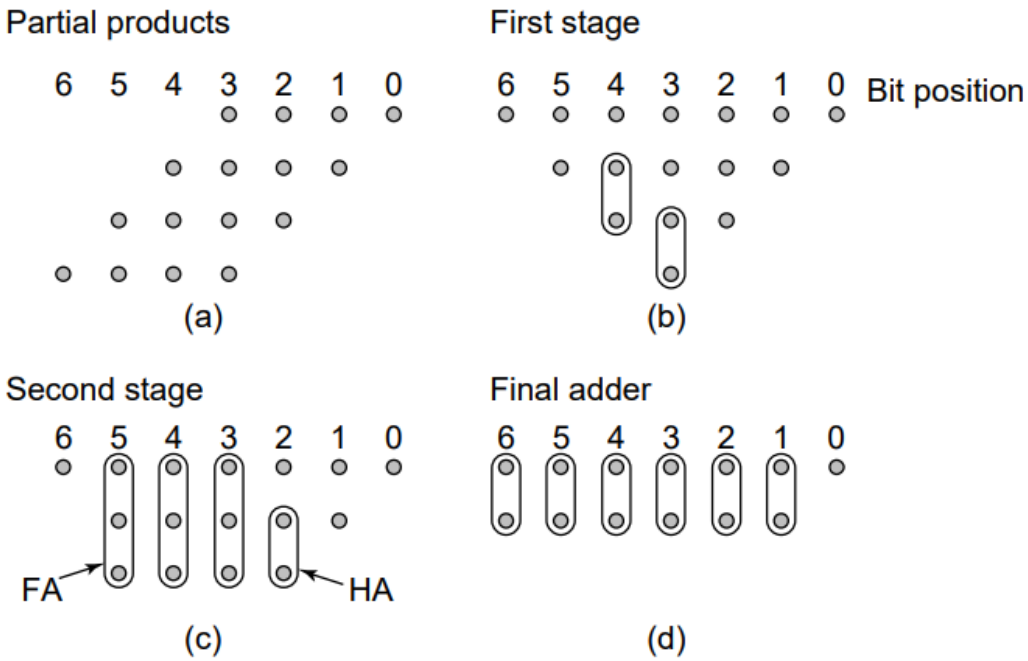
The MxN Array Multiplier — Critical Path



$$t_{mult} = [(M-1)+(N-2)]\,t_{carry} + (N-1)\,t_{sum} + t_{and}$$

Parallel addition of partial products does not take bits from previous coloumn unlike array multiplier.

Wallace-Tree Multiplier



In a conventional Wallace multiplier, the number of rows in subsequent stages can be calculated as:

$$r_{i+1} = 2[r_i/3] + r_i \bmod 3 \qquad (1)$$

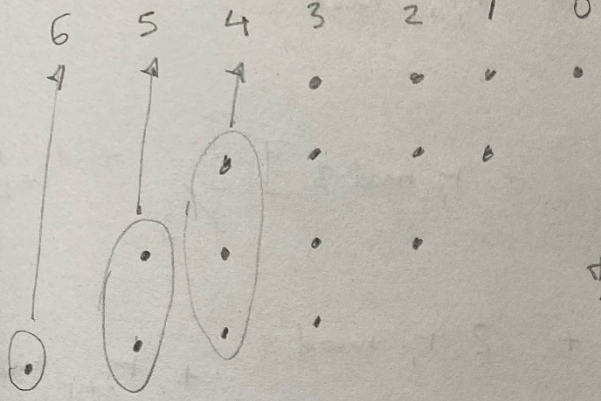
Where, r_i mod 3 denotes the smallest non-negative remainder of $r_i/3$. The number of rows in the subsequent

---

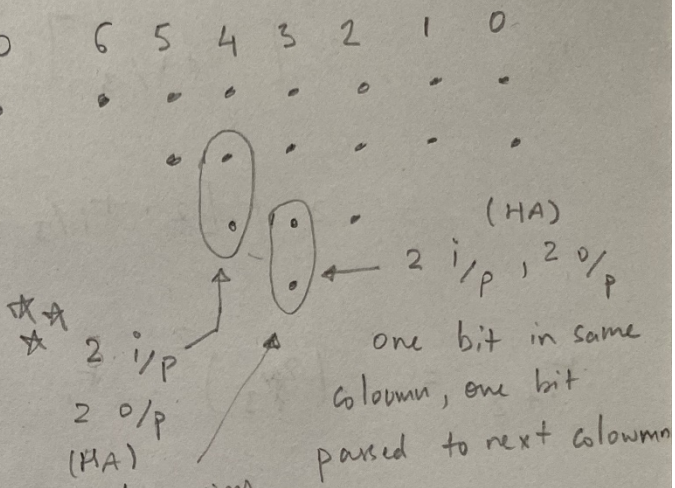
stages of a conventional 8-bit by 8-bit Wallace multiplier are calculated using (1), that are: r_0=N=9, r_1=6, r_2=4, r_3=3, r_4=2. Therefore, four stages are required to perform the reduction procedure, each with delay of one full adder.

Wallace-Tree Multiplier : [example 4×4]

Partial Products : "0" delay →

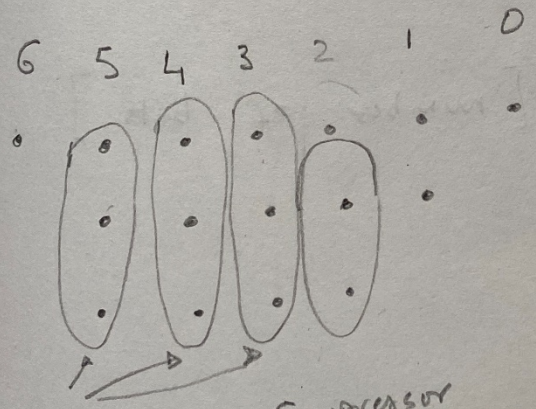


First stage



[Dont add previous o/p bit, one bit stays in column and the other bit passed]

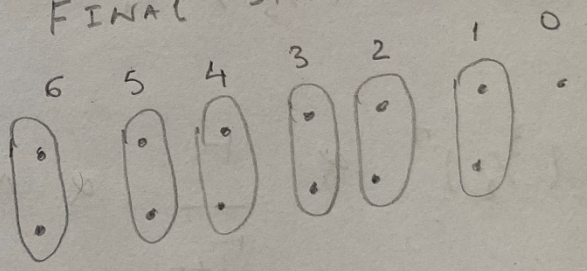
Second stage :

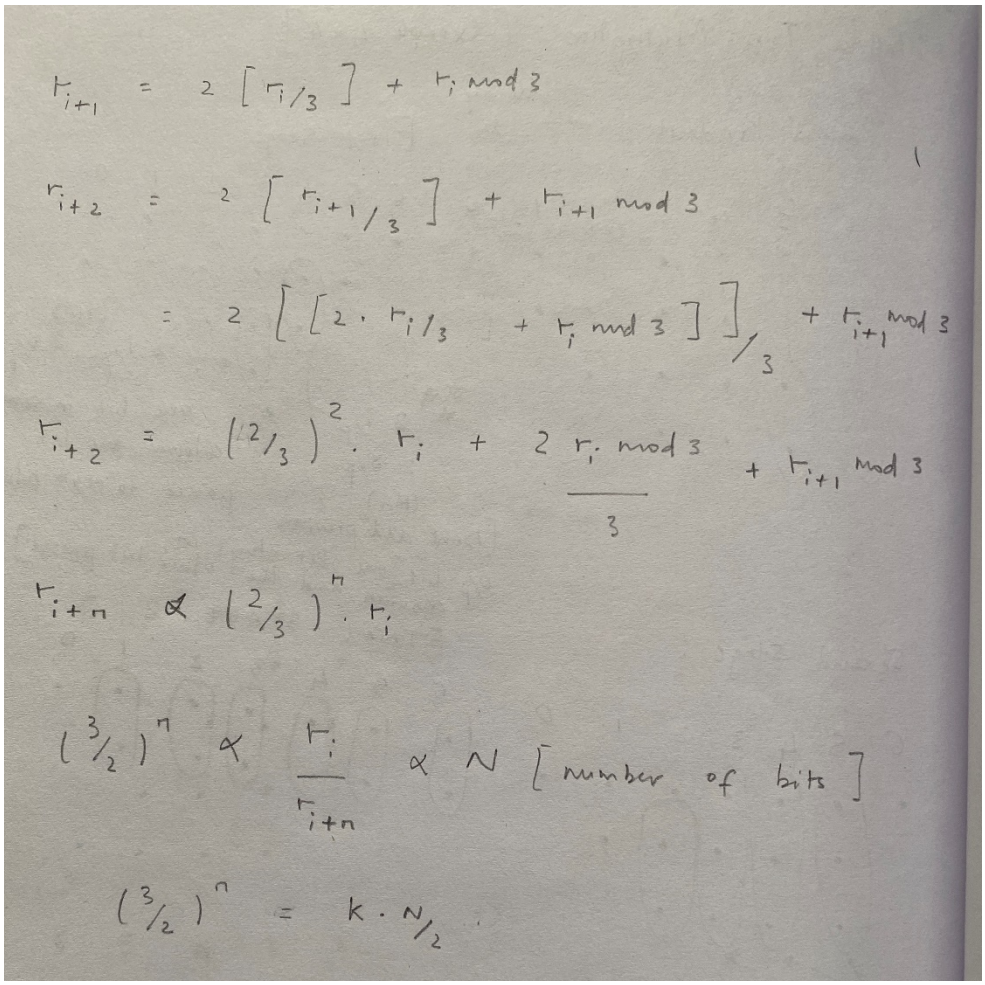


FA : 3 : 2 Compressor

3 i/p 2 /op

FINAL STAGE :





Wallace Tree Mult. Performance

Full adder is 3 to 2 compressor, N is number of bits

n is number of stages other than final stage

$$\left(\frac{3}{2}\right)^n = \frac{N}{2}$$

$$n \log_2\left(\frac{3}{2}\right) = \log_2\left(\frac{N}{2}\right)$$

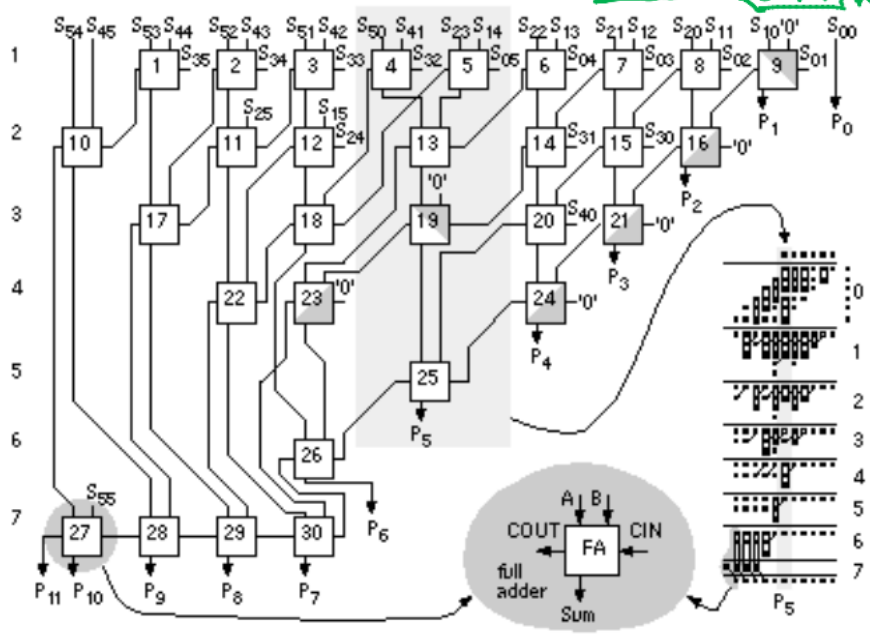
$$n = \frac{\log_2(N/2)}{\log_2(3/2)} = 1.71\left(\log_2 N - 1\right)$$

proportional

delay of final stage

$$t_{mult} = 1.71\left(\log_2 N - 1\right) t_{sum} + t_{adder}$$

A 6x6 Wallace Tree Multiplier



From http://www10.edacafe.com/book/ASIC/CH02/CH02.6.php

X_5 X_4 X_3 X_2 X_1 X_0

Y_5 Y_4 Y_3 Y_2 Y_1 Y_0

S_{50} S_{40} S_{30} S_{20} S_{10} S_{00}

S_{51} S_{41} S_{31} S_{21} S_{11} S_{01}

S_{52} S_{42} S_{32} S_{22} S_{12} S_{02}

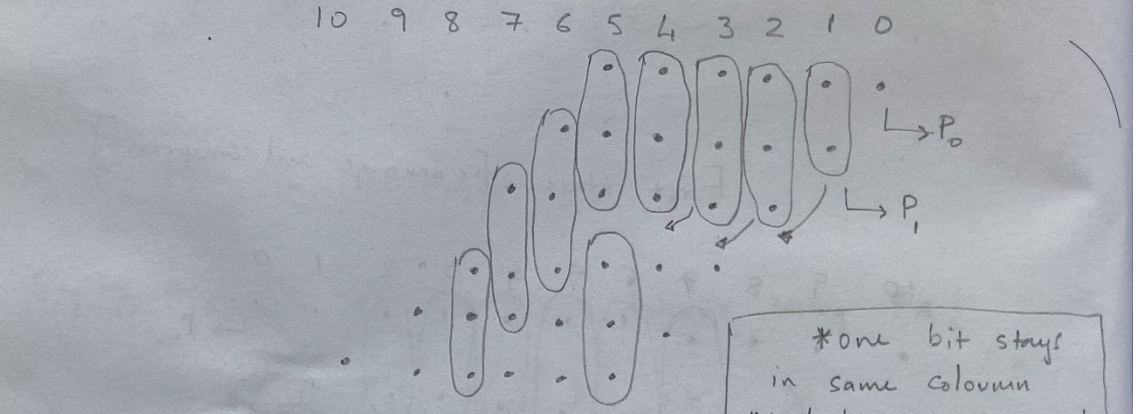
S_{53} S_{43} S_{33} S_{23} S_{13} S_{03}

S_{54} S_{44} S_{34} S_{24} S_{14} S_{04}

S_{55} S_{45} S_{35} S_{25} S_{15} S_{05}

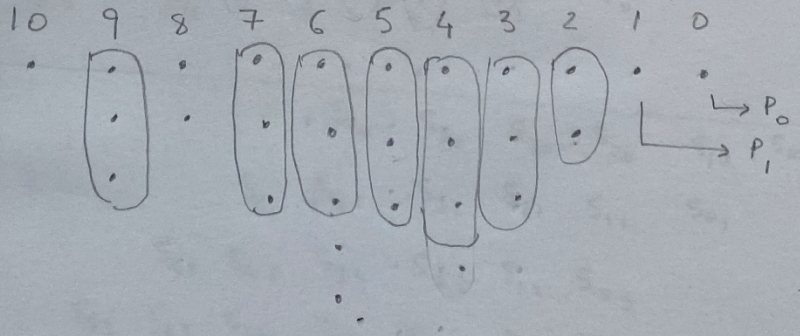
How to group:

0^{th} stage

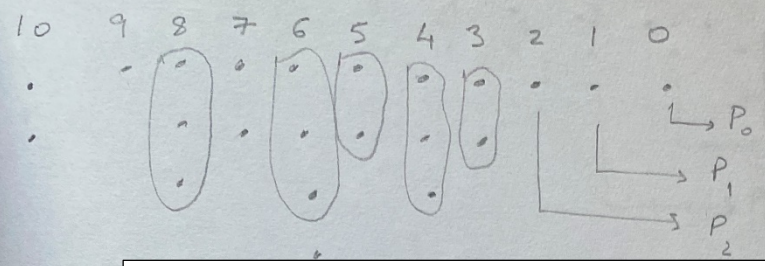


*one bit stays in same column next bit goes to next column

1^{st} stage

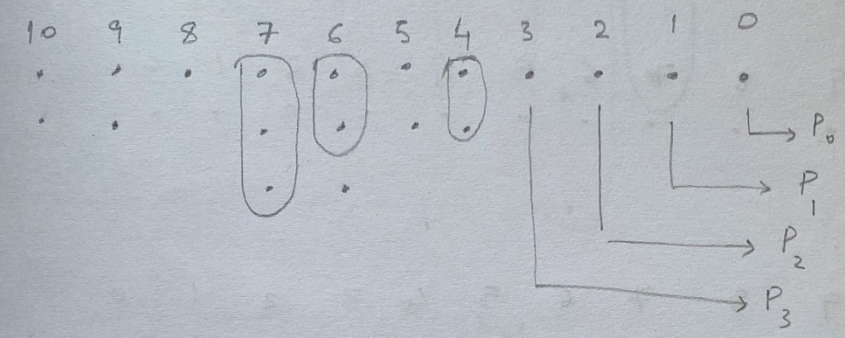


2nd stage

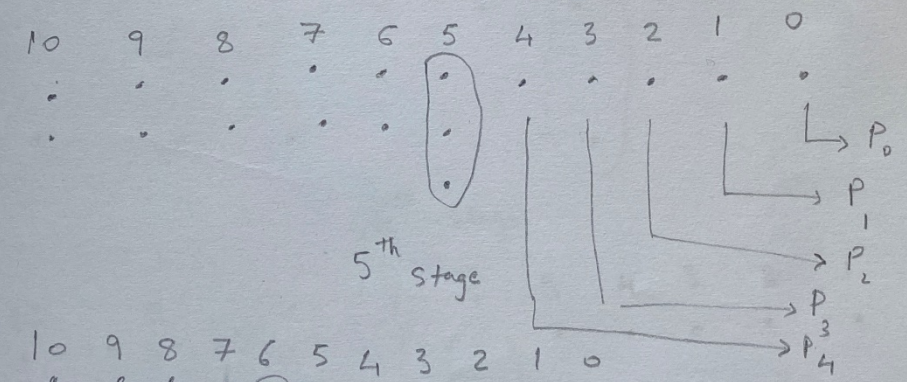


Direct Implementation of schematic shown in slide

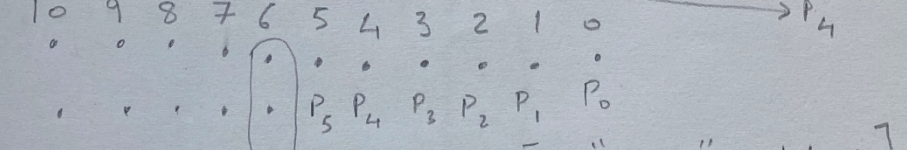
3rd stage



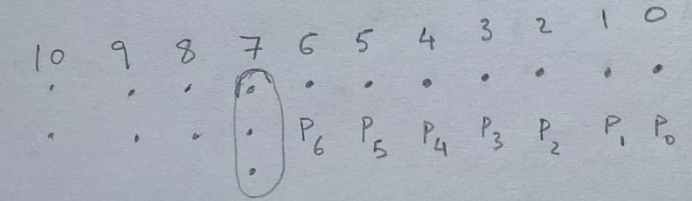
4th stage

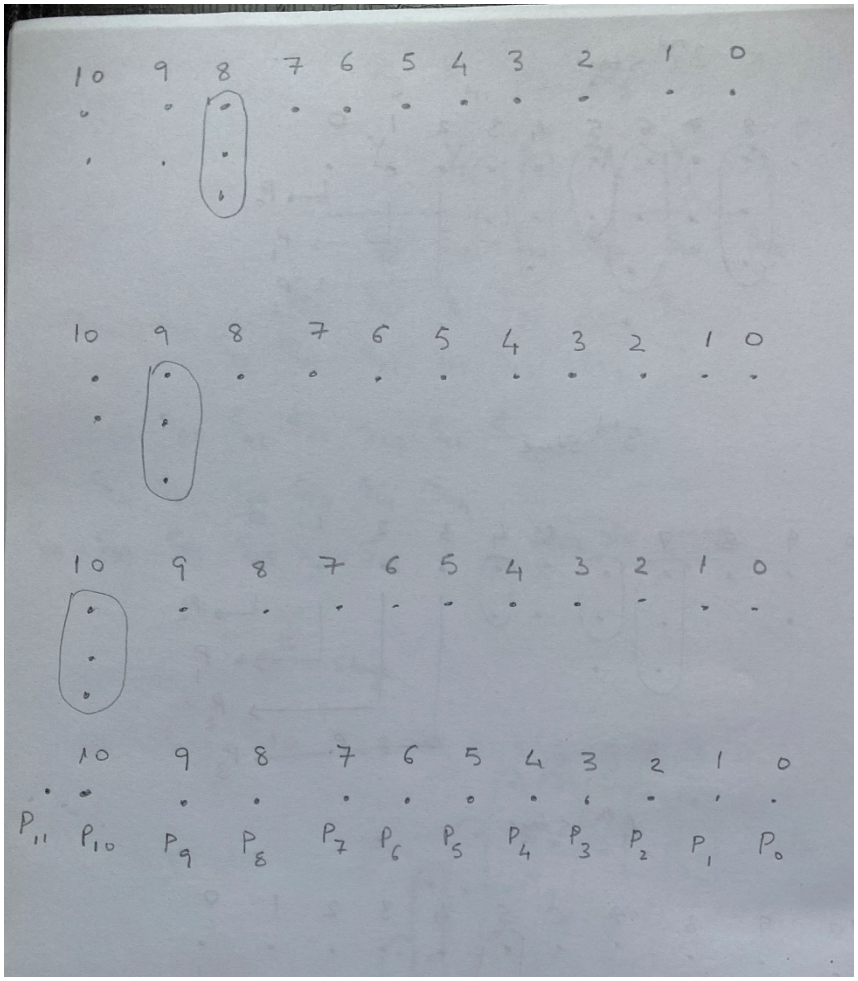


5th stage



6th Stage [4 "ripple" adders]

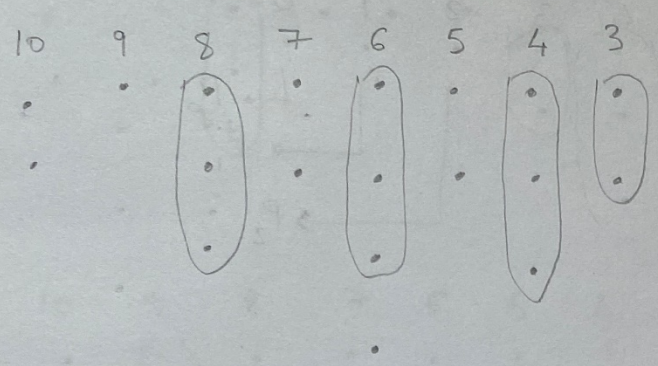




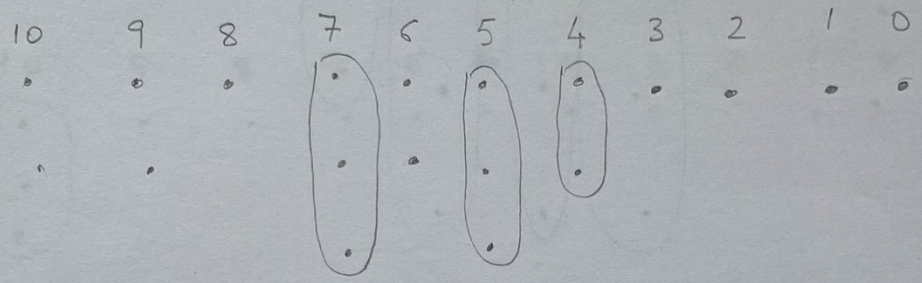
Procedure for reduction of stages:

- Whenever there are three bits in a coloumn use FA as 3:2 compressor
- Use HA as 2:2 compressor only when it will produce a product bit
- The picture on next page shows reduction using this method, it is different from textbook method only from stage 2 to final ripple carry stage

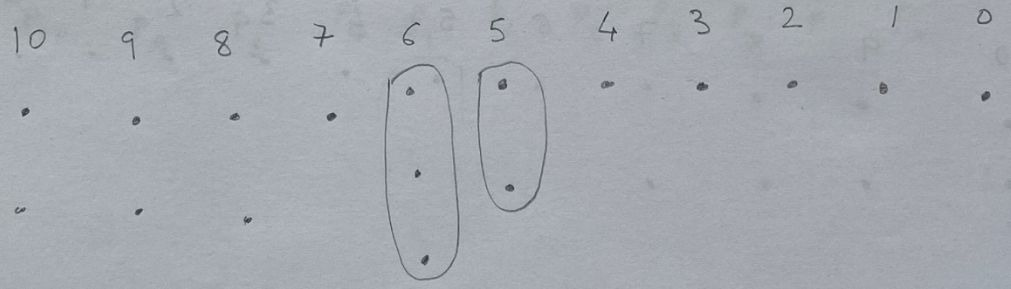
2nd stage



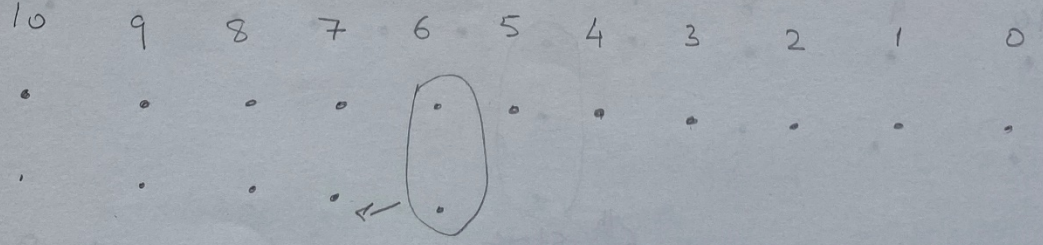
3rd stage



4th stage



5th stage

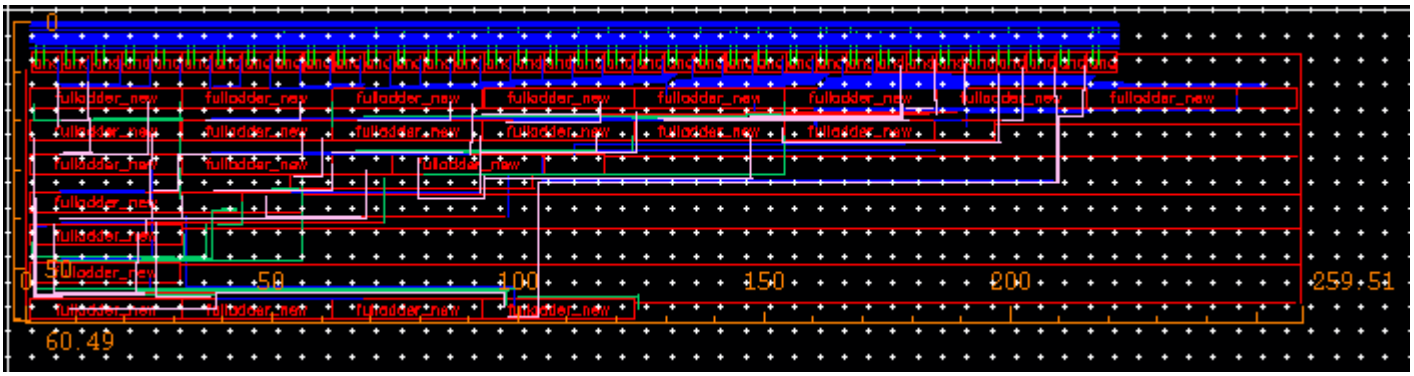


Ripple carry adder stage

Complete circuit

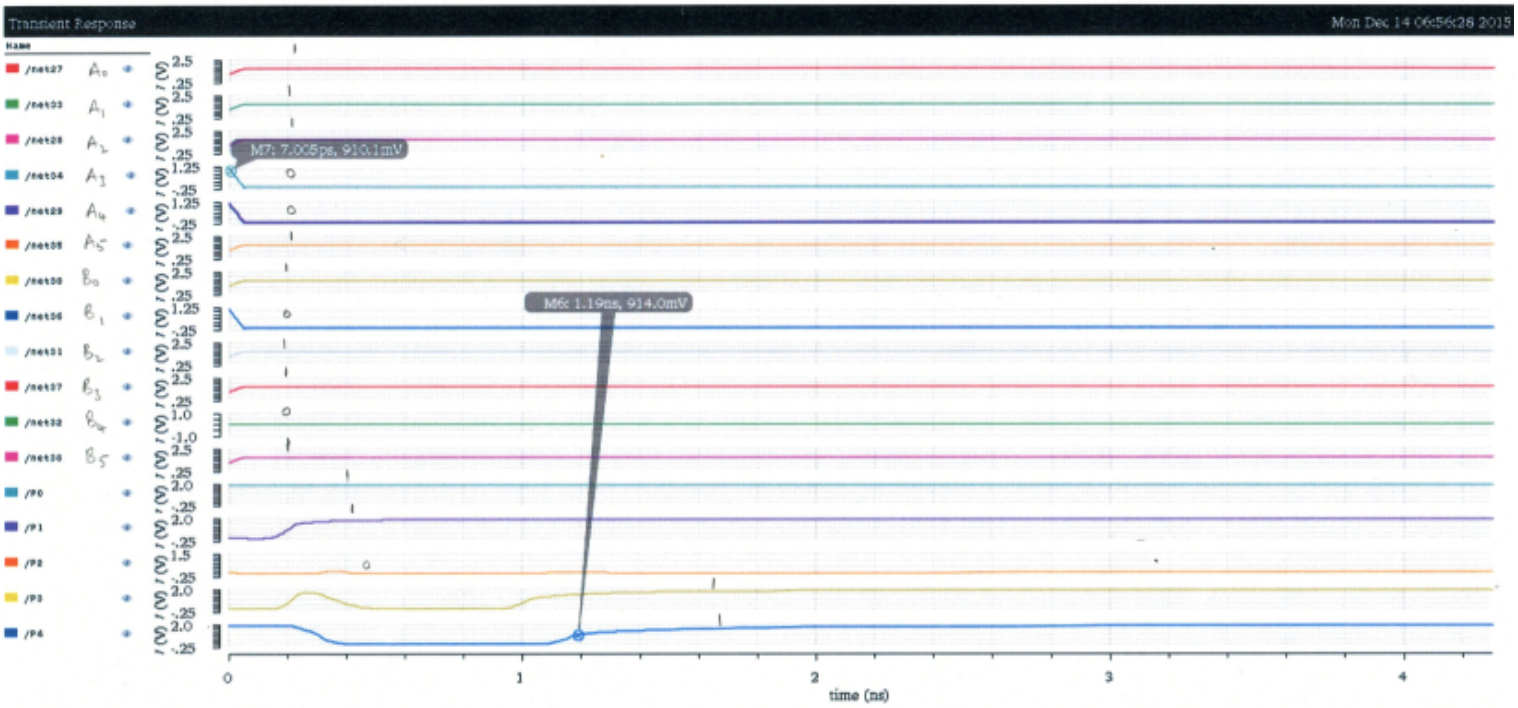
- Vdd=1.8V
- Area= 60.49um *259.51um=15723.71 um^2 =0.015725 mm^2
- How did you measure delay ? Testbench used?
- Delay for the sequence (101101 x 100111= 0110 1101 1011) = 1.39 ns

Measuring Area:



	B_5	B_4	B_3	B_2	B_1	B_0	×	A_5	A_4	A_3	A_2	A_1	A_0	=										
INPUTS:	1	0	1	1	0	1		1	0	0	1	1	1		0	1	1	0	1	1	0	1	1	0	1	1
															P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

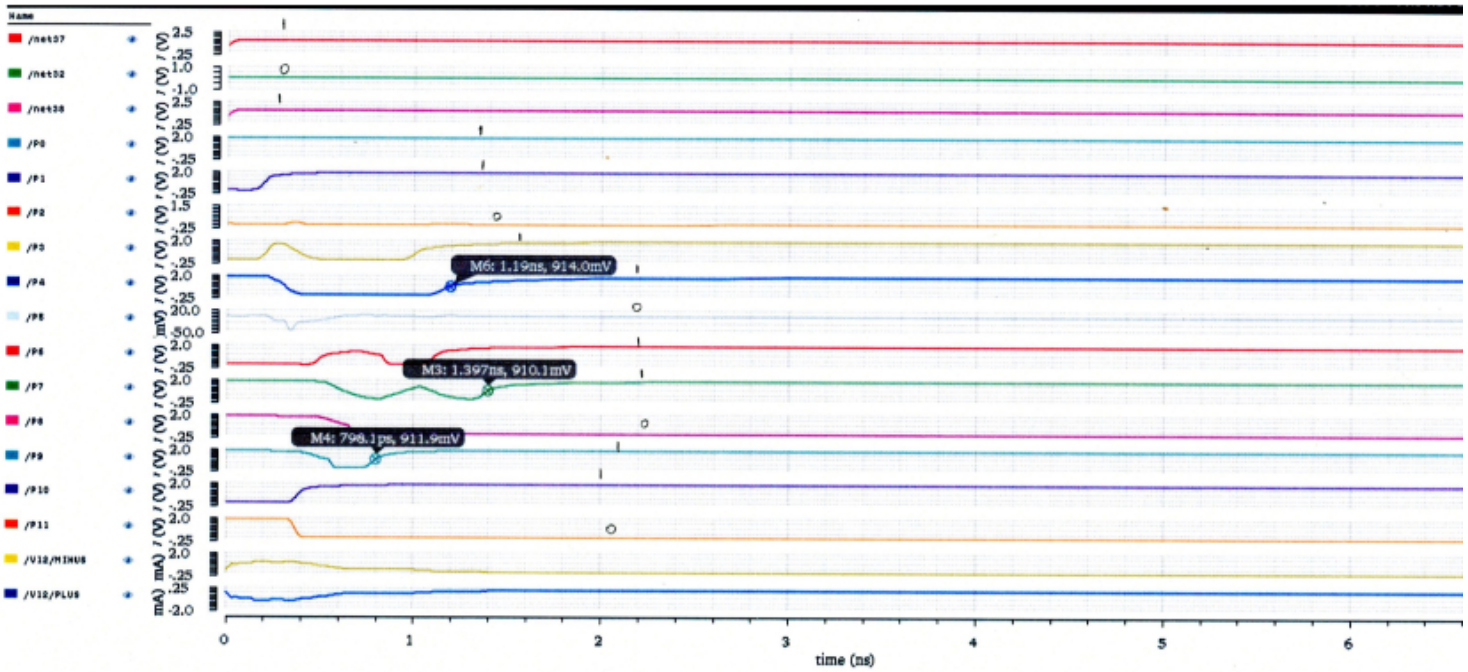
Binary '1' - 1.8V
Binary '0' - 0V



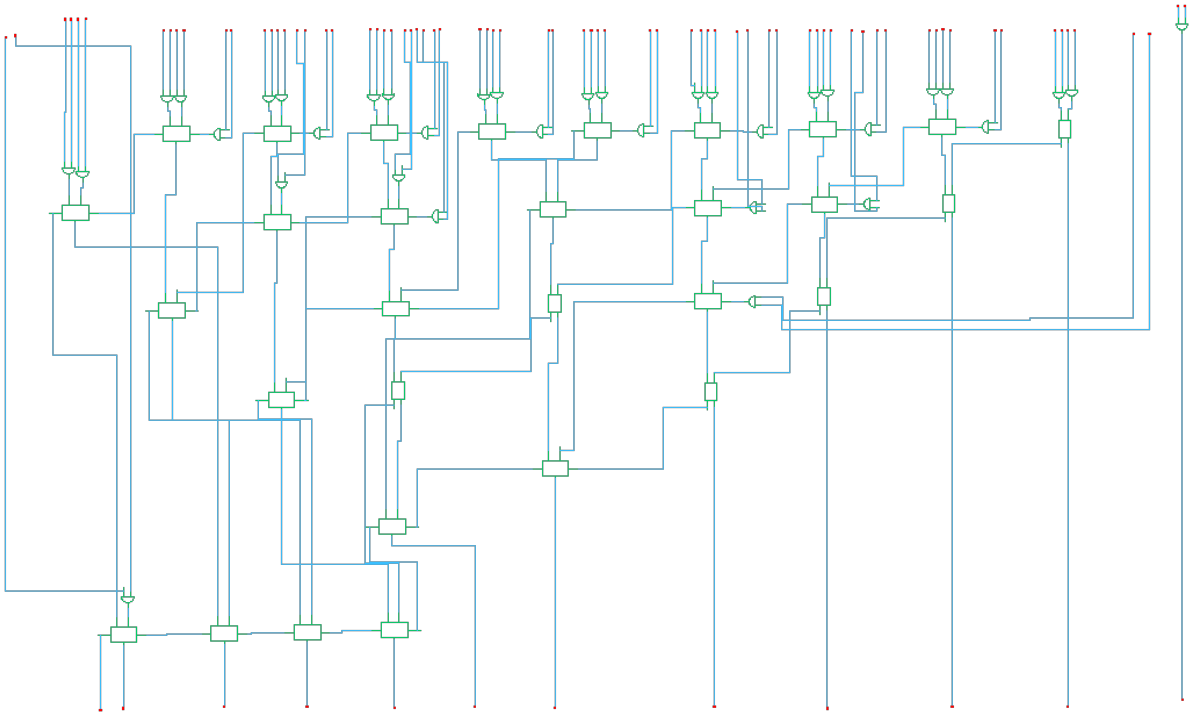
Propagation Delay = 1.197ns [output P_7] - 7.005ps

= 1.19ns

Simulation time : 200ns

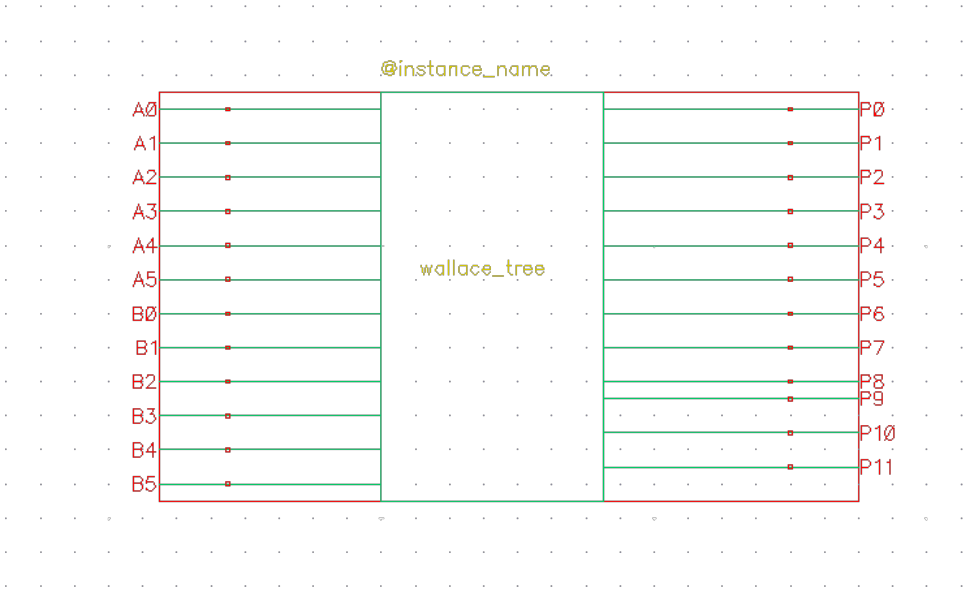


Wallace Tree Multiplier:

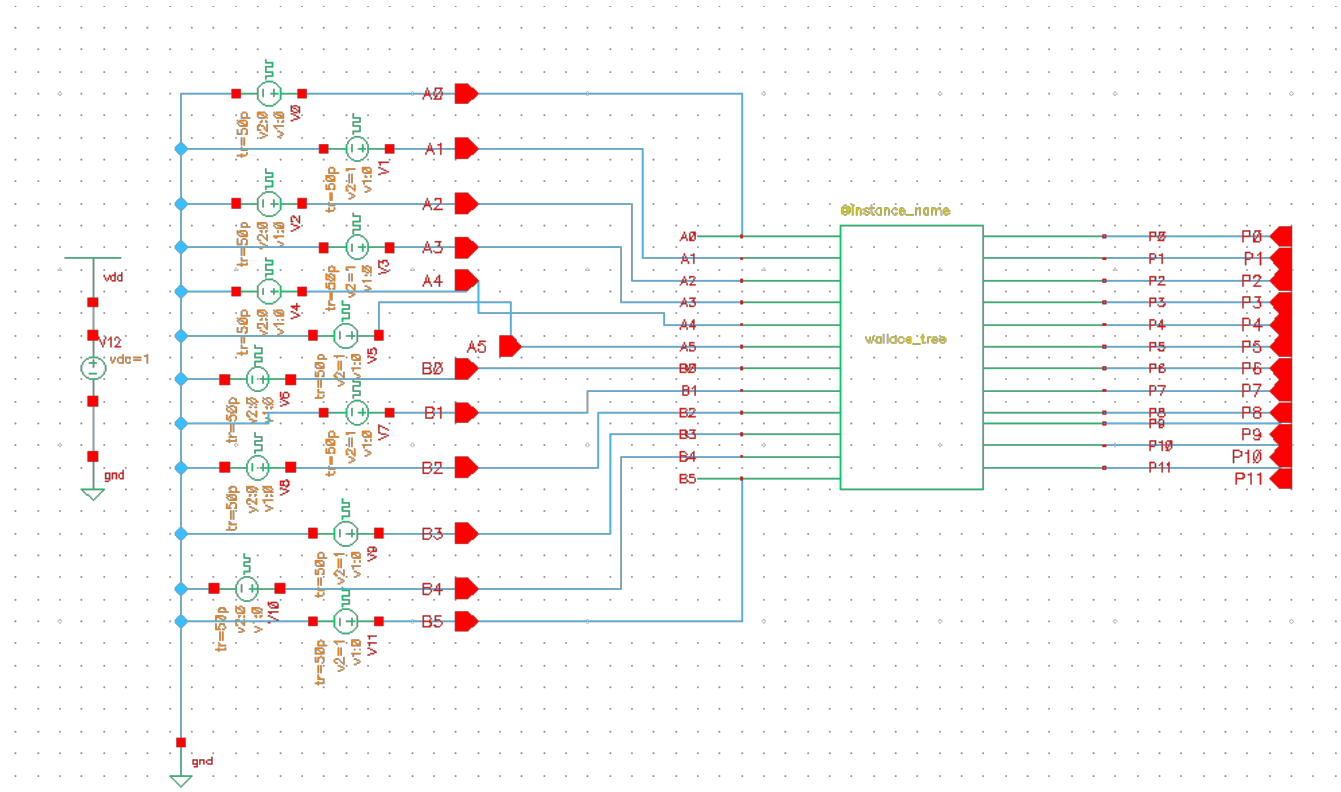


Wallce Tree Multiplier Schematic

Full-adders with a binary ‘0’ as one of the inputs were replaced with half-adders.



Wallce Tree Multiplier Symbol



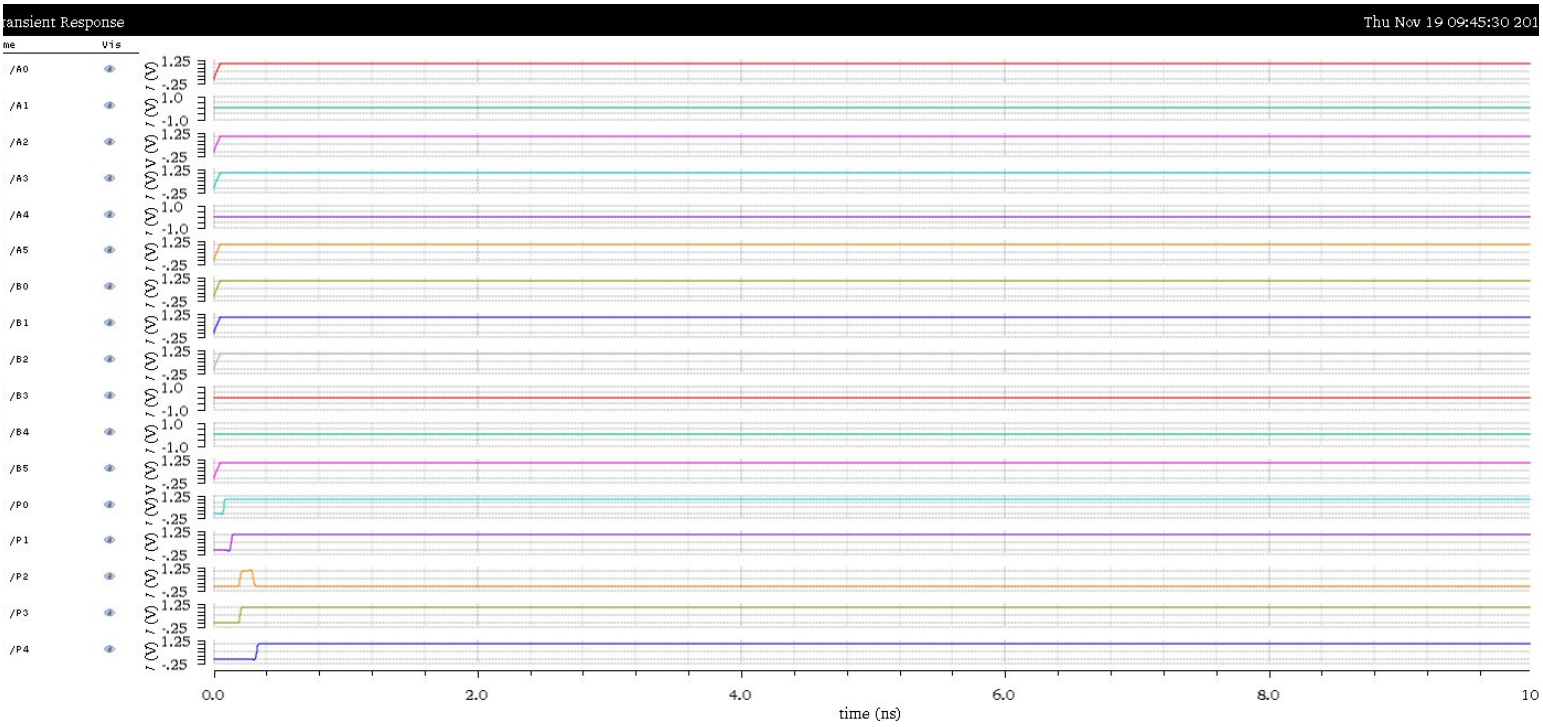
Wallace Tree Multiplier Test Bench

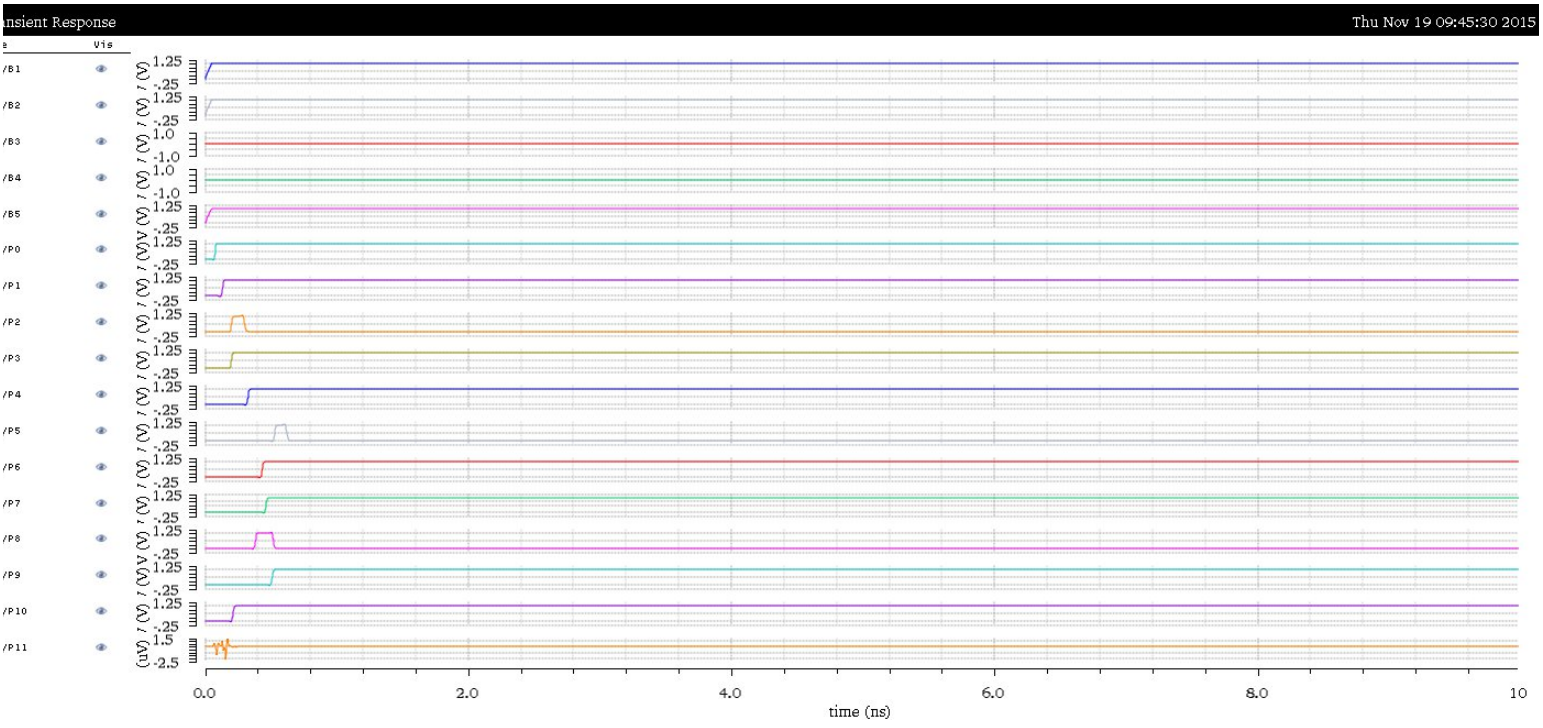
Simulations:

All the inputs were defined as with a voltage waveform defined by “vpulse”. Active high voltage was used with binary ‘1’ being represented by voltage level of ‘1v’, and binary ‘0’ being represented by a voltage level of ‘0v’.

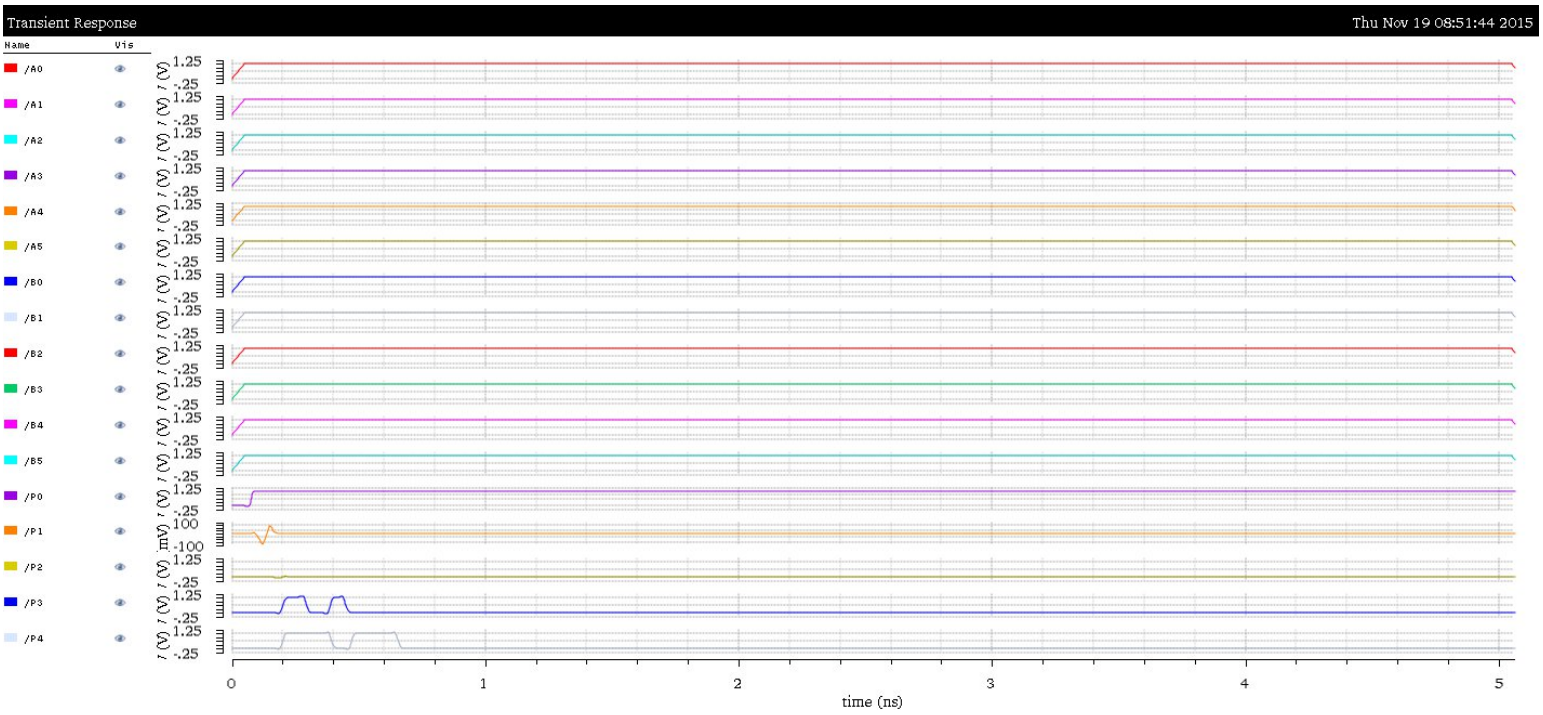
Binary ‘1’ was defined by “vpulse” by giving ‘v1’ and v2’ as 0v and 1v. Binary ‘0’ was similarly defined but both ‘v1’ and ‘v2’ were given a voltage level of ‘0v’.

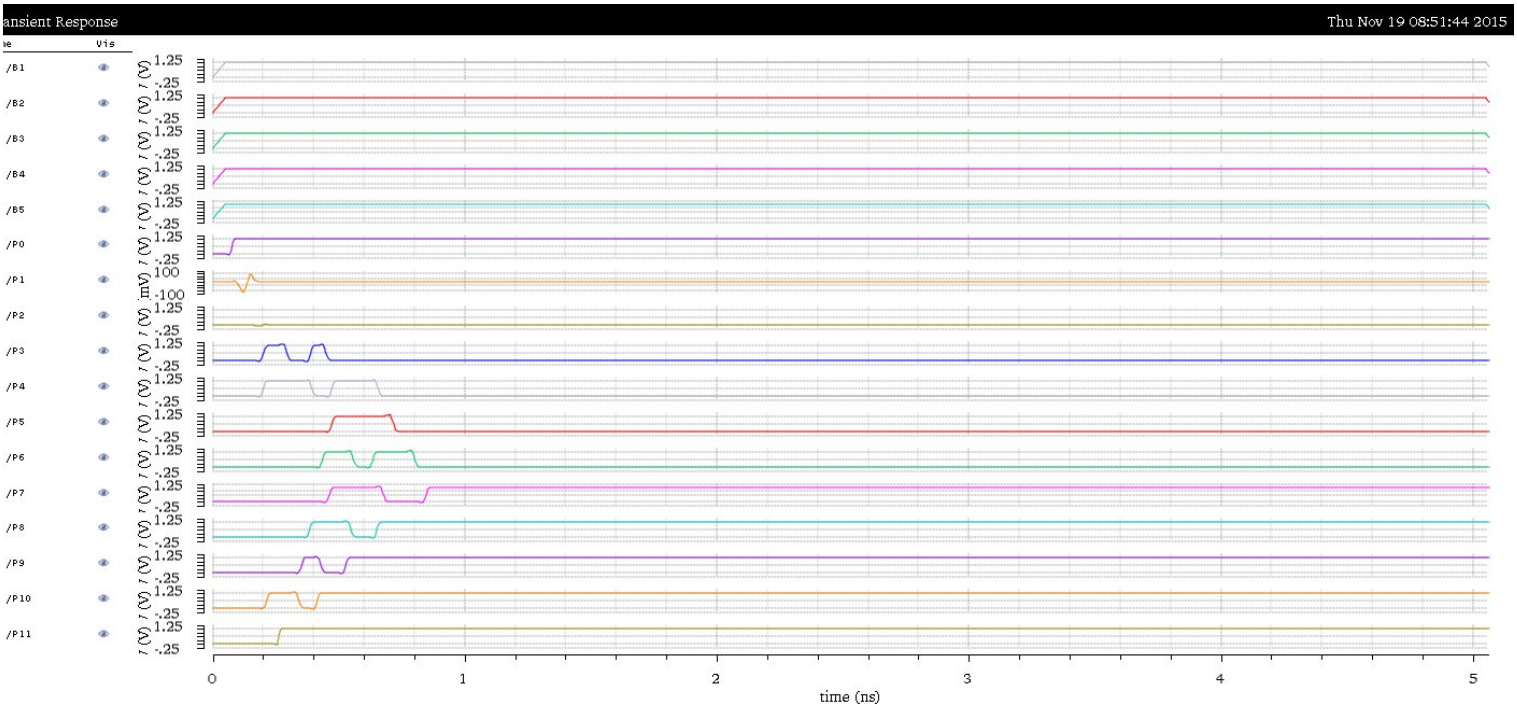
Combination 1:

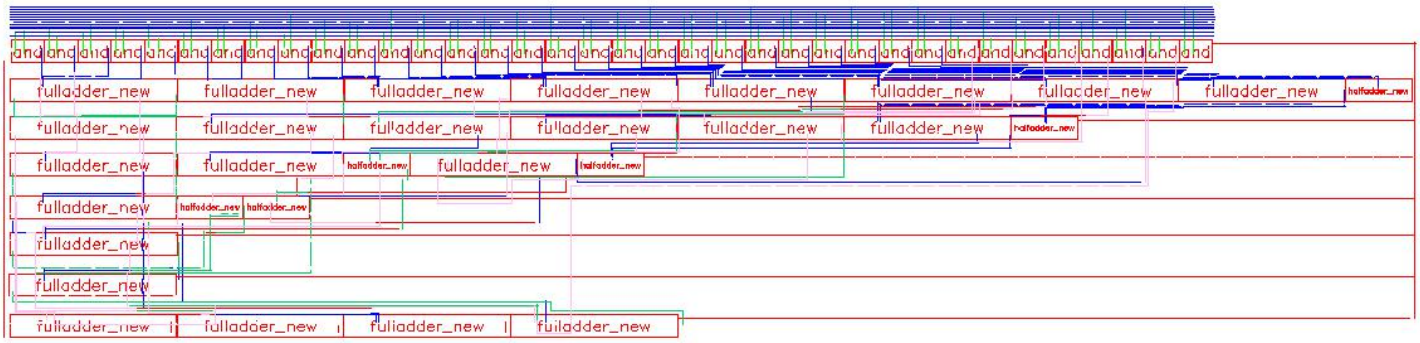




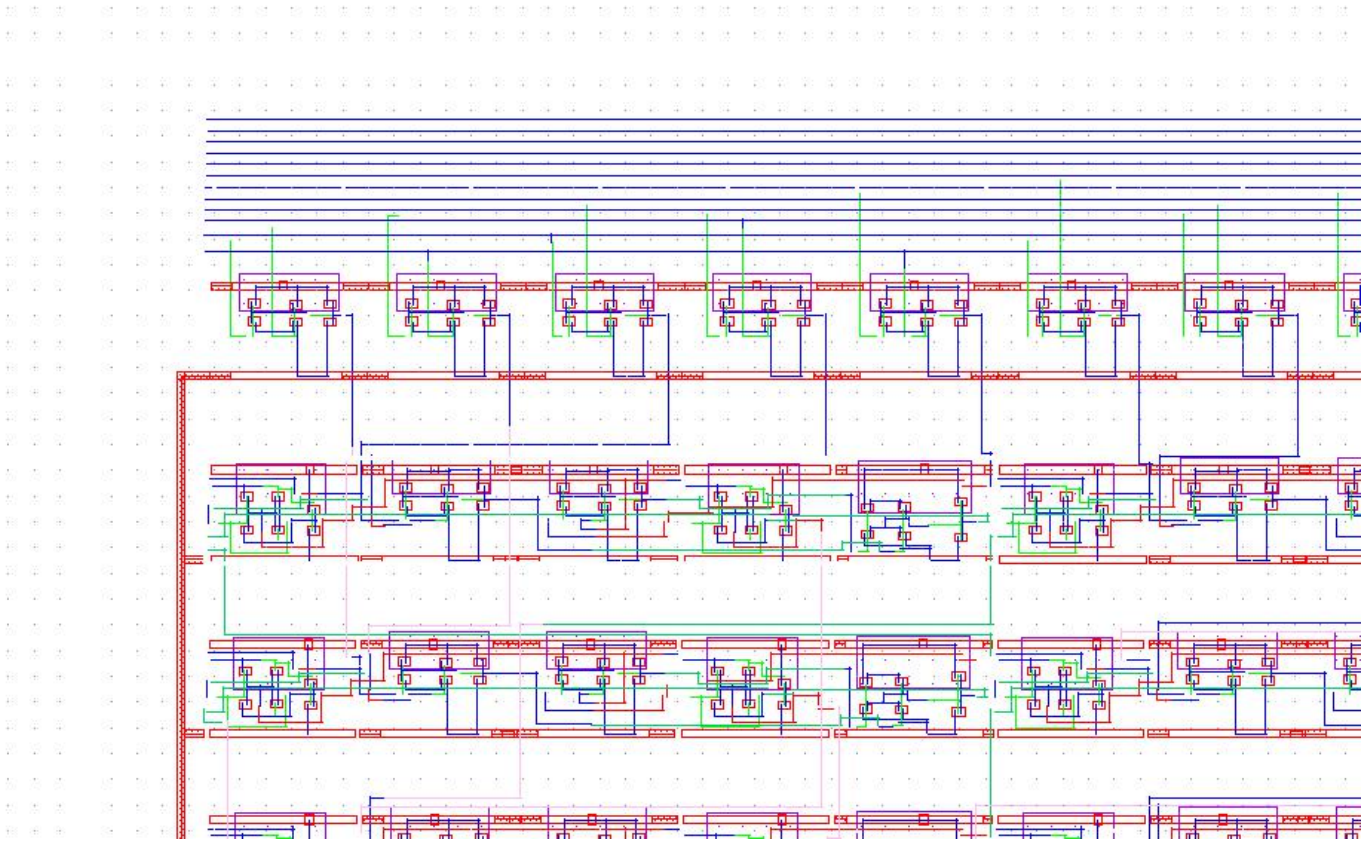
Combination 2:



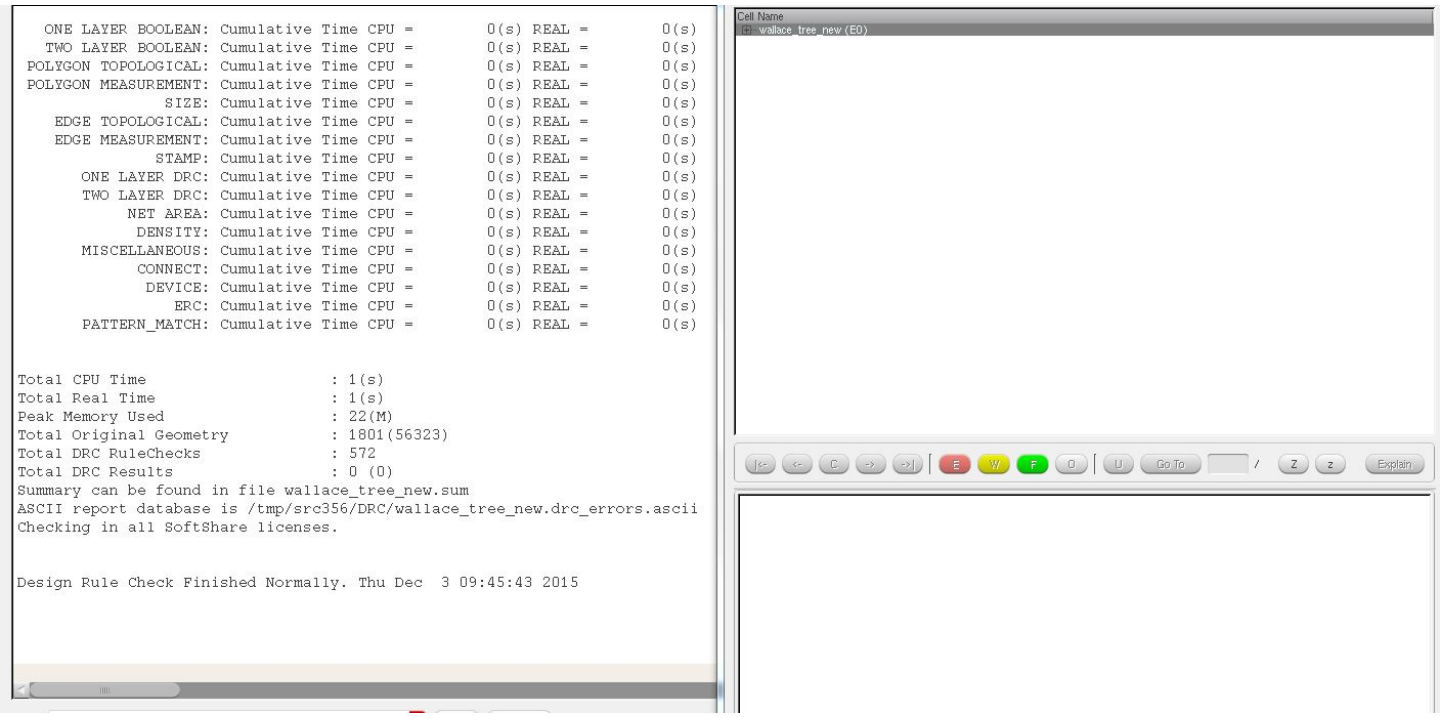




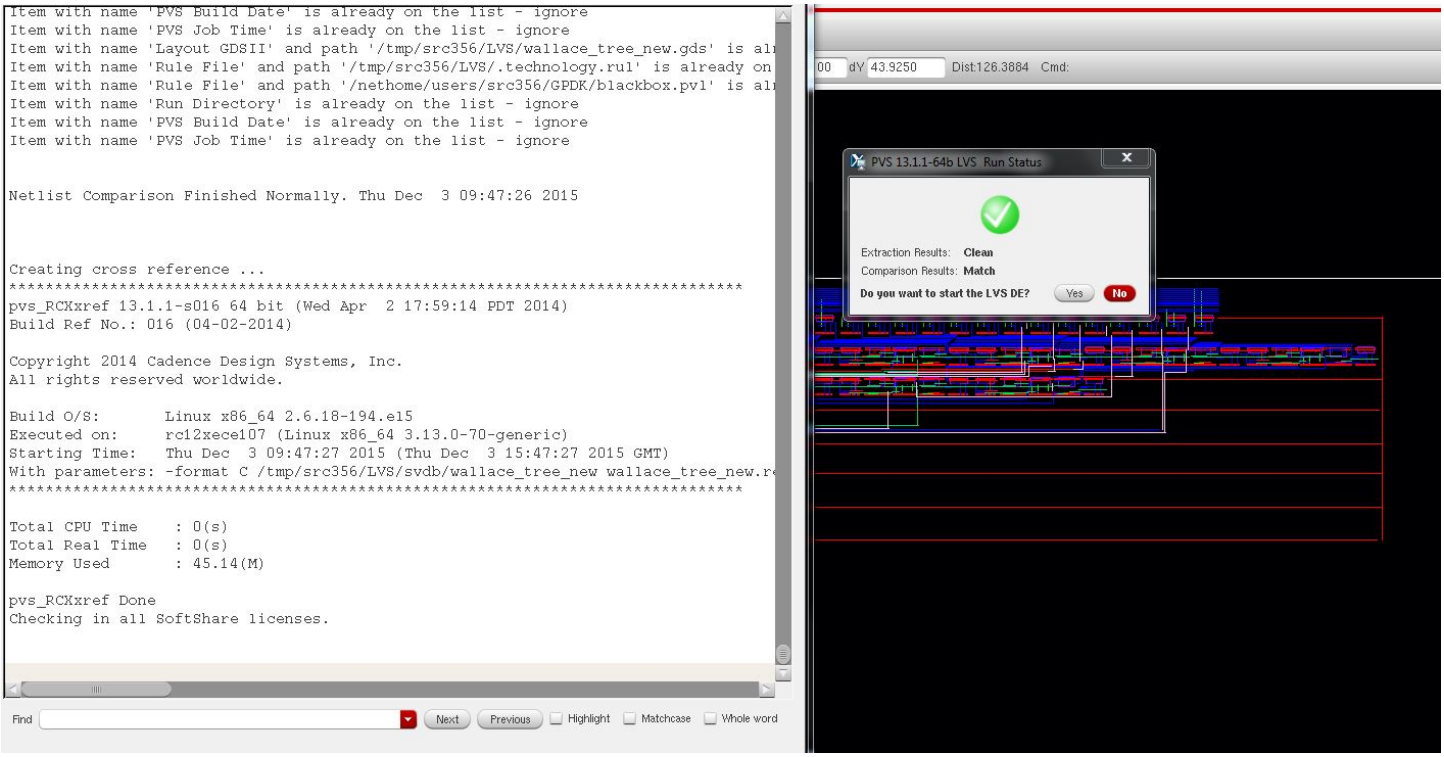
Vallace Tree Layout



Wallace Tree Multiplier Layout (Zoomed in)



DRC for Complete Wallace Tree Multiplier

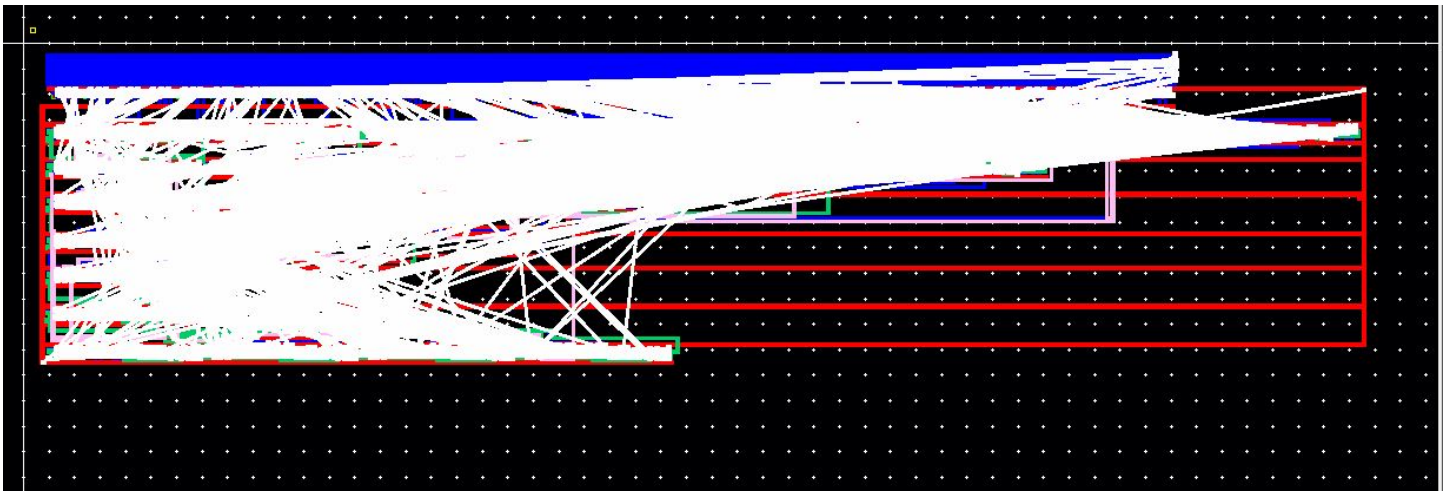


LVS Check for complete Wallace Tree Multiplier

AV extraction Wallace Tree:

They are parasitic capacitances, which are side effects formed by different layers you used for your layout

More info on this



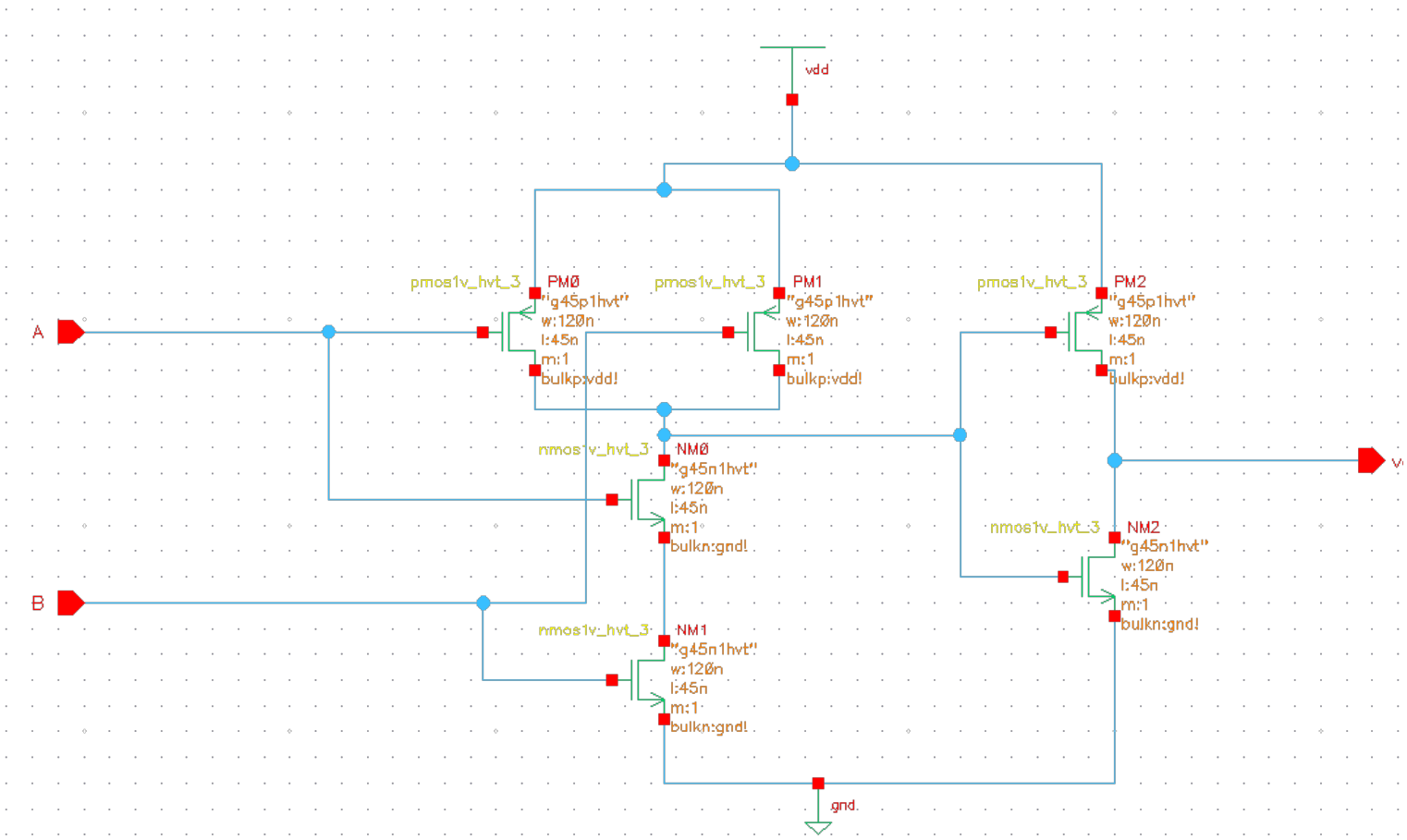
Basic Components:

- And Gate
- OR Gate
- XOR Gate
- Half Adder
- Full Adder

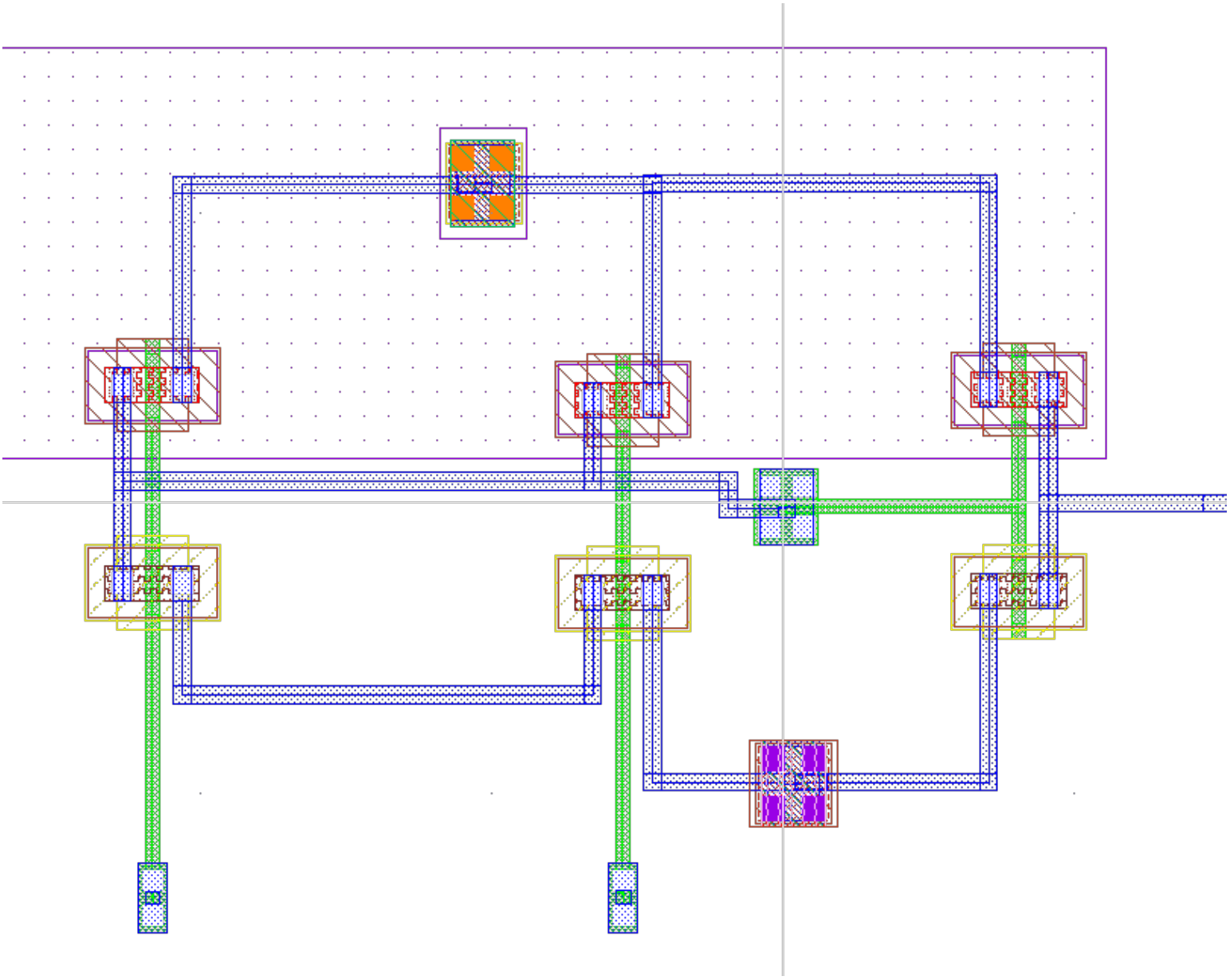
AND Gate:

1. Transistor level schematic:

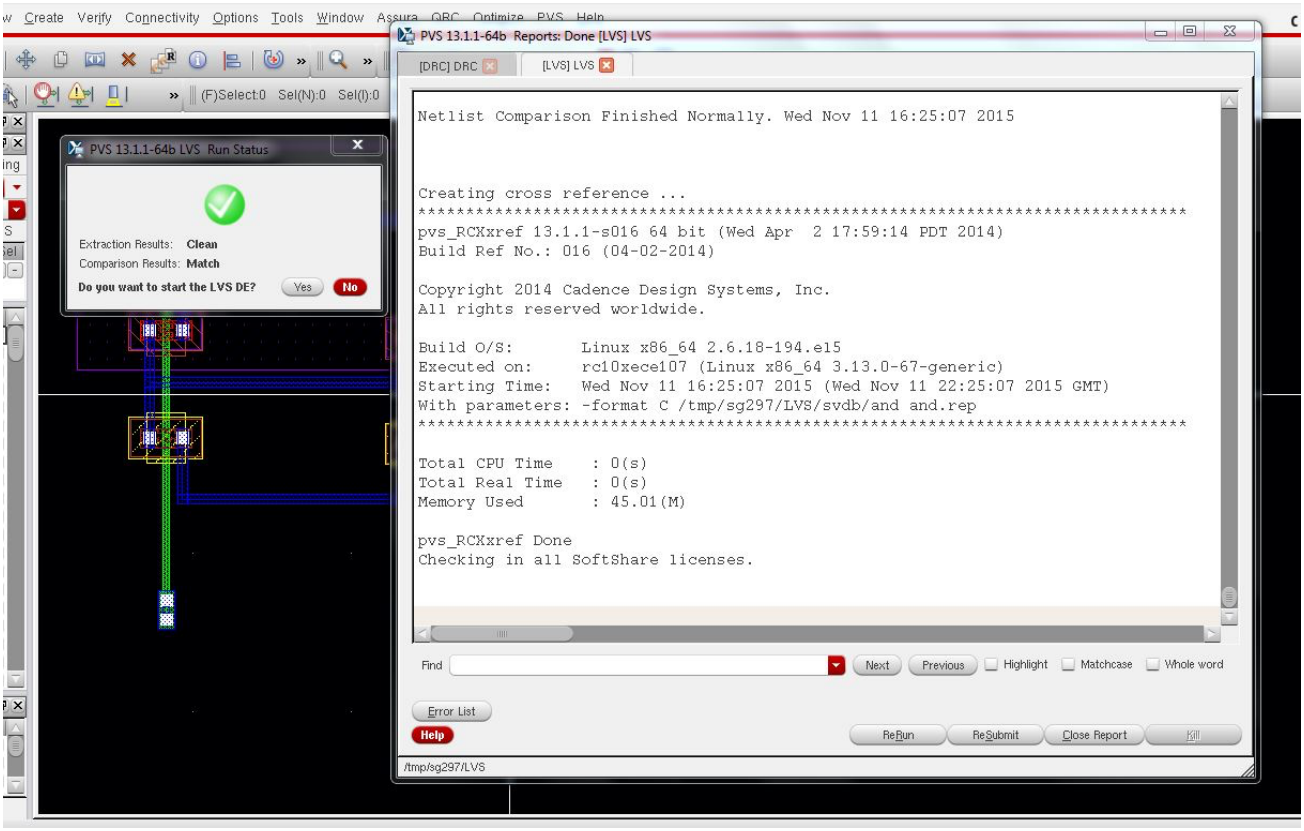
Inputs: A,B.
Output: vout



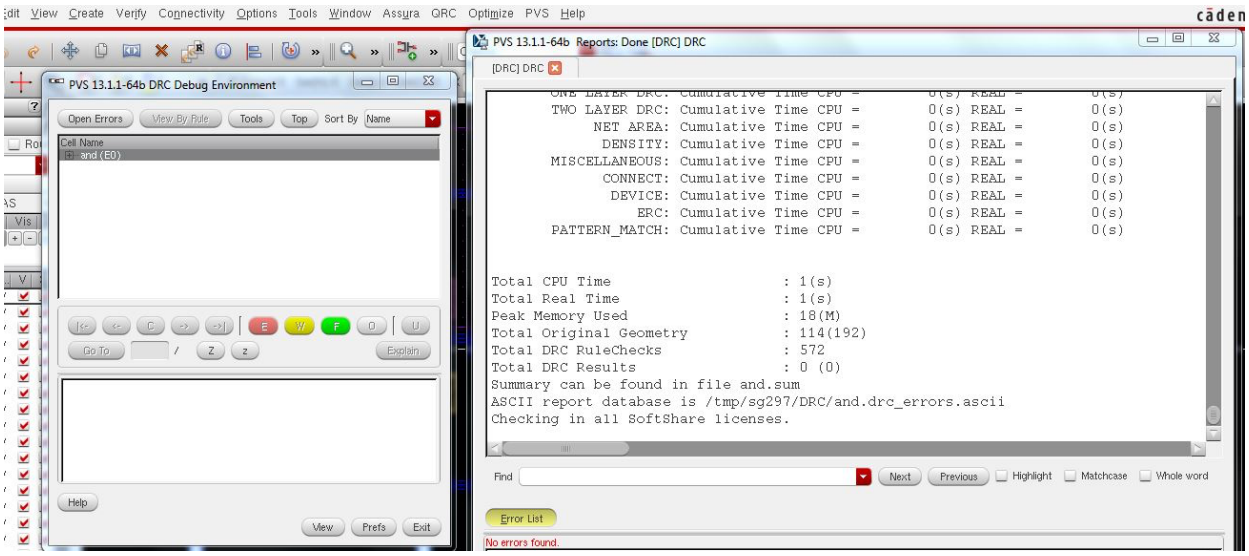
2. Layout



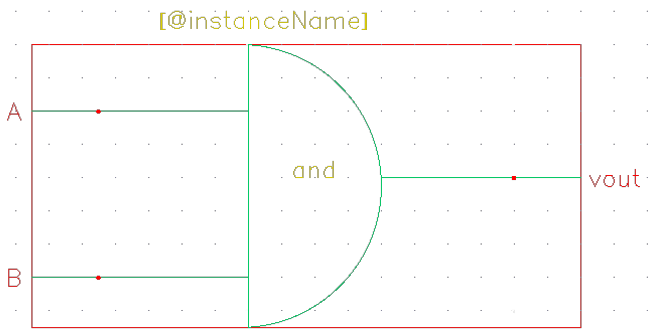
3.LVS


PVS 13.1.1-64b LVS Run Status
Extraction Results: Clean
Comparison Results: Match
Do you want to start the LVS DE?
Yes
No
PVS 13.1.1-64b Reports: Done [LVS] LVS
[DRC] DRC
[LVS] LVS
Netlist Comparison Finished Normally. Wed Nov 11 16:25:07 2015
Creating cross reference ...
pvs_RCXxref 13.1.1-s016 64 bit (Wed Apr 2 17:59:14 PDT 2014)
Build Ref No.: 016 (04-02-2014)
Copyright 2014 Cadence Design Systems, Inc.
All rights reserved worldwide.
Build O/S: Linux x86_64 2.6.18-194.e15
Executed on: rc10xece107 (Linux x86_64 3.13.0-67-generic)
Starting Time: Wed Nov 11 16:25:07 2015 (Wed Nov 11 22:25:07 2015 GMT)
With parameters: -format C /tmp/sg297/LVS/svdb/and and.rep
Total CPU Time : 0(s)
Total Real Time : 0(s)
Memory Used : 45.01(M)
pvs_RCXxref Done
Checking in all SoftShare licenses.
Find
Next
Previous
Highlight
Matchcase
Whole word
Error List
Help
ReRun
ReSubmit
Close Report
Kill
/tmp/sg297/LVS

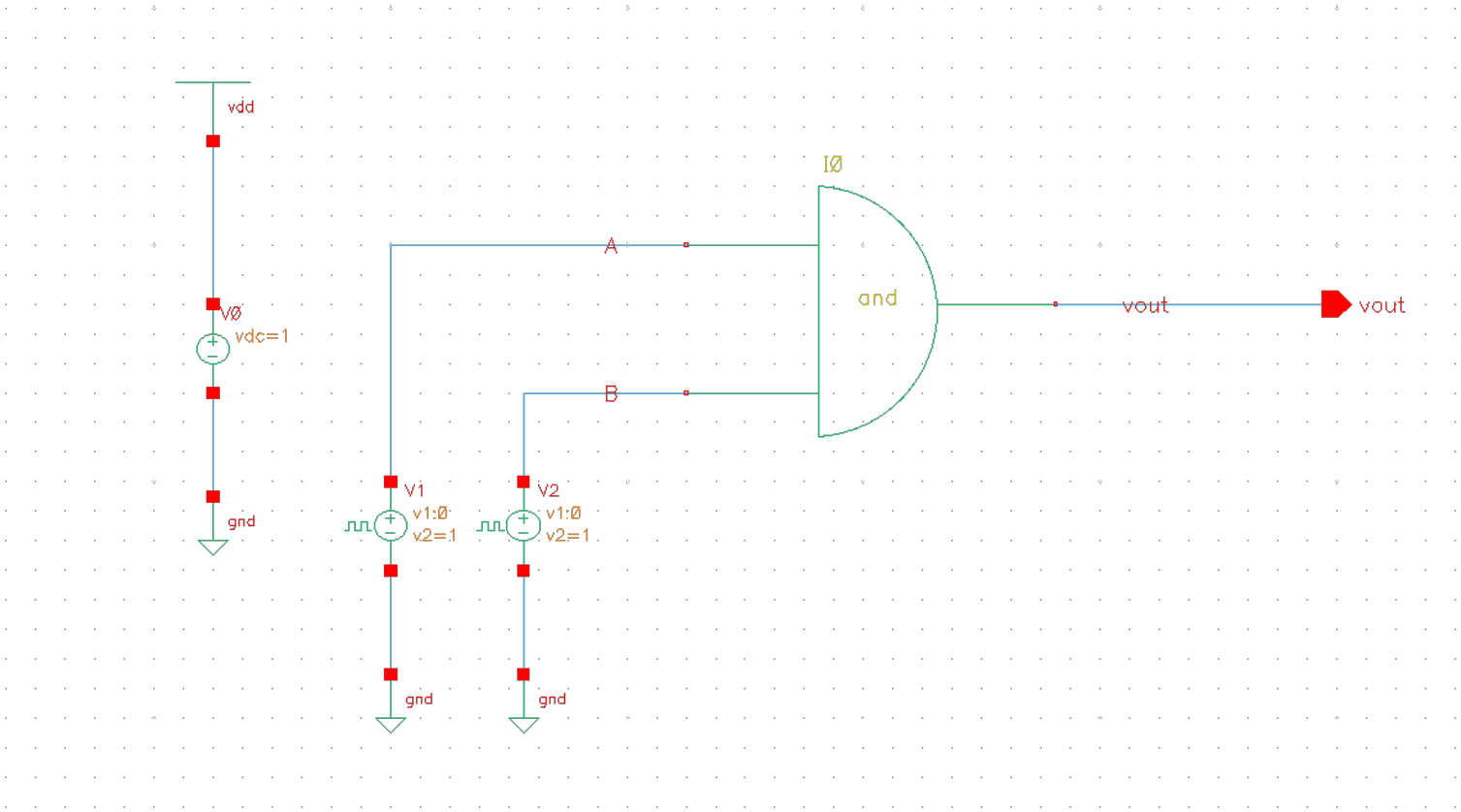
4.DRC


PVS 13.1.1-64b DRC Debug Environment
Open Errors
View By Rule
Tools
Top
Sort By
Name
Cell Name
and (E0)
Help
View
Prefs
Exit
PVS 13.1.1-64b Reports: Done [DRC] DRC
[DRC] DRC
TWO LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
NET AREA: Cumulative Time CPU = 0(s) REAL = 0(s)
DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
Total CPU Time : 1(s)
Total Real Time : 1(s)
Peak Memory Used : 18(M)
Total Original Geometry : 114(192)
Total DRC RuleChecks : 572
Total DRC Results : 0 (0)
Summary can be found in file and.sum
ASCII report database is /tmp/sg297/DRC/and.drc_errors.ascii
Checking in all SoftShare licenses.
Find
Next
Previous
Highlight
Matchcase
Whole word
Error List
No errors found.

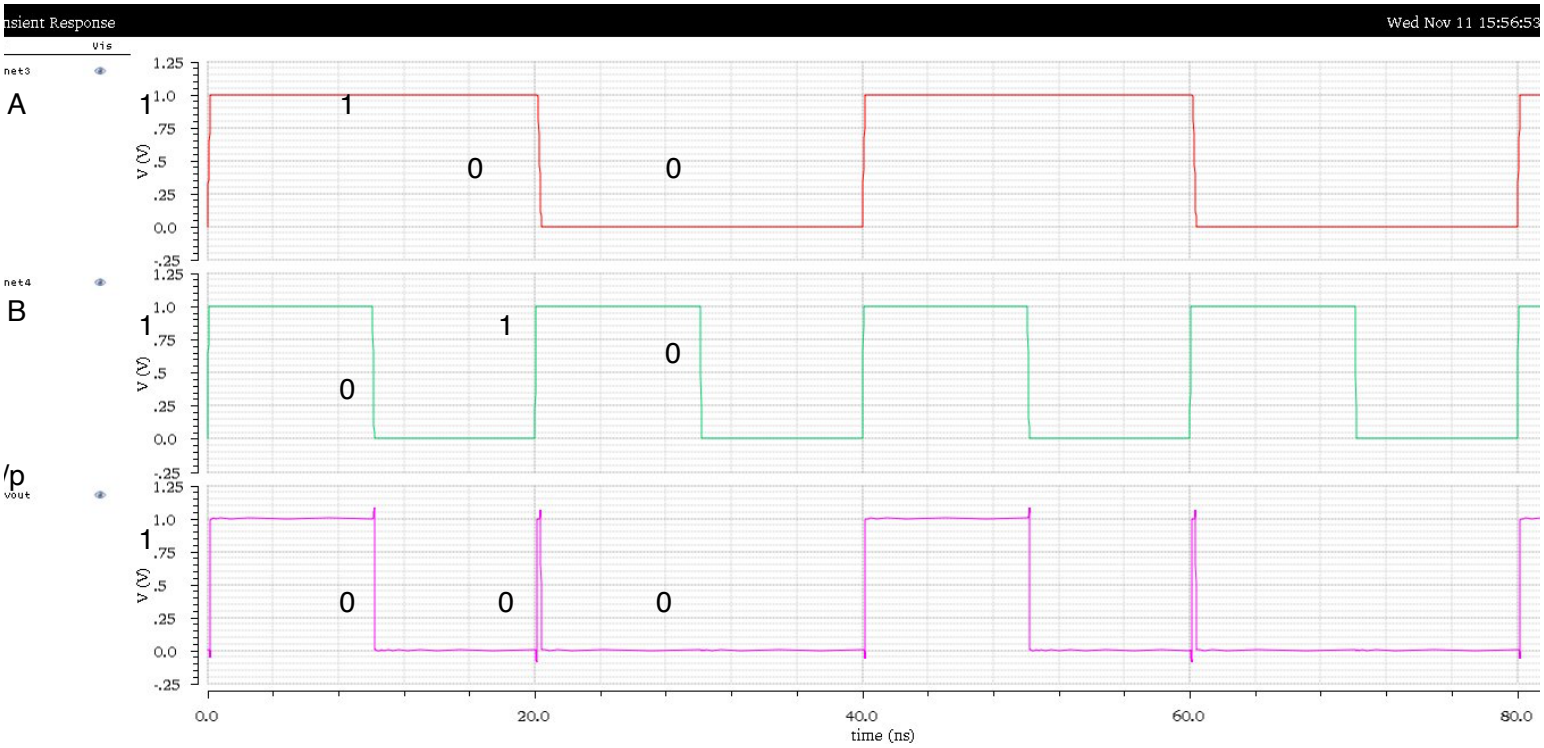
5. Symbol



6. Testbench Schematic



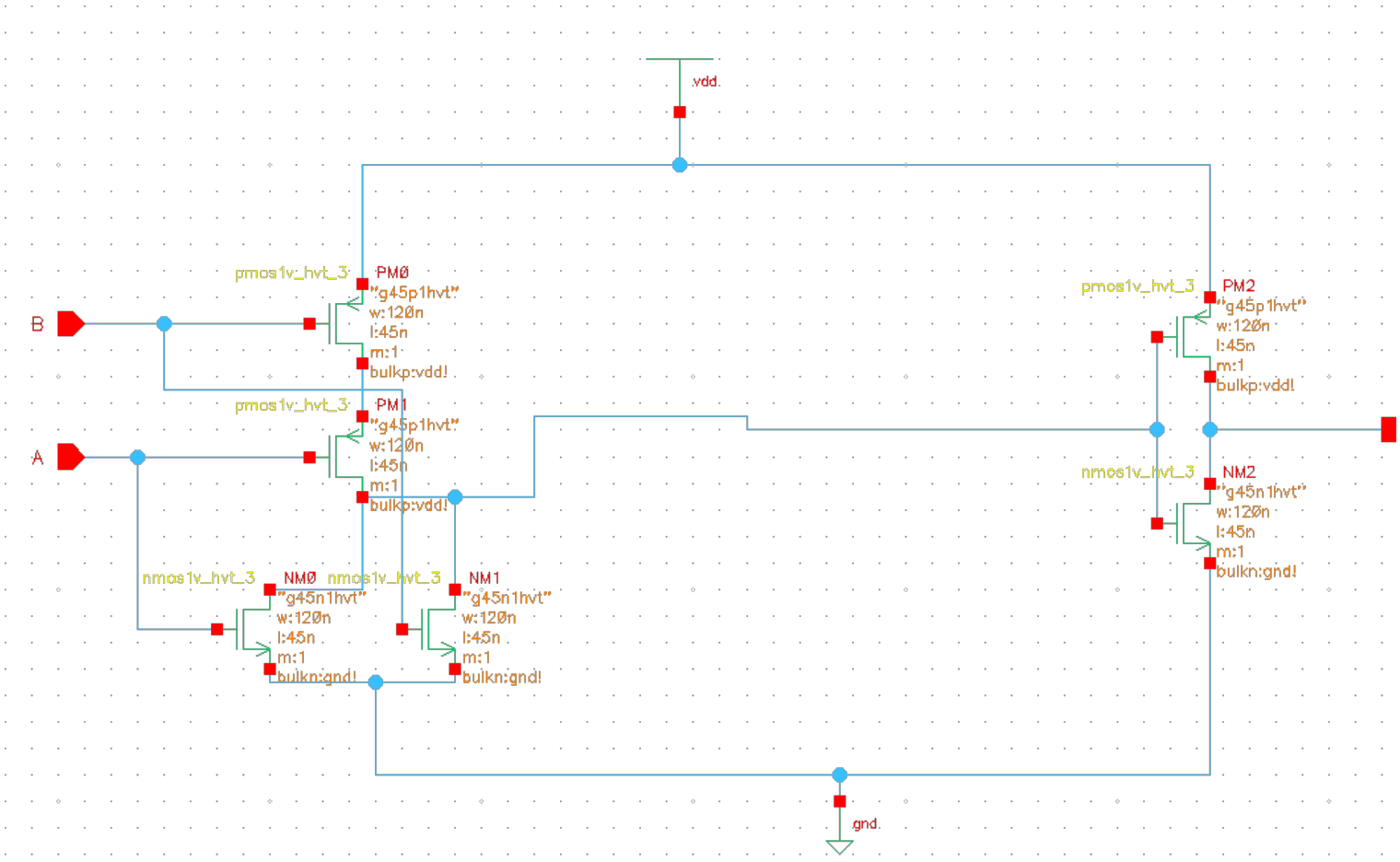
7. Output simulations



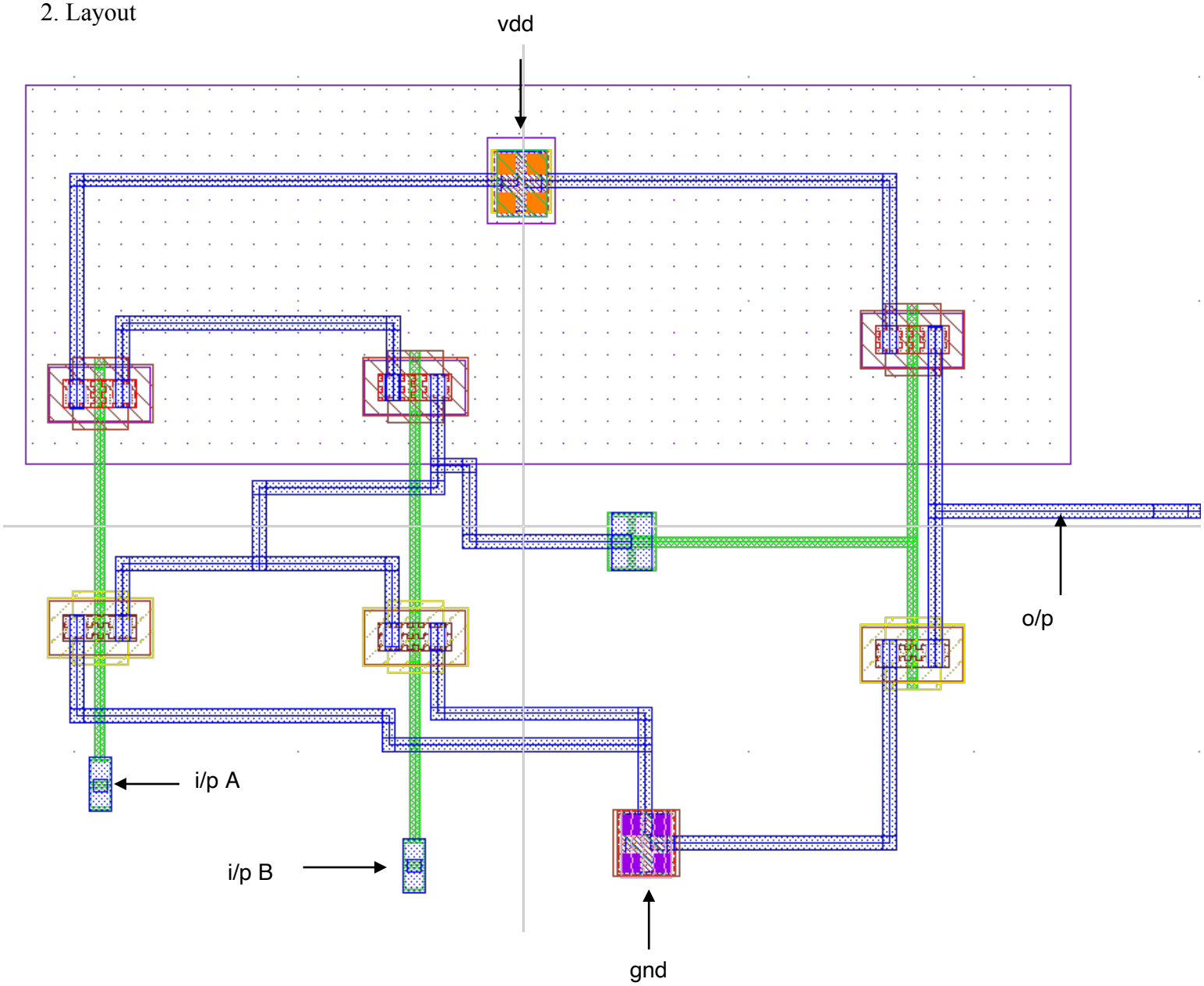
A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate:

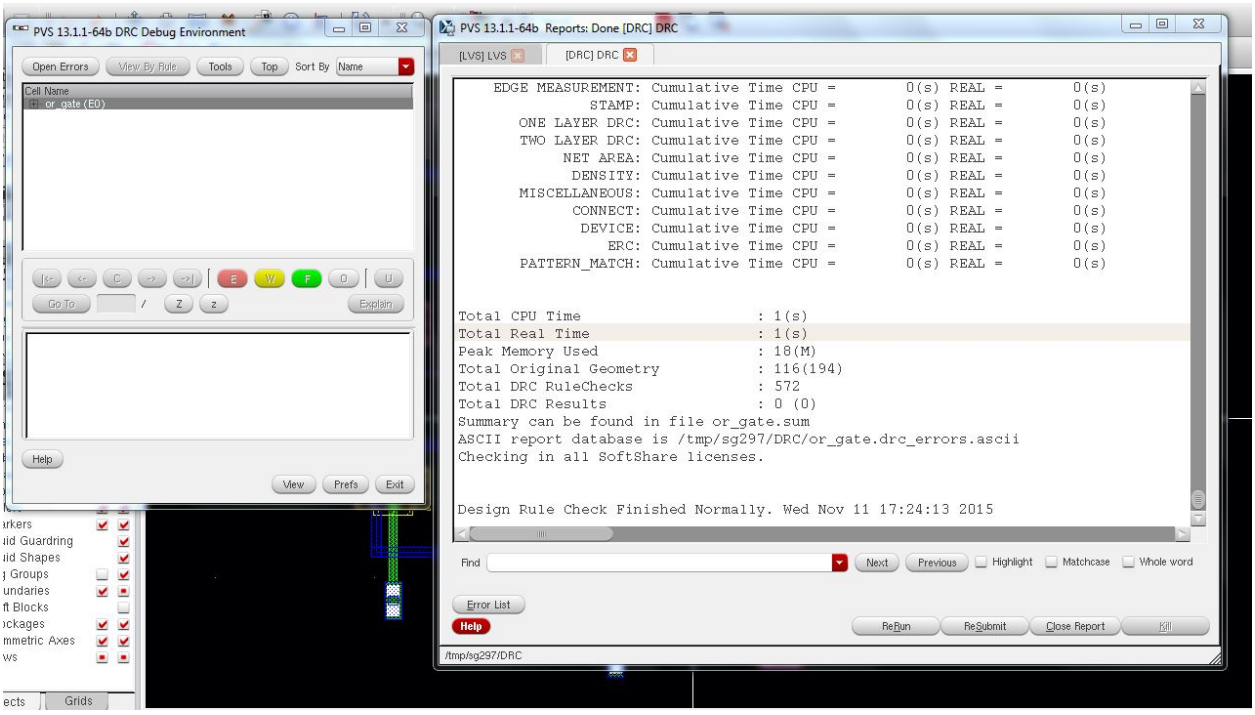
1. Transistor level schematic
Inputs: A,B
Output:vout



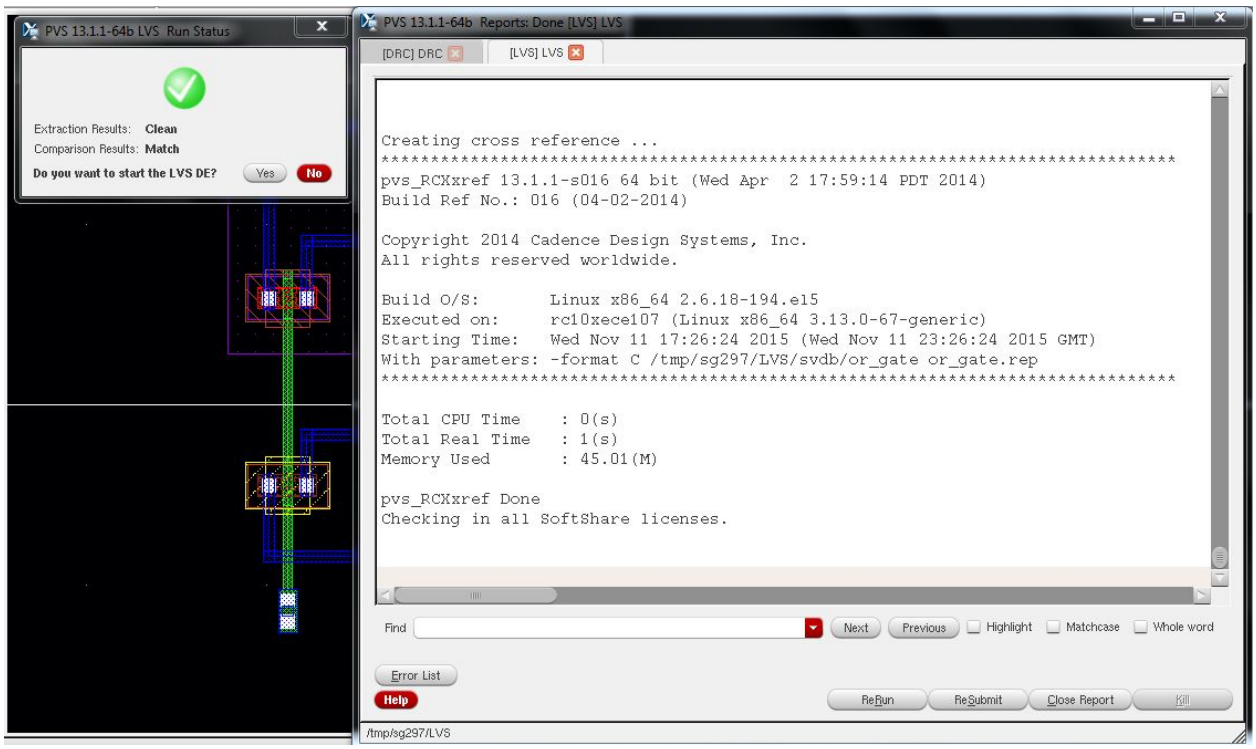
2. Layout



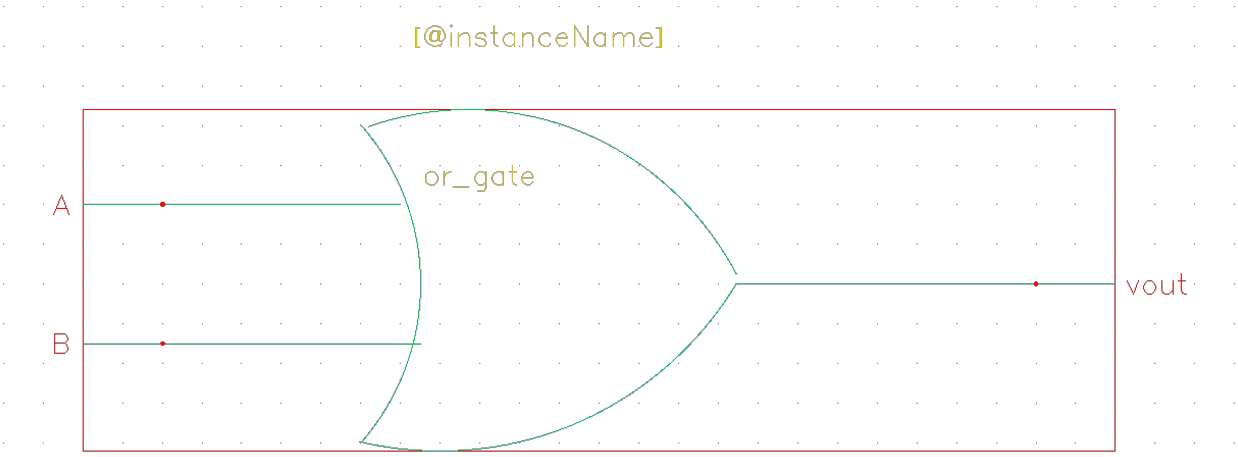
3. DRC


PVS 13.1.1-64b DRC Debug Environment
Open Errors
View By Rule
Tools
Top
Sort By
Name
Cell Name
or_gate (EO)
Go To
Explain
Help
View
Prefs
Exit
PVS 13.1.1-64b Reports: Done [DRC] DRC
[LVS] LVS
[DRC] DRC
EDGE MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
STAMP: Cumulative Time CPU = 0(s) REAL = 0(s)
ONE LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
NET AREA: Cumulative Time CPU = 0(s) REAL = 0(s)
DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
Total CPU Time : 1(s)
Total Real Time : 1(s)
Peak Memory Used : 18(M)
Total Original Geometry : 116(194)
Total DRC RuleChecks : 572
Total DRC Results : 0 (0)
Summary can be found in file or_gate.sum
ASCII report database is /tmp/sg297/DRC/or_gate.drc_errors.ascii
Checking in all SoftShare licenses.
Design Rule Check Finished Normally. Wed Nov 11 17:24:13 2015
Find
Next
Previous
Highlight
Matchcase
Whole word
Error List
Help
ReRun
ReSubmit
Close Report
Kill
/tmp/sg297/DRC
Grids

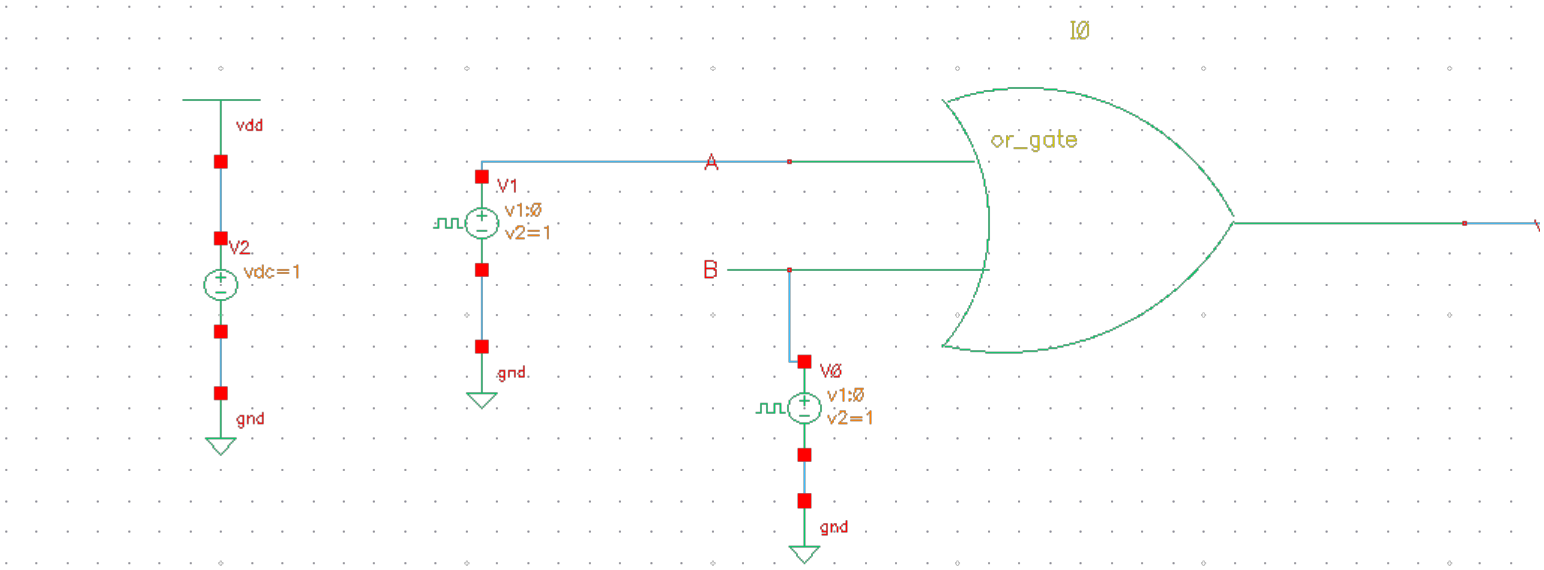
4.LVS


PVS 13.1.1-64b LVS Run Status
Extraction Results: Clean
Comparison Results: Match
Do you want to start the LVS DE?
Yes
No
PVS 13.1.1-64b Reports: Done [LVS] LVS
[DRC] DRC
[LVS] LVS
Creating cross reference ...
pvs_RCXxref 13.1.1-s016 64 bit (Wed Apr 2 17:59:14 PDT 2014)
Build Ref No.: 016 (04-02-2014)
Copyright 2014 Cadence Design Systems, Inc.
All rights reserved worldwide.
Build O/S: Linux x86_64 2.6.18-194.el5
Executed on: rc10xece107 (Linux x86_64 3.13.0-67-generic)
Starting Time: Wed Nov 11 17:26:24 2015 (Wed Nov 11 23:26:24 2015 GMT)
With parameters: -format C /tmp/sg297/LVS/svdb/or_gate or_gate.rep
Total CPU Time : 0(s)
Total Real Time : 1(s)
Memory Used : 45.01(M)
pvs_RCXxref Done
Checking in all SoftShare licenses.
Find
Next
Previous
Highlight
Matchcase
Whole word
Error List
Help
ReRun
ReSubmit
Close Report
Kill
/tmp/sg297/LVS

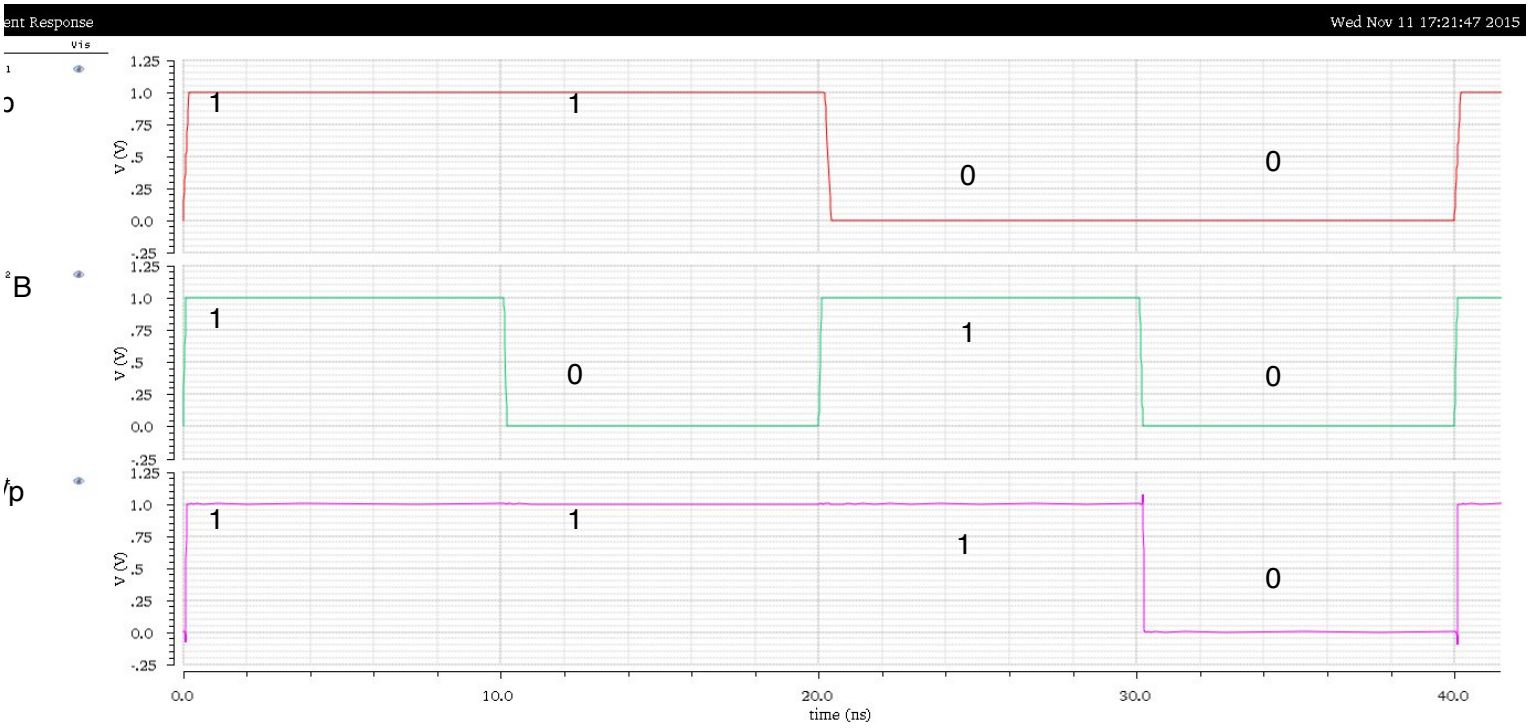
5.Symbol



6. Testbench Schematic



7. Output Simulations



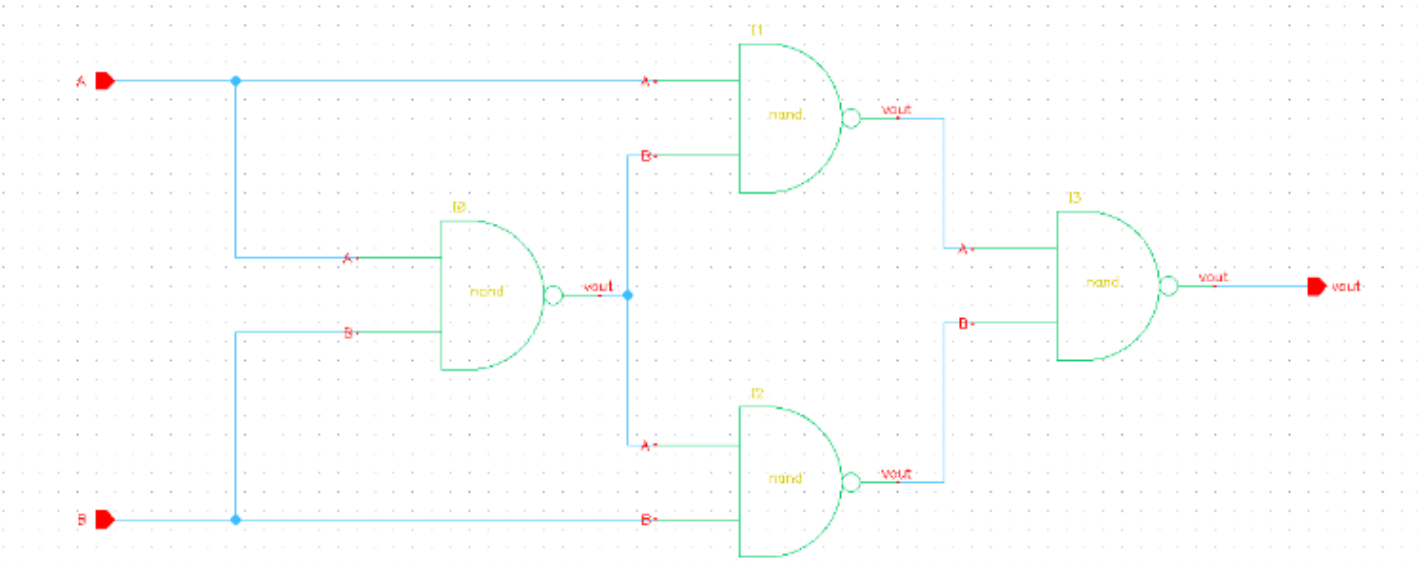
As observed from the graph the output simulation match that of the truth table shown below.

A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

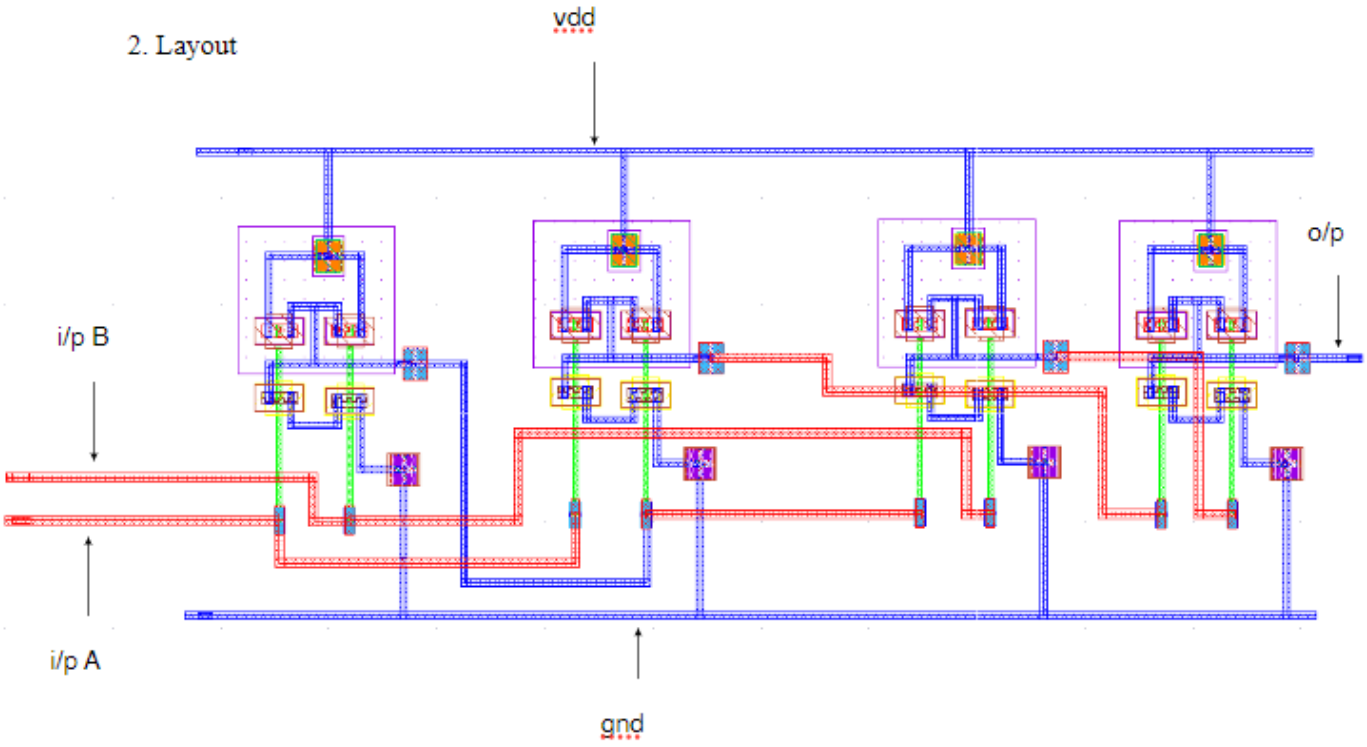
XOR Gate:

Previously implemented schematic/layout of XOR with NAND gates(16 Fets):

1. Schematic



2. Layout



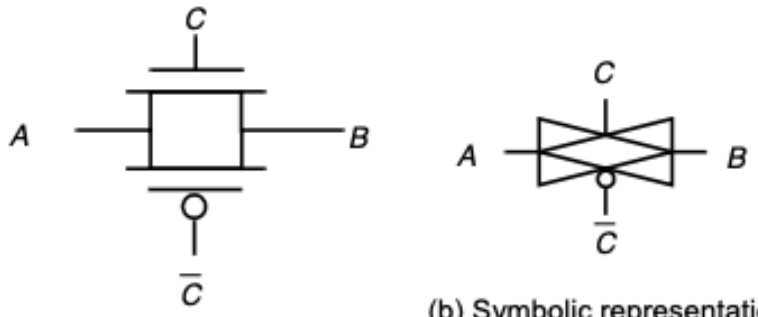


Figure 6.44 CMOS transmission gate.

Solution 3: Transmission Gate Logic. The most widely-used solution to deal with the voltage-drop problem is the use of *transmission gates*. It builds on the complementary properties of NMOS and PMOS transistors: NMOS devices pass a strong 0 but a weak 1, while PMOS transistors pass a strong 1 but a weak 0. The ideal approach is to use an NMOS to pull-down and a PMOS to pull-up. The transmission gate combines the best of both device flavors by placing a NMOS device in parallel with a PMOS device (Figure 6.44a). The control signals to the transmission gate (C and $\overline{C}$) are complementary. The transmission gate acts as a bidirectional switch controlled by the gate signal C. When C = 1, both MOSFETs are on, allowing the signal to pass through the gate. In short,

$$A = B \quad \text{if} \quad C = 1 \tag{6.26}$$

On the other hand, $C = 0$ places both transistors in cutoff, creating an open circuit between nodes A and B. Figure 6.44b shows a commonly used transmission-gate symbol.

Consider the case of charging node B to V_{DD} for the transmission gate circuit in Figure 6.45a. Node A is driven to V_{DD} and transmission gate is enabled (C = 1 and $\overline{C}$= 0). If only the NMOS pass-device were present, node B charges up to V_{DD}-V_{Tn} at which point the NMOS device turns off. However, since the PMOS device is present and turned on ($V_{GSp} = -V_{DD}$), charging continues all the way up to V_{DD}. Figure 6.45b shows the opposite case, this is discharging node B to 0. B is initially at V_{DD} when node A is driven low. The PMOS transistor by itself can only pull down node B to V_{Tp} at which point it turns *off*. The parallel NMOS device however stays turned on (since its $V_{GS} = V_{DD}$) and pulls node B all the way to *GND*. Though the transmission gate requires two transistors and more control signals, it enables rail-to-rail swing.

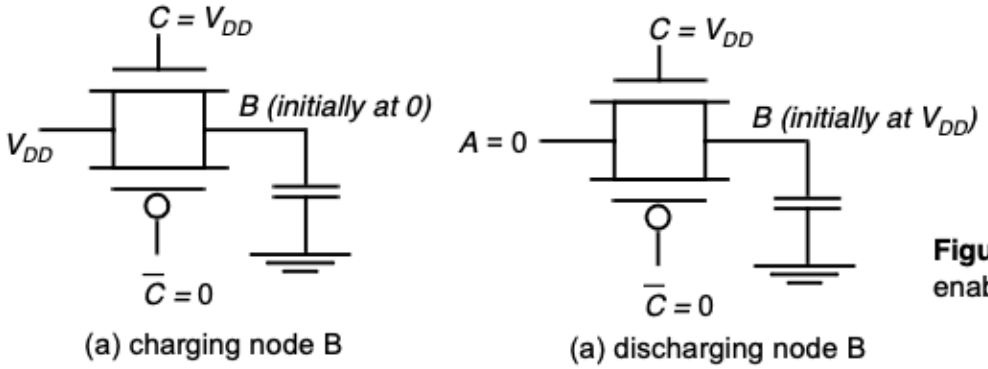


Figure 6.45 Transmission gates enable rail-to-rail switching

Another example of the effective use of transmission gates is the popular XOR circuit shown in Figure 6.47. The complete implementation of this gate requires only six transistors (including the inverter used for the generation of $\bar{B}$), compared to the twelve transistors required for a complementary implementation. To understand the operation of this circuit, we have to analyze the $B = 0$ and $B = 1$ cases separately. For $B = 1$, transistors M_1 and M_2 act as an inverter while the transmission gate M_3/M_4 is off; hence $F = \bar{A}B$. In the opposite case, M_1 and M_2 are disabled, and the transmission gate is operational, or $F = A\bar{B}$. The combination of both results in the XOR function. Notice that, regardless of the values of A and B, node F always has a connection to either V_{DD} or GND and is hence a low-impedance node. When designing static-pass transistor networks, it is essential to adhere to the low-impedance rule under all circumstances. Other examples where transmission-gate logic is effectively used are fast adder circuits and registers.

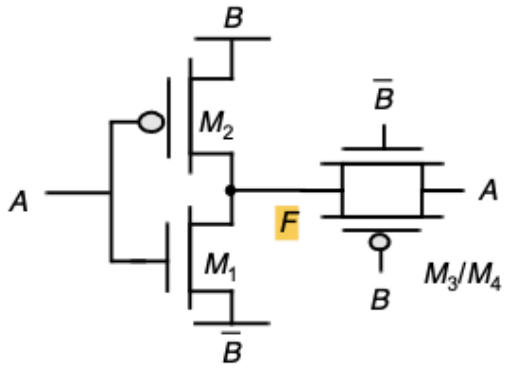
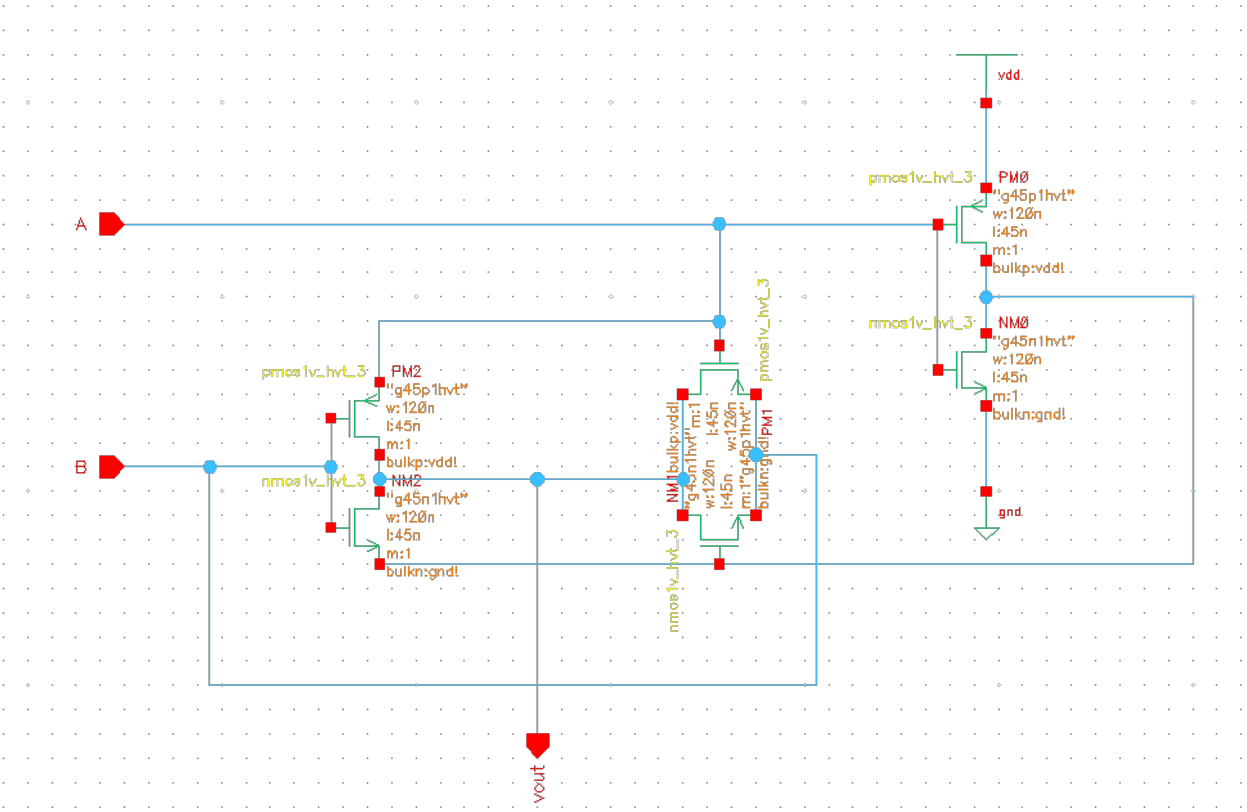


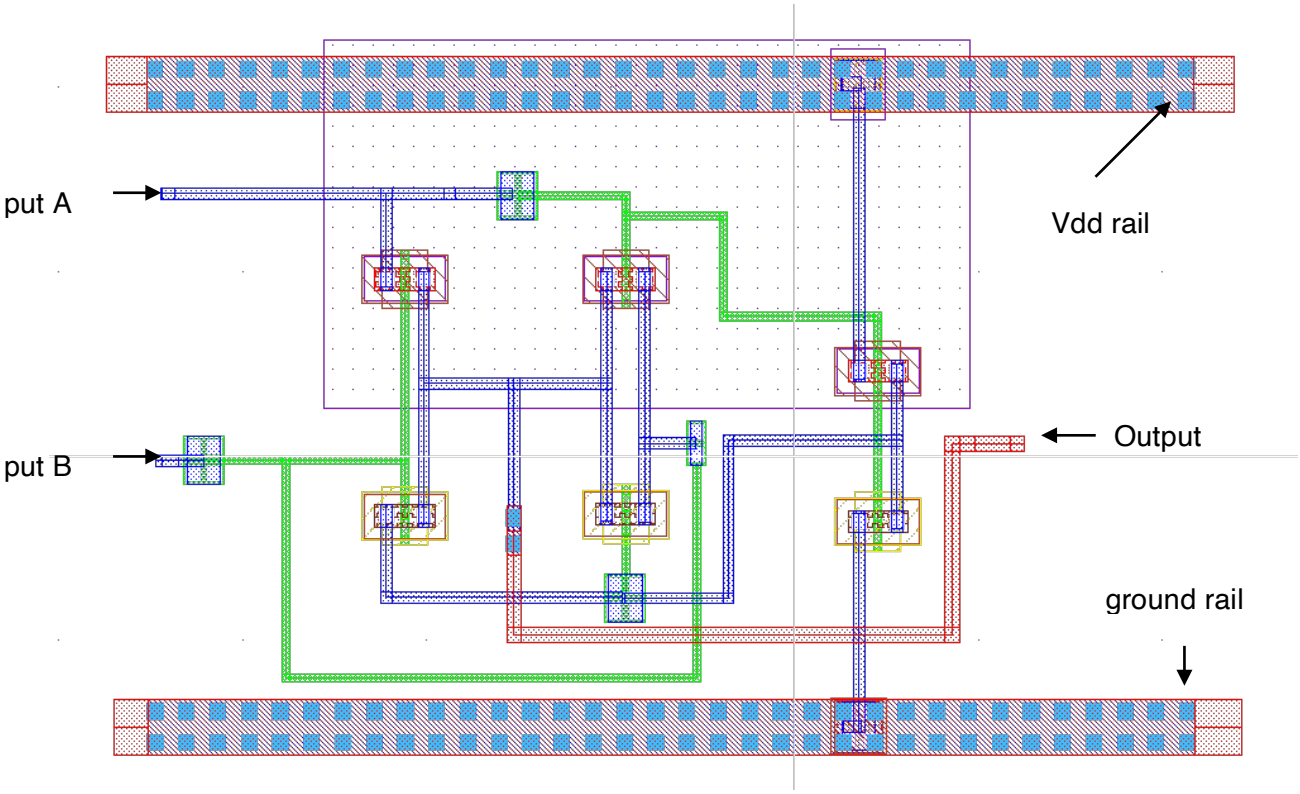
Figure 6.47 Transmission gate XOR.

- Schematic

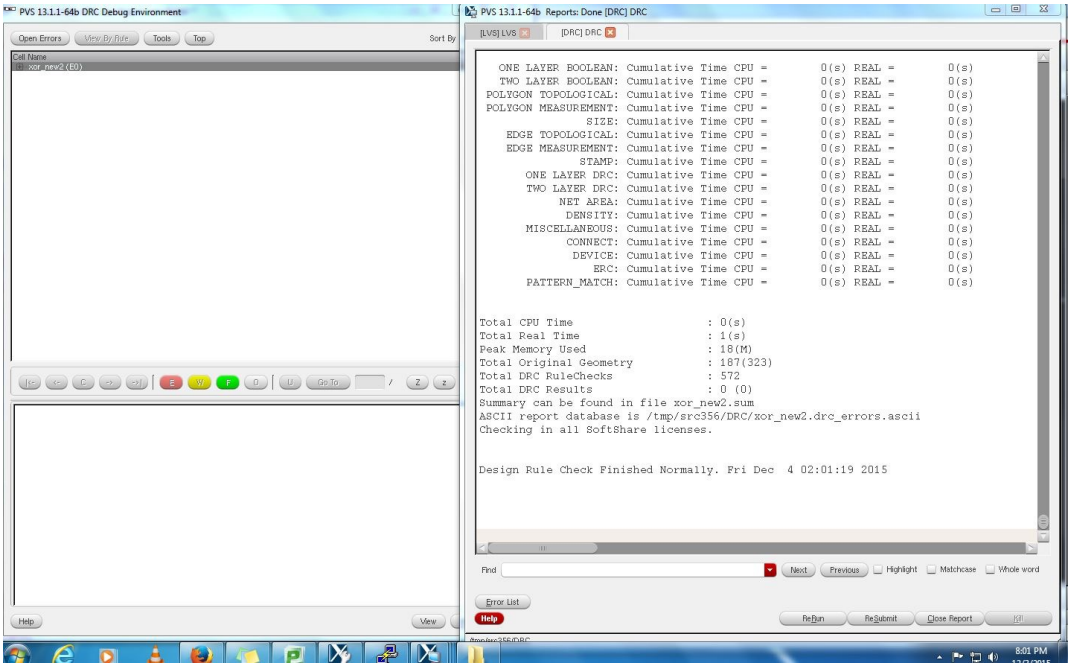
Previously XOR gate was implemented with NAND gates, the current XOR gate has been implemented with transmission gate logic to optimize for area . Six FETs are used here.



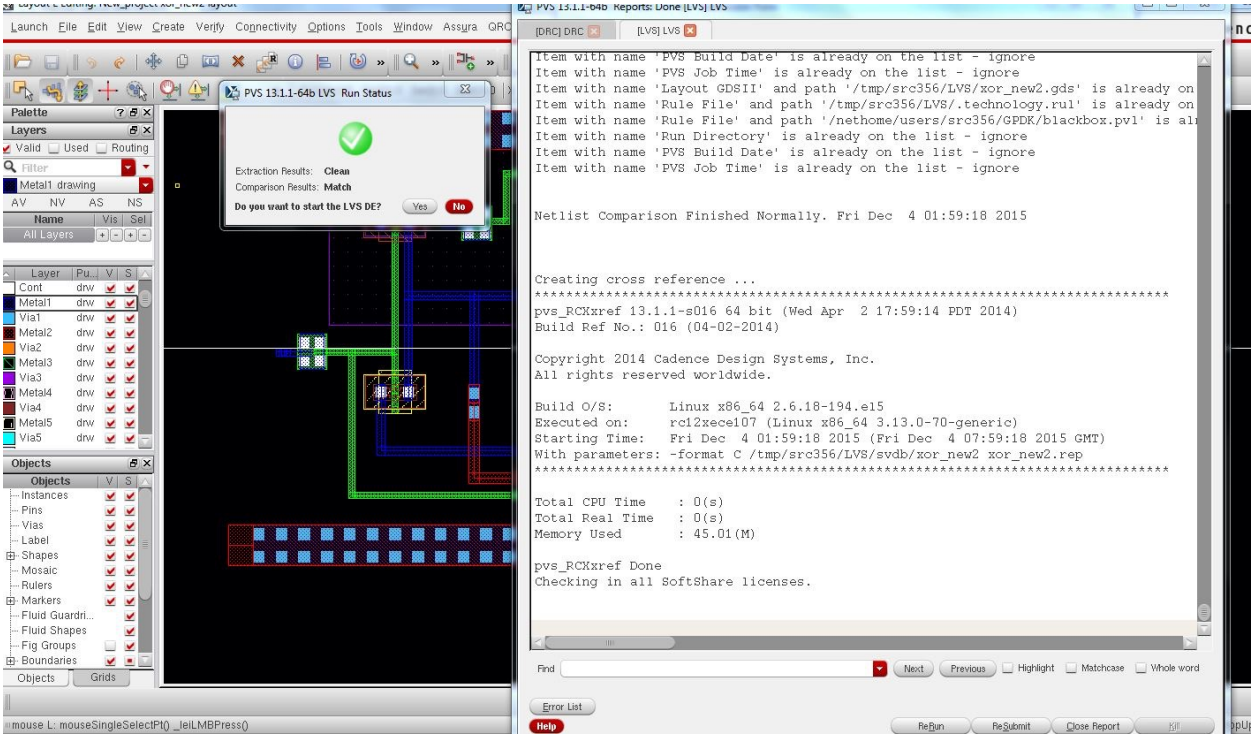
- Layout



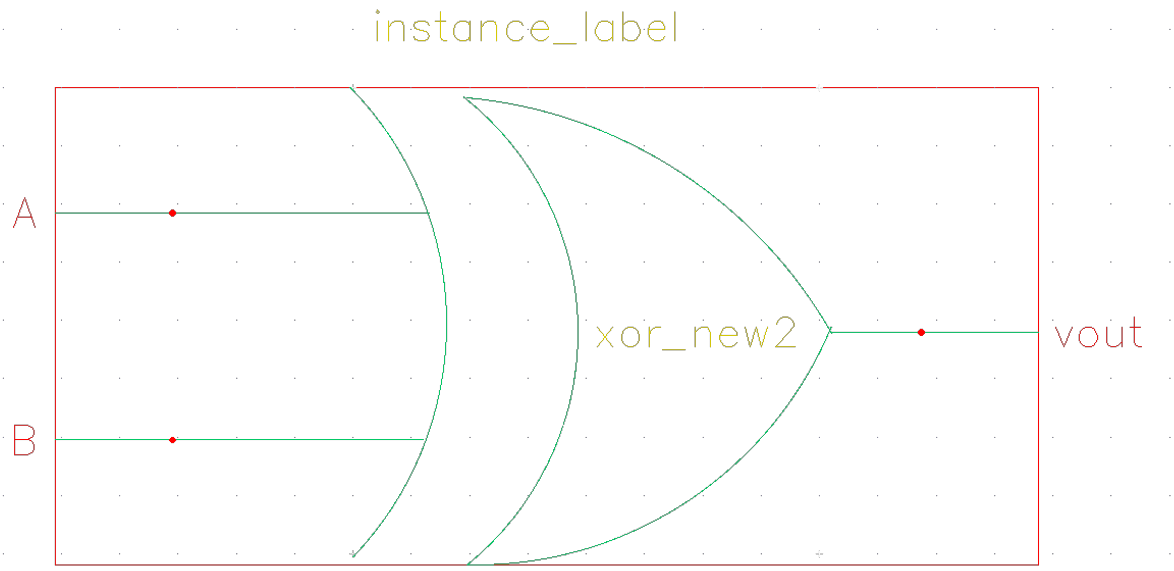
- DRC


PVS 13.1.1-64b DRC Debug Environment
PVS 13.1.1-64b Reports: Done [DRC] DRC
[LVS] LVS
[DRC] DRC
ONE LAYER BOOLEAN: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER BOOLEAN: Cumulative Time CPU = 0(s) REAL = 0(s)
POLYGON TOPOLOGICAL: Cumulative Time CPU = 0(s) REAL = 0(s)
POLYGON MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
SIZE: Cumulative Time CPU = 0(s) REAL = 0(s)
EDGE TOPOLOGICAL: Cumulative Time CPU = 0(s) REAL = 0(s)
EDGE MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
STAMP: Cumulative Time CPU = 0(s) REAL = 0(s)
ONE LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
NET AREA: Cumulative Time CPU = 0(s) REAL = 0(s)
DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
Total CPU Time : 0(s)
Total Real Time : 1(s)
Peak Memory Used : 18(M)
Total Original Geometry : 187(323)
Total DRC RuleChecks : 572
Total DRC Results : 0 (0)
Summary can be found in file xor_new2.sum
ASCII report database is /tmp/src356/DRC/xor_new2.drc_errors.ascii
Checking in all SoftShare licenses.
Design Rule Check Finished Normally. Fri Dec 4 02:01:19 2015
Find
Next
Previous
Highlight
Matchcase
Whole word
Error List
Help
ReRun
ReSubmit
Close Report
Kill

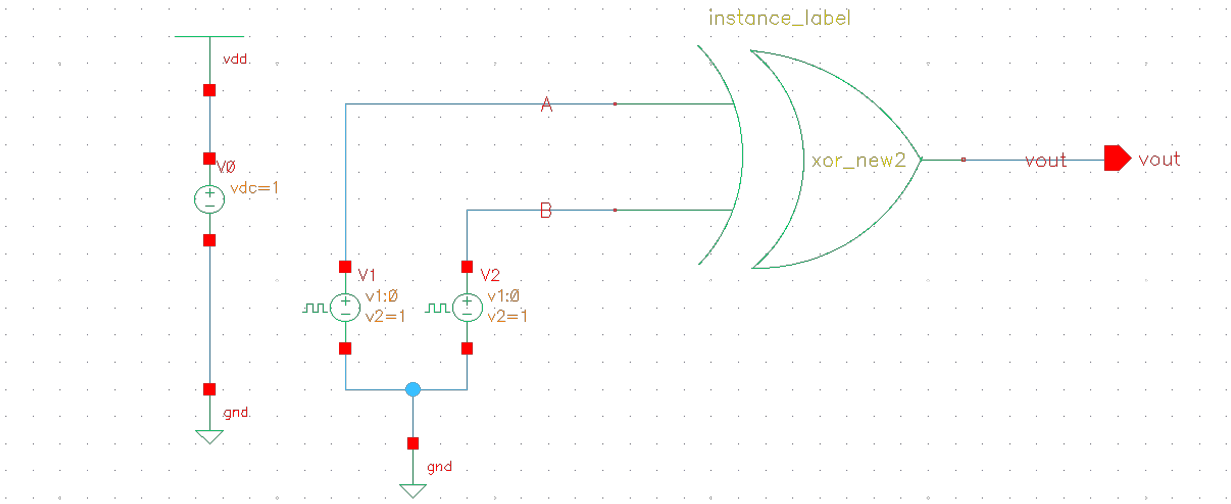
- LVS


Launch File Edit View Create Verify Connectivity Options Tools Window Assura
PVS 13.1.1-64b LVS Run Status
Extraction Results: Clean
Comparison Results: Match
Do you want to start the LVS DE?
Yes
No
Palette
Layers
Objects
[DRC] DRC
[LVS] LVS
Item with name 'PVS Build Date' is already on the list - ignore
Item with name 'PVS Job Time' is already on the list - ignore
Item with name 'Layout GDSII' and path '/tmp/src356/LVS/xor_new2.gds' is already on
Item with name 'Rule File' and path '/tmp/src356/LVS/.technology.rul' is already on
Item with name 'Rule File' and path '/nethome/users/src356/GPDK/blackbox.pvl' is al
Item with name 'Run Directory' is already on the list - ignore
Item with name 'PVS Build Date' is already on the list - ignore
Item with name 'PVS Job Time' is already on the list - ignore
Netlist Comparison Finished Normally. Fri Dec 4 01:59:18 2015
Creating cross reference ...
pvs_RCXxref 13.1.1-s016 64 bit (Wed Apr 2 17:59:14 PDT 2014)
Build Ref No.: 016 (04-02-2014)
Copyright 2014 Cadence Design Systems, Inc.
All rights reserved worldwide.
Build O/S: Linux x86_64 2.6.18-194.el5
Executed on: rc12xece107 (Linux x86_64 3.13.0-70-generic)
Starting Time: Fri Dec 4 01:59:18 2015 (Fri Dec 4 07:59:18 2015 GMT)
With parameters: -format C /tmp/src356/LVS/svdb/xor_new2 xor_new2.rep
Total CPU Time : 0(s)
Total Real Time : 0(s)
Memory Used : 45.01(M)
pvs_RCXxref Done
Checking in all SoftShare licenses.
Find
Next
Previous
Highlight
Matchcase
Whole word
Error List
Help
ReRun
ReSubmit
Close Report
Kill

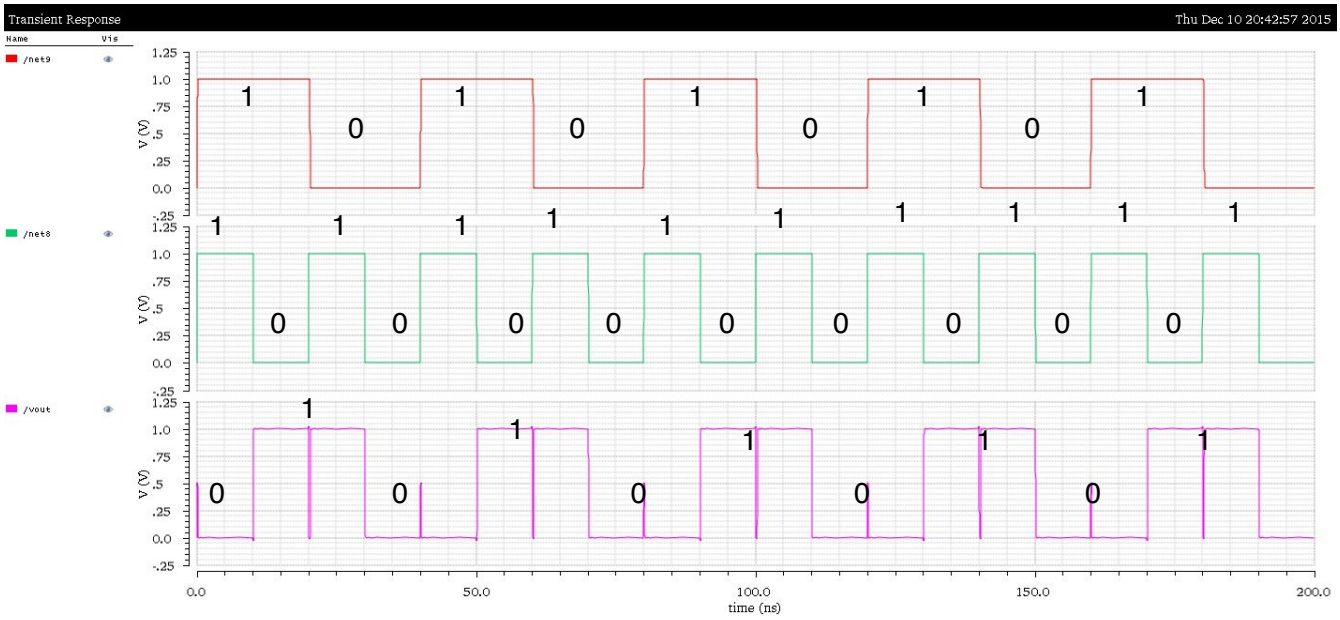
- Symbol
  -



Test bench Schematic

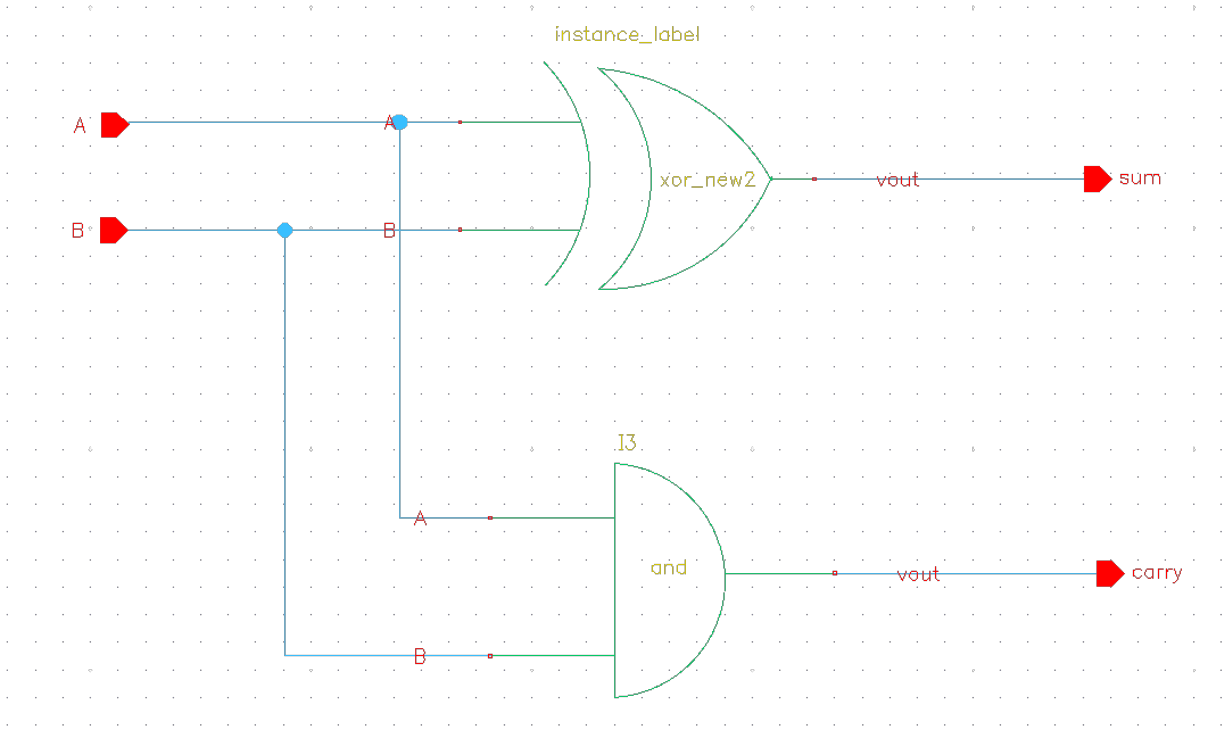


- Output Simulations

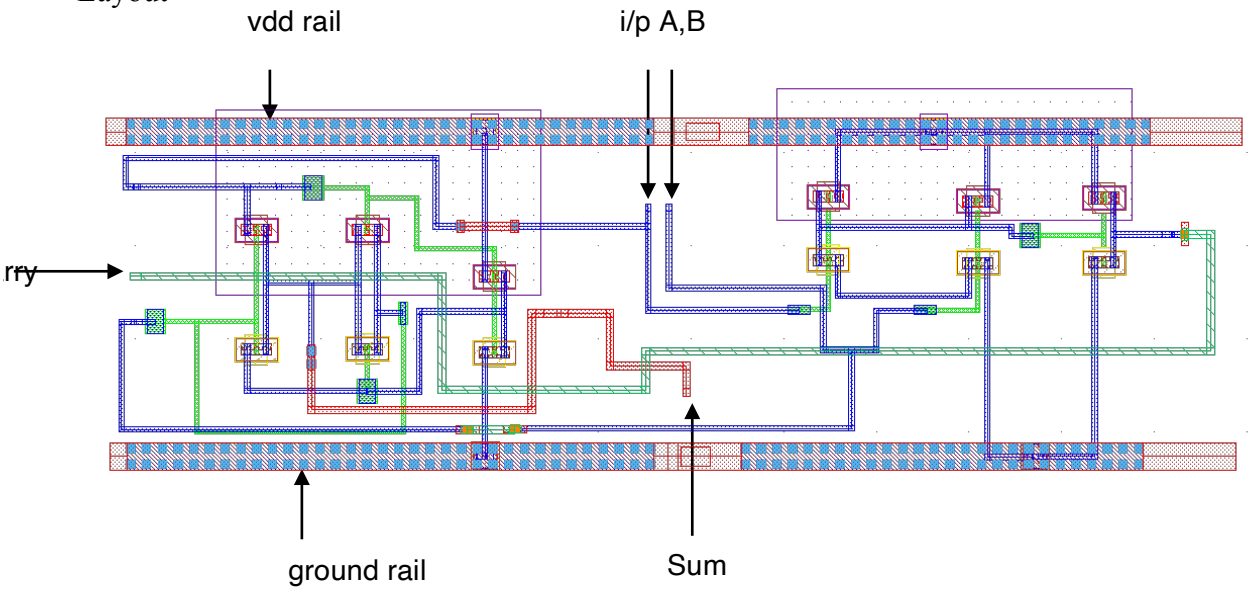


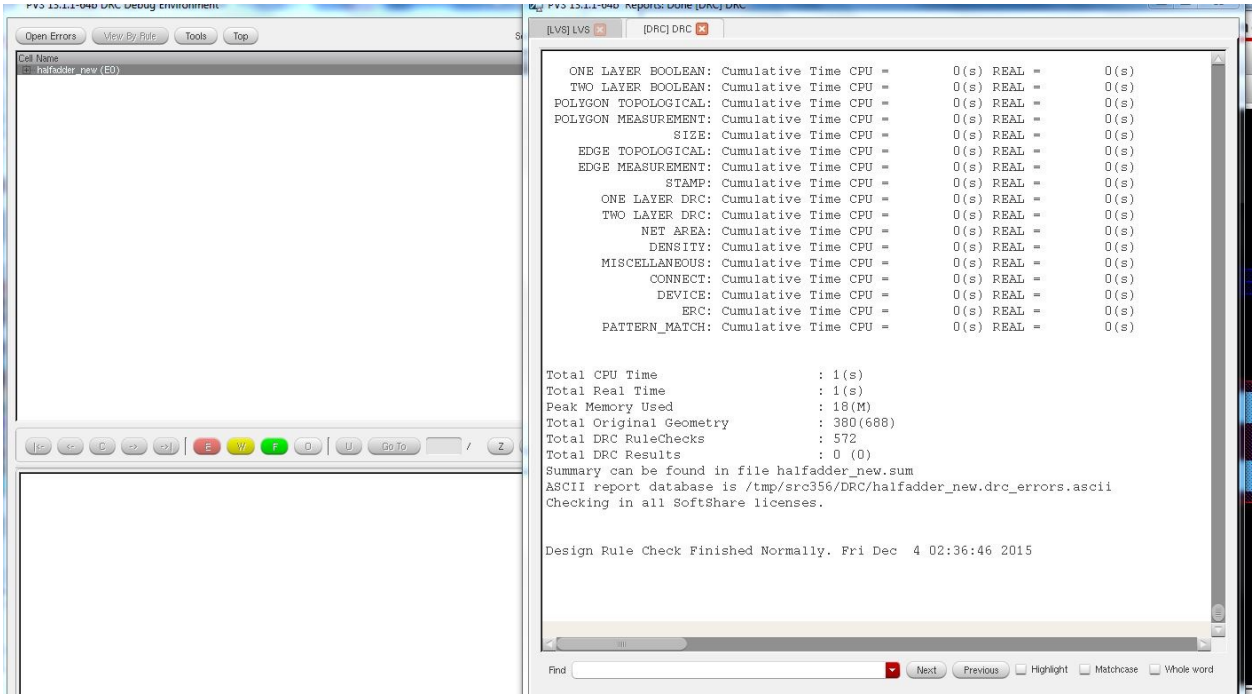
Half Adder:

- Schematic

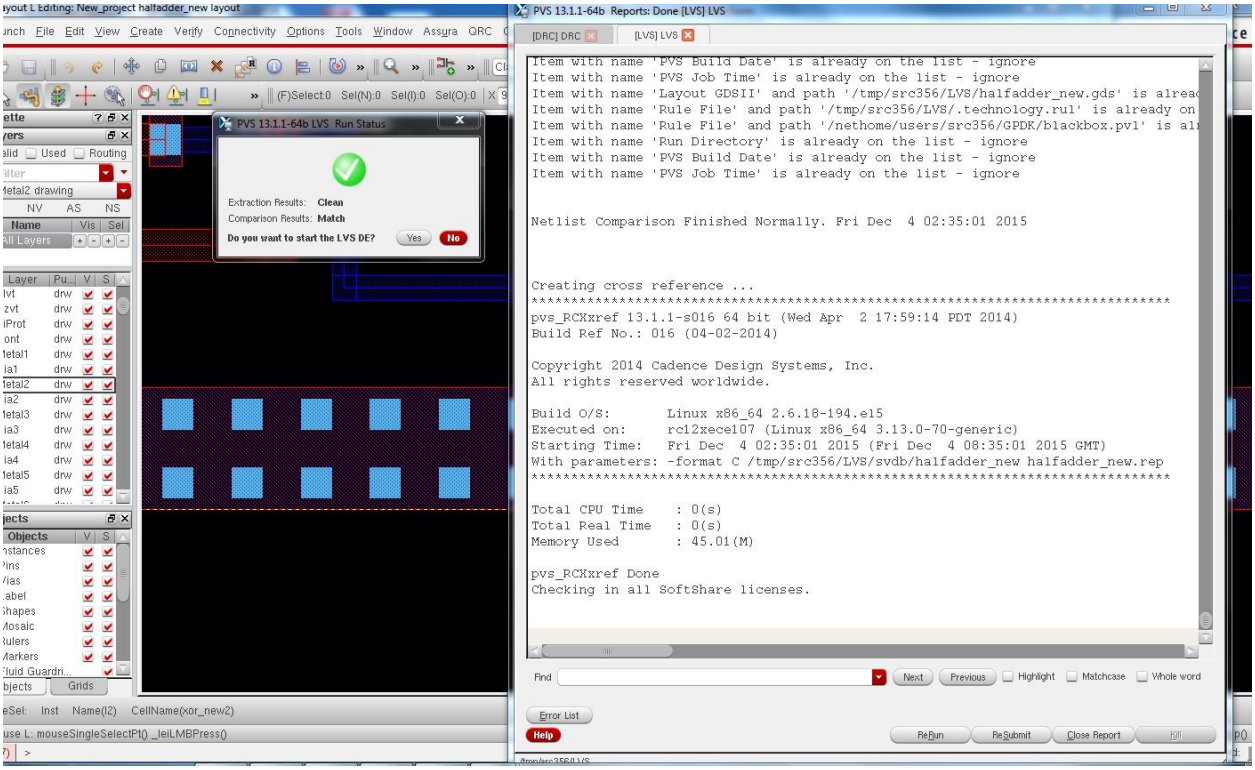


- Layout



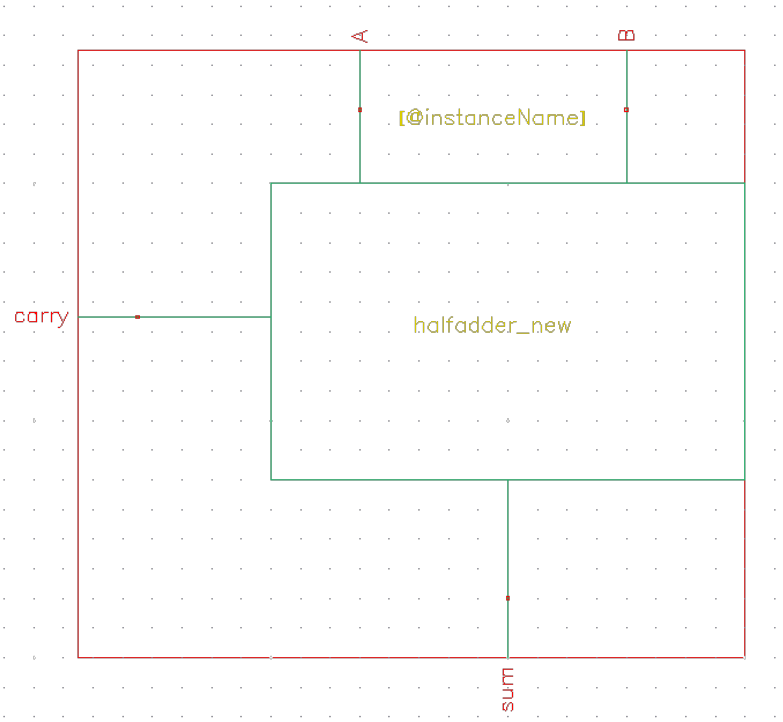

Open Errors
View By Rule
Tools
Top
Cell Name
halfadder_new (EO)
[LVS] LVS
[DRC] DRC
ONE LAYER BOOLEAN: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER BOOLEAN: Cumulative Time CPU = 0(s) REAL = 0(s)
POLYGON TOPOLOGICAL: Cumulative Time CPU = 0(s) REAL = 0(s)
POLYGON MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
SIZE: Cumulative Time CPU = 0(s) REAL = 0(s)
EDGE TOPOLOGICAL: Cumulative Time CPU = 0(s) REAL = 0(s)
EDGE MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
STAMP: Cumulative Time CPU = 0(s) REAL = 0(s)
ONE LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
NET AREA: Cumulative Time CPU = 0(s) REAL = 0(s)
DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
Total CPU Time : 1(s)
Total Real Time : 1(s)
Peak Memory Used : 18(M)
Total Original Geometry : 380(688)
Total DRC RuleChecks : 572
Total DRC Results : 0 (0)
Summary can be found in file halfadder_new.sum
ASCII report database is /tmp/src356/DRC/halfadder_new.drc_errors.ascii
Checking in all SoftShare licenses.
Design Rule Check Finished Normally. Fri Dec 4 02:36:46 2015
Find
Next
Previous
Highlight
Matchcase
Whole word

- DRC

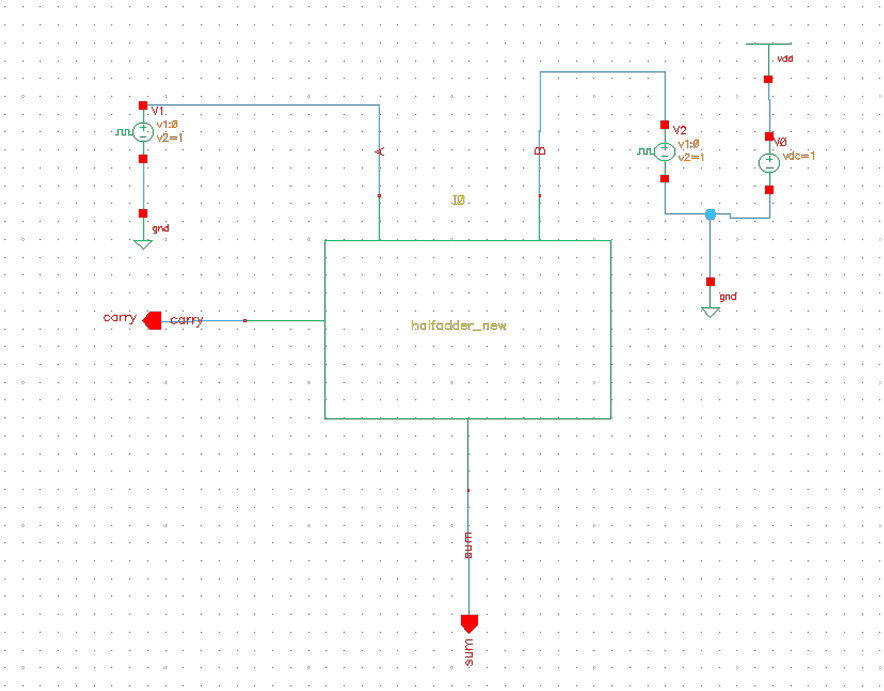


- LVS

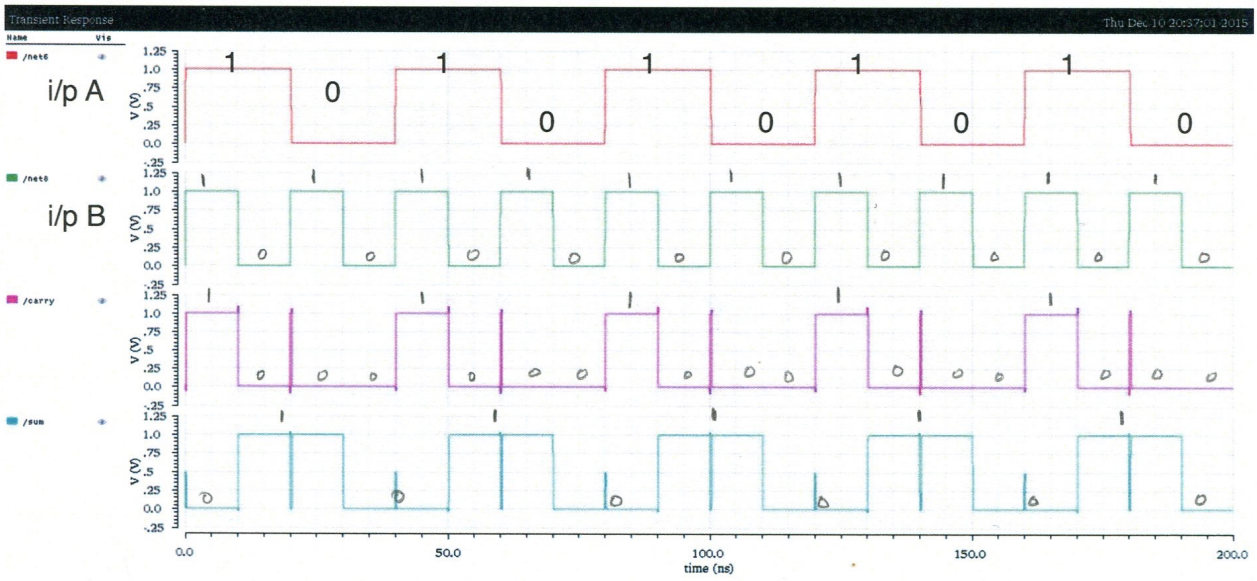
- Symbol



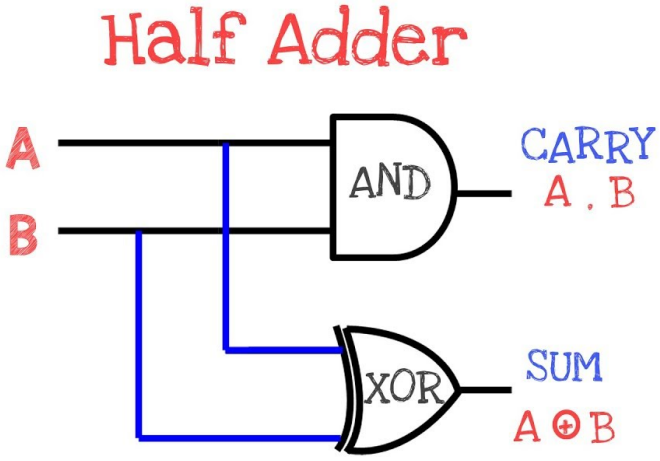
- Test bench schematic


V1
v1=0
v2=1
gnd
A
B
10
V2
v1=0
v2=1
vdd
V0
vdc=1
gnd
carry
carry
halfadder_new
sum
sum

- Output Simulations



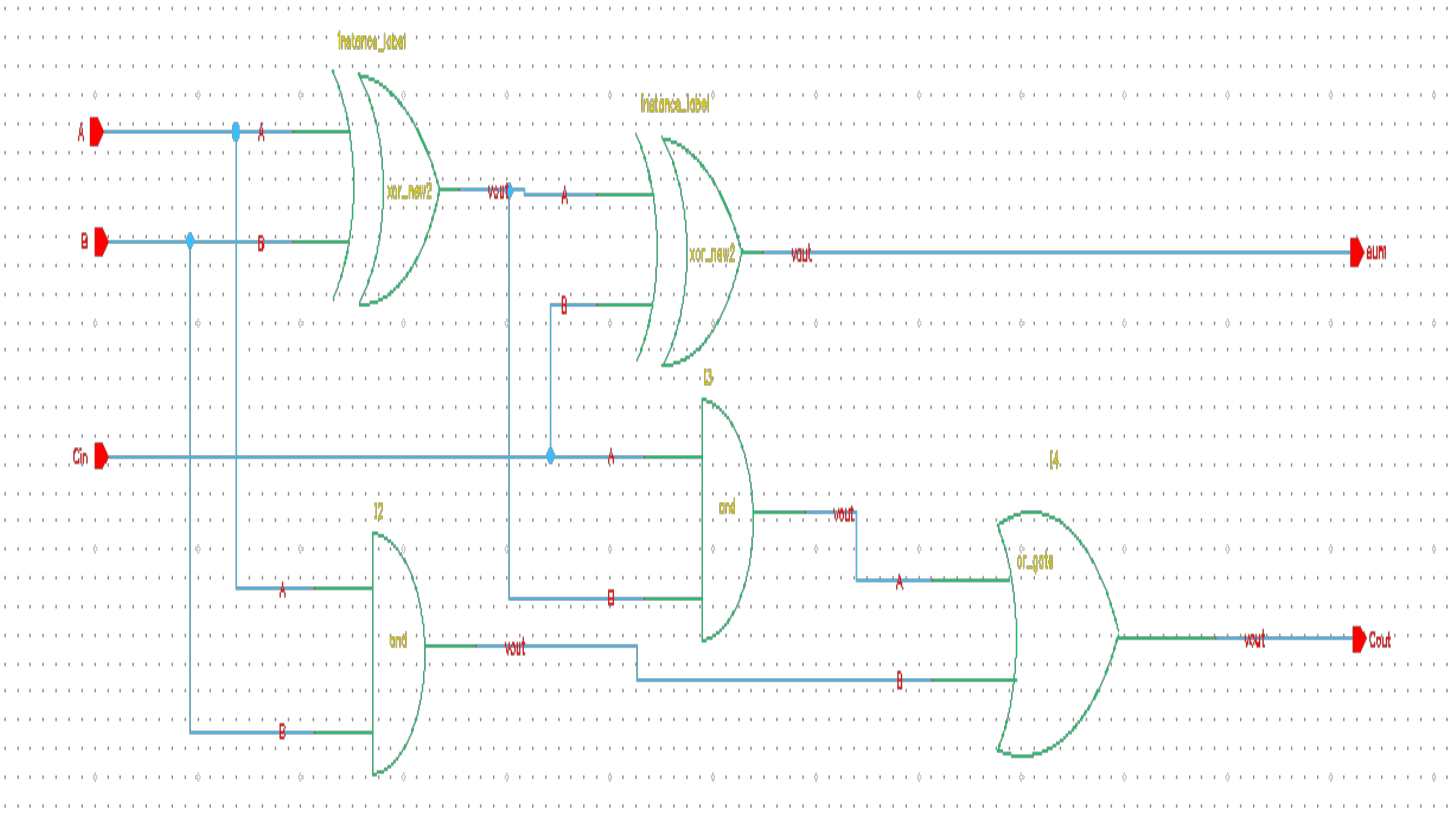
As observed from the graph the outputs 'sum' and 'carry' follow the logic function of a half-adder.



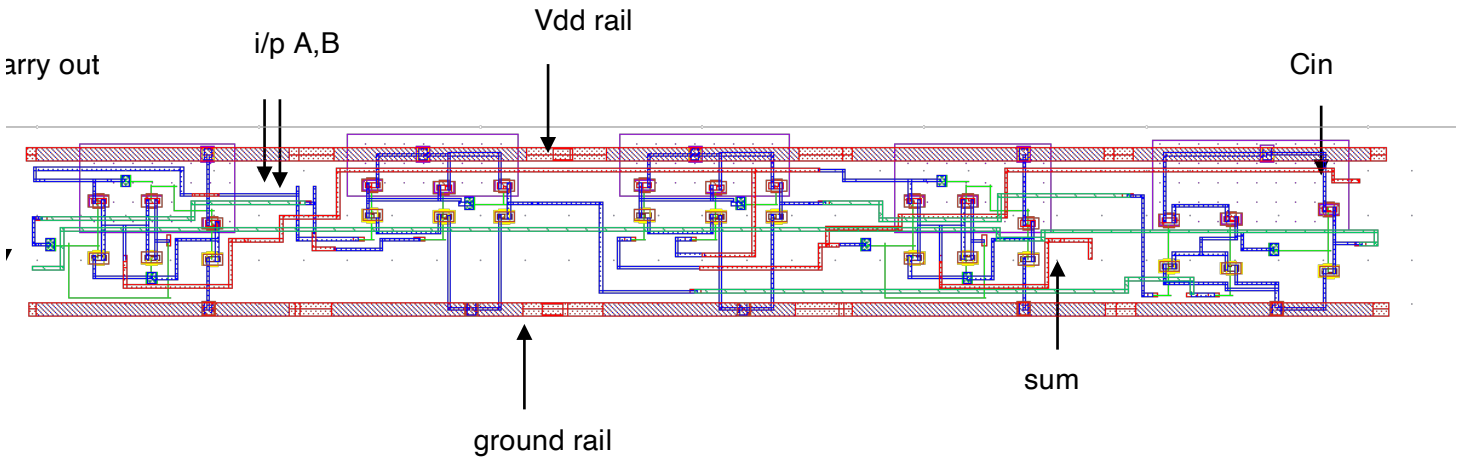
Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder:

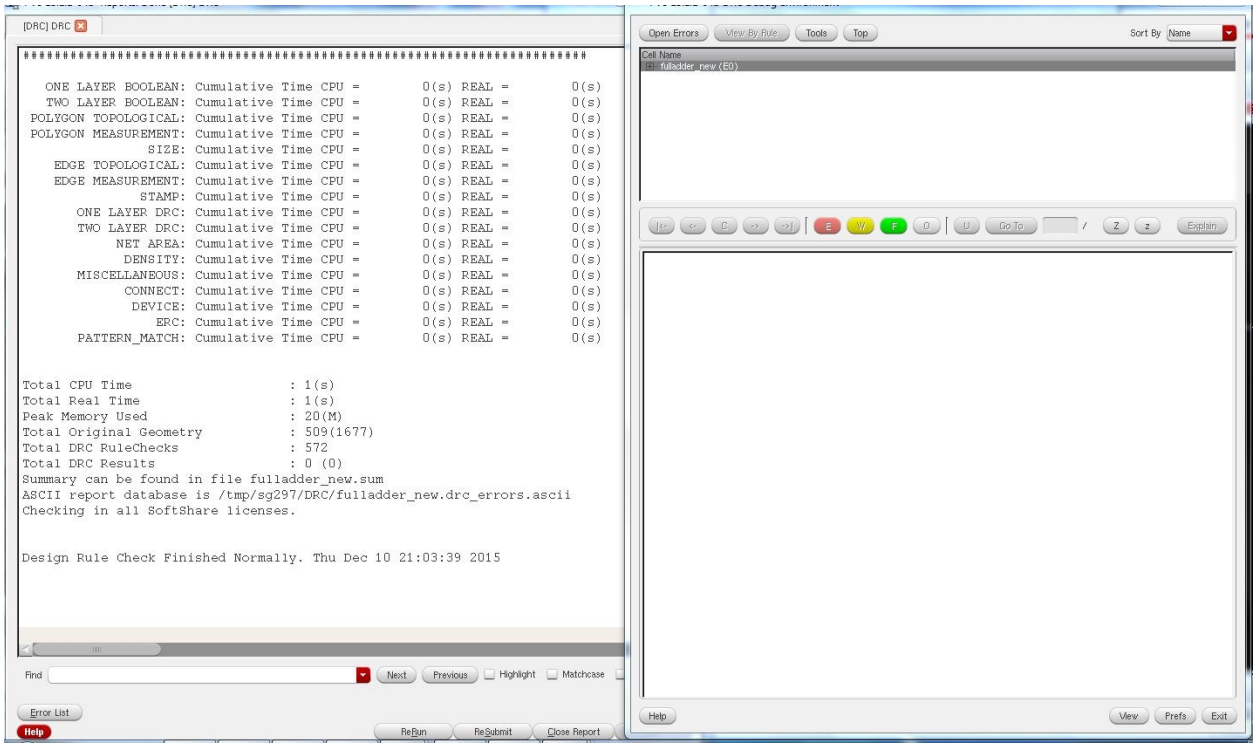
- Schematic



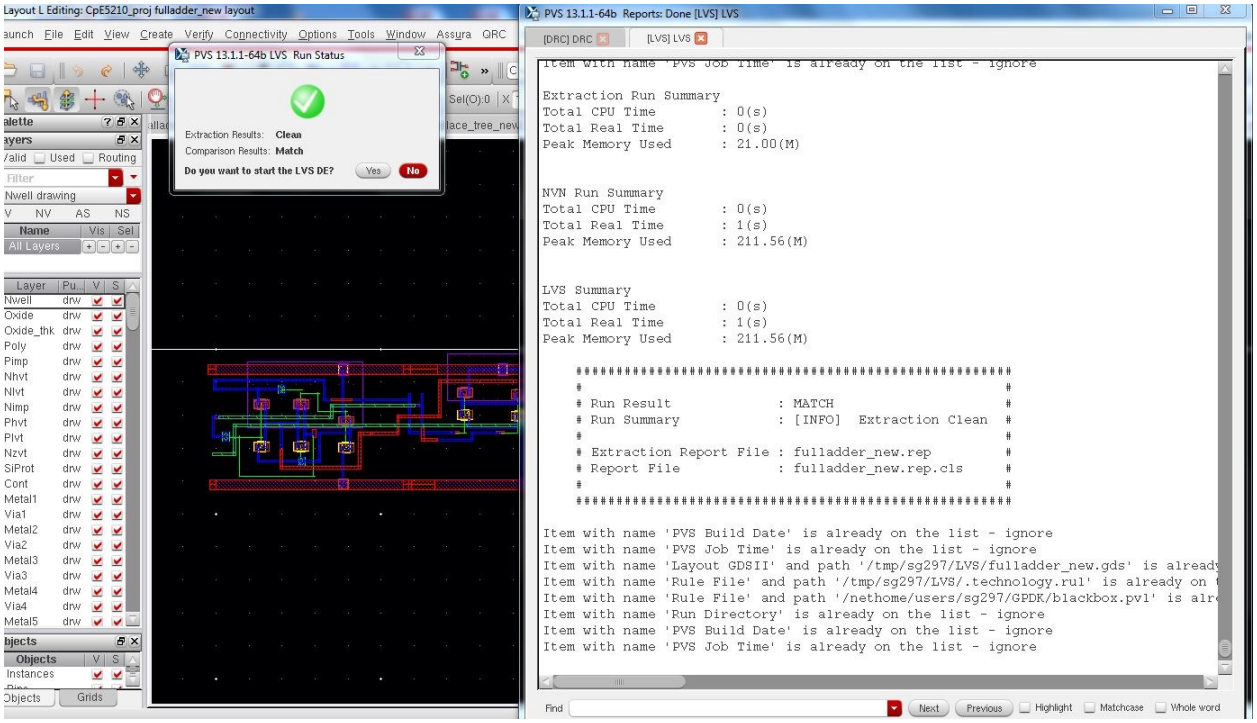
- Layout



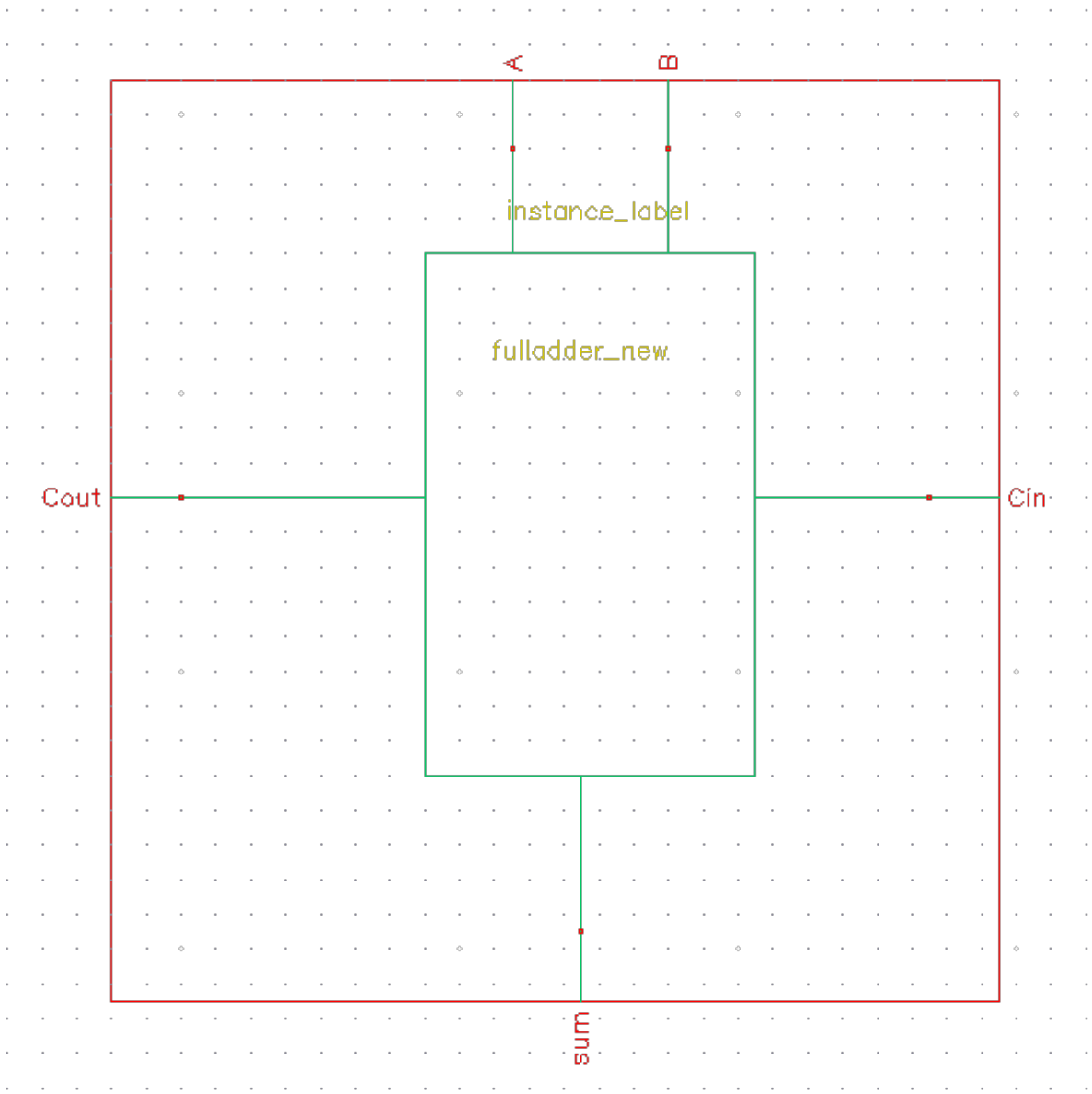
- DRC


[DRC] DRC
ONE LAYER BOOLEAN: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER BOOLEAN: Cumulative Time CPU = 0(s) REAL = 0(s)
POLYGON TOPOLOGICAL: Cumulative Time CPU = 0(s) REAL = 0(s)
POLYGON MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
SIZE: Cumulative Time CPU = 0(s) REAL = 0(s)
EDGE TOPOLOGICAL: Cumulative Time CPU = 0(s) REAL = 0(s)
EDGE MEASUREMENT: Cumulative Time CPU = 0(s) REAL = 0(s)
STAMP: Cumulative Time CPU = 0(s) REAL = 0(s)
ONE LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
TWO LAYER DRC: Cumulative Time CPU = 0(s) REAL = 0(s)
NET AREA: Cumulative Time CPU = 0(s) REAL = 0(s)
DENSITY: Cumulative Time CPU = 0(s) REAL = 0(s)
MISCELLANEOUS: Cumulative Time CPU = 0(s) REAL = 0(s)
CONNECT: Cumulative Time CPU = 0(s) REAL = 0(s)
DEVICE: Cumulative Time CPU = 0(s) REAL = 0(s)
ERC: Cumulative Time CPU = 0(s) REAL = 0(s)
PATTERN_MATCH: Cumulative Time CPU = 0(s) REAL = 0(s)
Total CPU Time : 1(s)
Total Real Time : 1(s)
Peak Memory Used : 20(M)
Total Original Geometry : 509(1677)
Total DRC RuleChecks : 572
Total DRC Results : 0 (0)
Summary can be found in file fulladder_new.sum
ASCII report database is /tmp/sg297/DRC/fulladder_new.drc_errors.ascii
Checking in all SoftShare licenses.
Design Rule Check Finished Normally. Thu Dec 10 21:03:39 2015
Open Errors
View By Rule
Tools
Top
Sort By
Name
Cell Name
fulladder_new (E0)
Go To
Explain
Find
Next
Previous
Highlight
Matchcase
Error List
Help
ReRun
ReSubmit
Close Report
View
Prefs
Exit

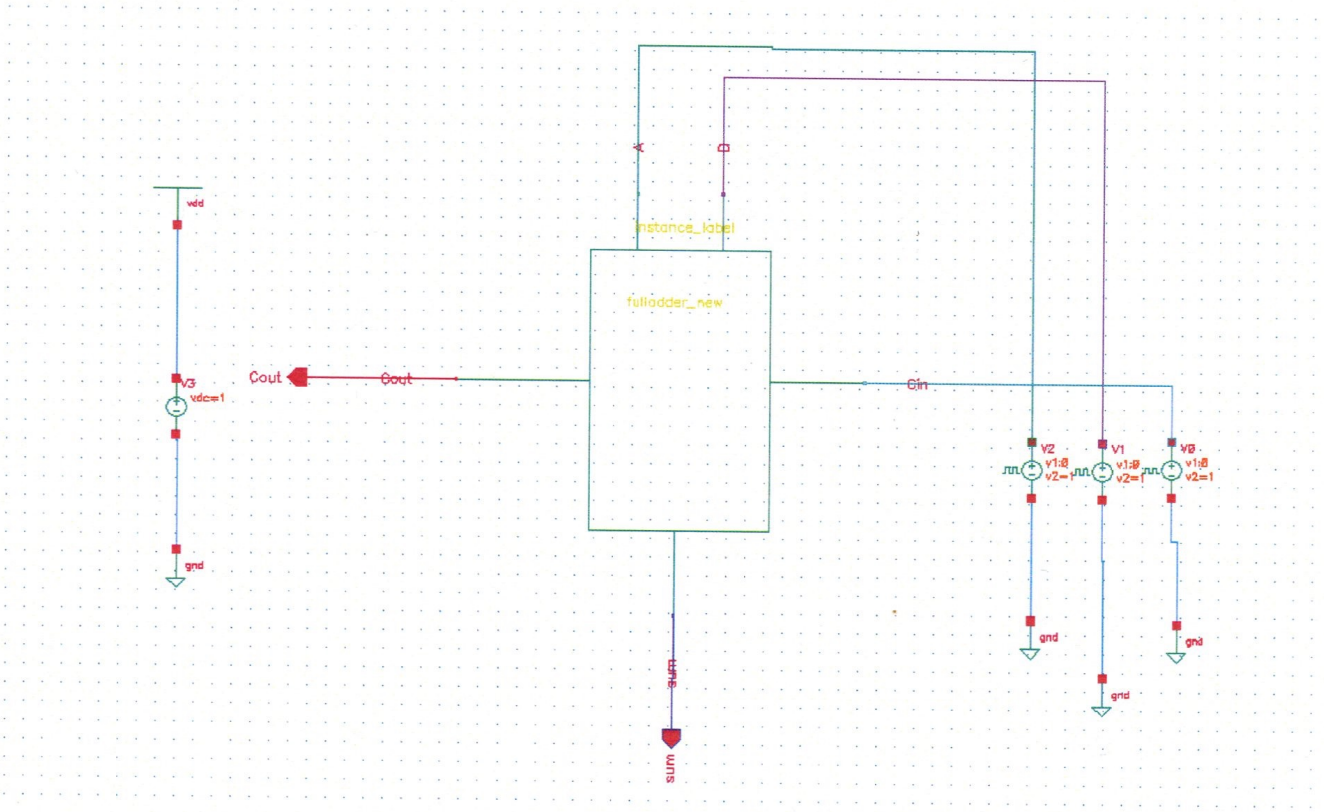
- LVS


Layout L Editing: CpE5210_proj fulladder_new layout
PVS 13.1.1-64b LVS Run Status
Extraction Results: Clean
Comparison Results: Match
Do you want to start the LVS DE?
Yes
No
PVS 13.1.1-64b Reports: Done [LVS] LVS
[DRC] DRC
[LVS] LVS
Item with name 'PVS Job Time' is already on the list - ignore
Extraction Run Summary
Total CPU Time : 0(s)
Total Real Time : 0(s)
Peak Memory Used : 21.00(M)
NVN Run Summary
Total CPU Time : 0(s)
Total Real Time : 1(s)
Peak Memory Used : 211.56(M)
LVS Summary
Total CPU Time : 0(s)
Total Real Time : 1(s)
Peak Memory Used : 211.56(M)
Run Result : MATCH
Run Summary : [INFO] Extraction Clean
Extraction Report File : fulladder_new.rep
Report File : fulladder_new.rep.cls
Item with name 'PVS Build Date' is already on the list - ignore
Item with name 'PVS Job Time' is already on the list - ignore
Item with name 'Layout GDSII' and path '/tmp/sg297/LVS/fulladder_new.gds' is already
Item with name 'Rule File' and path '/tmp/sg297/LVS/.technology.rul' is already on
Item with name 'Rule File' and path '/nethome/users/sg297/GPDK/blackbox.pvl' is alr
Item with name 'Run Directory' is already on the list - ignore
Item with name 'PVS Build Date' is already on the list - ignore
Item with name 'PVS Job Time' is already on the list - ignore
Find
Next
Previous
Highlight
Matchcase
Whole word

- Symbol

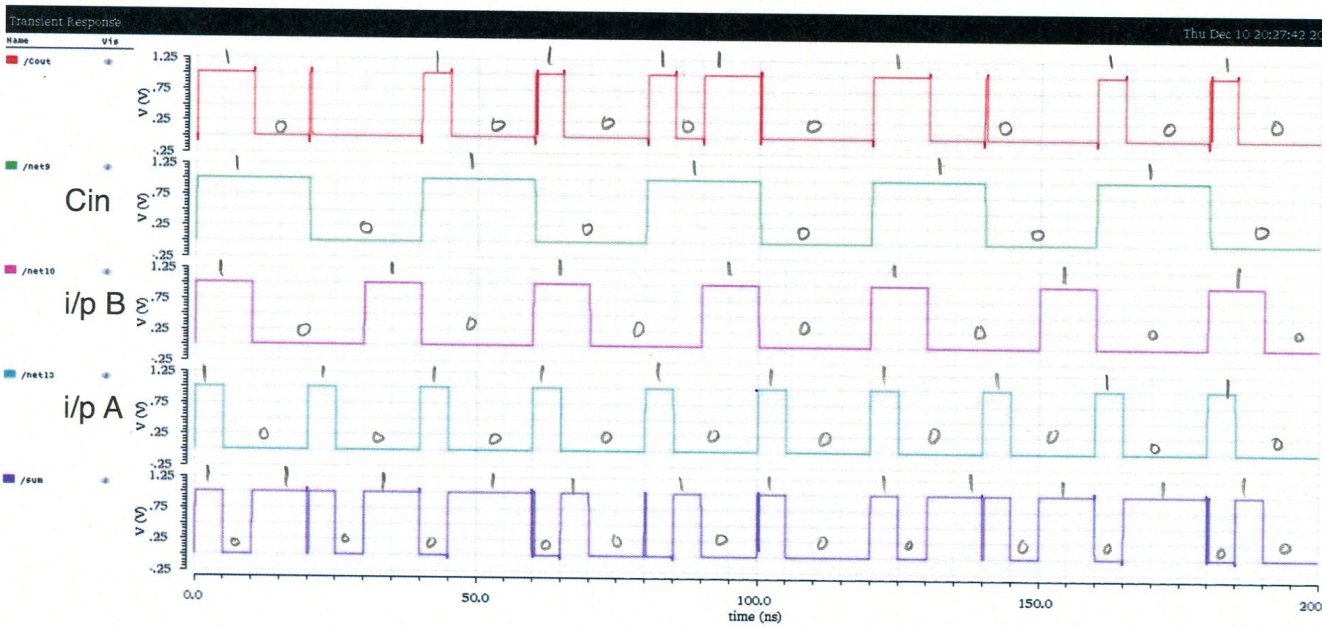


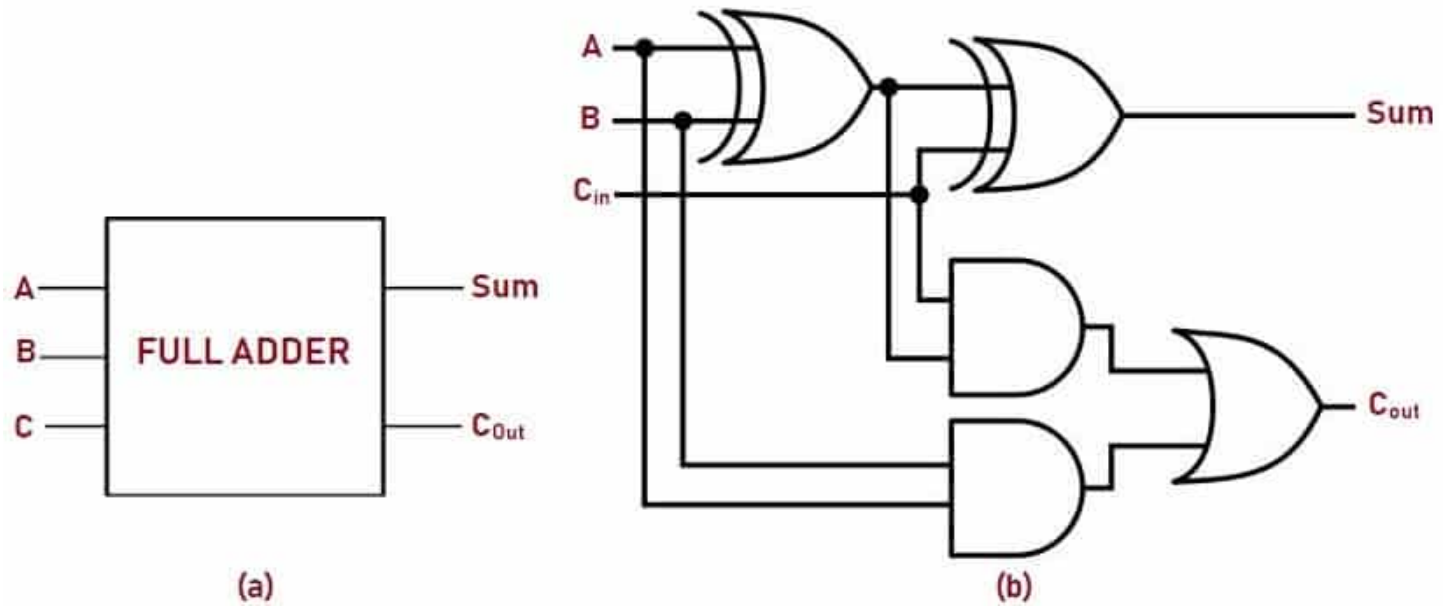
- Test bench Schematic



- Output Simulations

As observed from the graph outputs 'sum' and 'carry' follow the logic function of full-adder.





d Full Adder

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + (A \oplus B)\, C_{in}$$

Inputs			Outputs	
A	B	C_{in}	Sum	Carry
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