Project 1 ECE 45600 - 8-bit Manchester Adder

April 8, 2025

Abstract

This is the report for project 1 for ECE456. The goal is to design an 8-bit Manchester adder based on static logic.

Contents

1 Introduction

The reason why a Manchester Carry Adder is faster than other adders is it's optimized carry chain. That is, it calculates the carry bits as fast as possible. This is because the critical path of the circuit will always be primarily affected by the delay of the carry.
The goal of this project is to design a Manchester Carry Adder for 8-bits based on Static Logic.

2 Design Strategy

A single bit adder takes the two bits to be added (A_i, B_i), as well as the carry-in bit (C_{in}) to calculate the sum bit (S) and the carry-out bit (C_{out}). In static logic, this can be done using three signals:

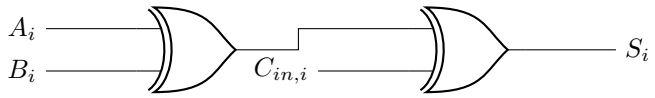
- Generate (g_i): $g_i = A_i \cdot B_i = A_i \bar{+} B_i$
- Delete (d_i): $d_i = \bar{A}_i \cdot \bar{B}_i$
- Propagate (p_i): $p_i = A_i \oplus B_i$

2.1 Sum Bit

The sum bit, colloquially calculated by xor-ing all the inputs, thus, in this design, it was calculated as follows:

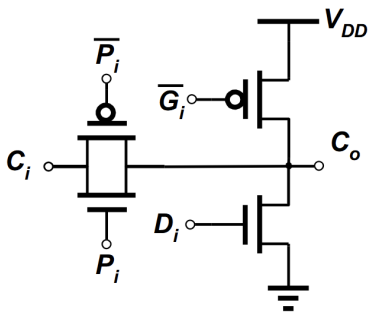
$$S_i = p_i \oplus C_{in,i}$$

which, in circuit form is as follows:



2.2 Carry-Out Bit

The carry-out bit is a little more complex.
In static logic, the following circuit is used to generate the bit:



p_i is generated using an XOR gate. Normally g_i would require an AND gate, but, since we are using $\bar{g}_i$, we can use a NAND gate instead. Finally, d_i is generated using a NOR gate.

2.3 Full Cell

The full cell is composed of a carry cell and a sum cell, which compute the corresponding values. The output of the carry cell connects to the input of the next cell, if available.

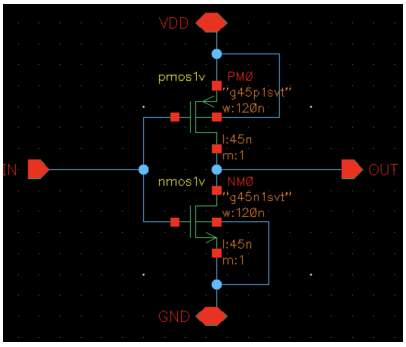
3 Circuit

We will present the design of the circuit from bottom to top.

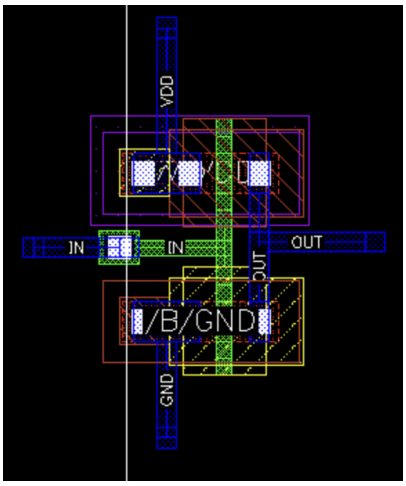
3.1 Logic Gates

3.1.1 Inverter

The following is the schematic of the inverter gate, as designed in Cadence Virtuoso:

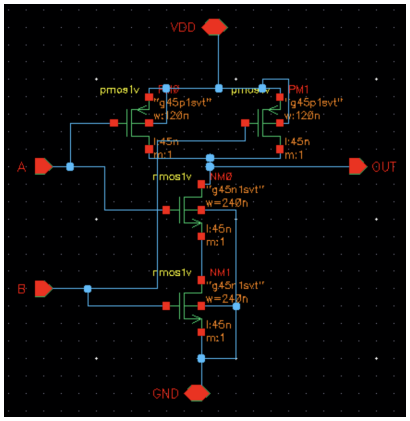


and the layout:

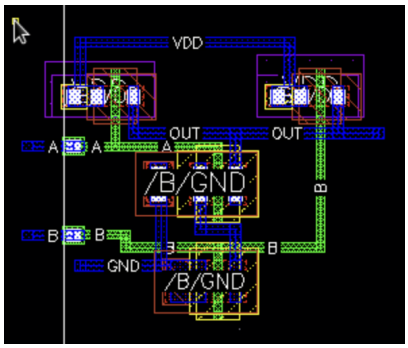


3.1.2 NAND

The following is the schematic of the NAND gate, as designed in Cadence Virtuoso:

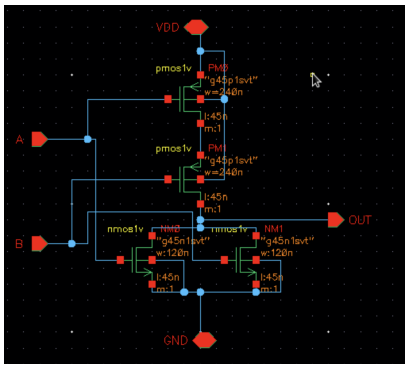


and the layout:

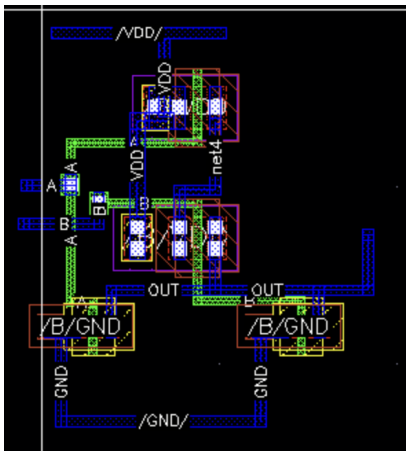


3.1.3 NOR

The following is the schematic of the NOR gate, as designed in Cadence Virtuoso:

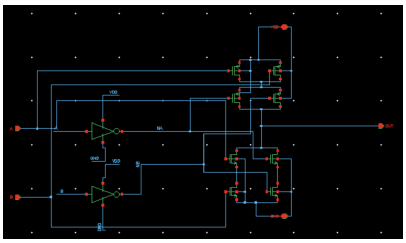


and the layout:

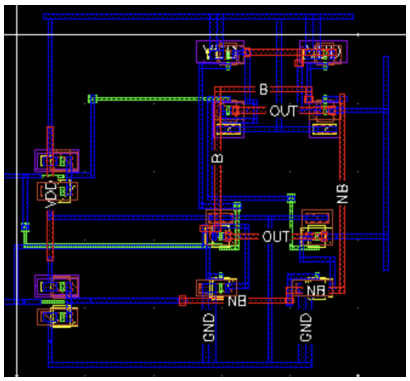


3.1.4 XOR

The following is the schematic of the XOR gate, as designed in Cadence Virtuoso:



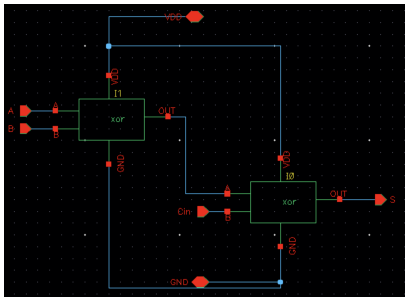
and the layout:



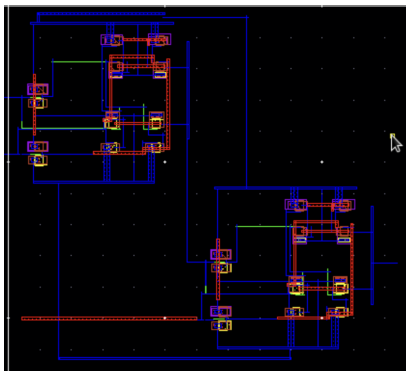
3.2 Sub-Cells

3.2.1 Sum Cell

The following is the schematic of the Sum Cell, as designed in Cadence Virtuoso:

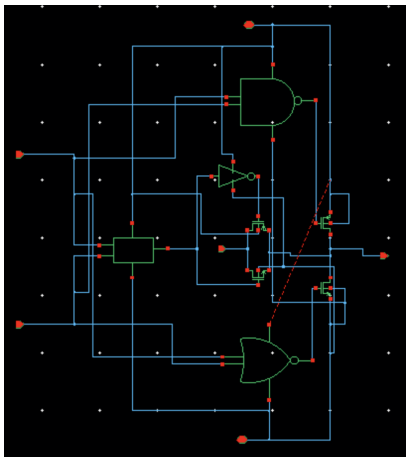


and the layout:

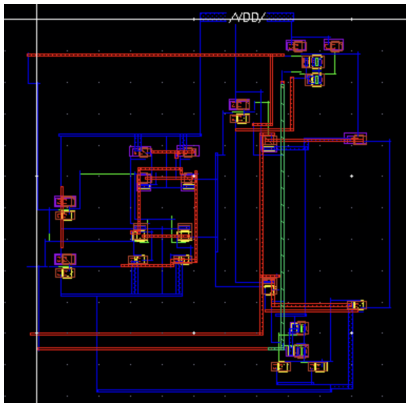


3.2.2 Carry Cell

The following is the schematic of the Sum Cell, as designed in Cadence Virtuoso:

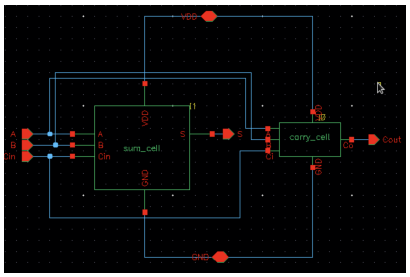


and the layout:

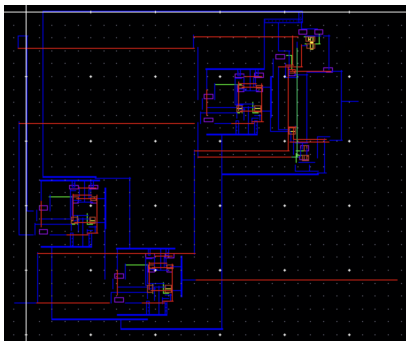


3.3 1-Bit Adder

The following is the schematic of the Sum Cell, as designed in Cadence Virtuoso:



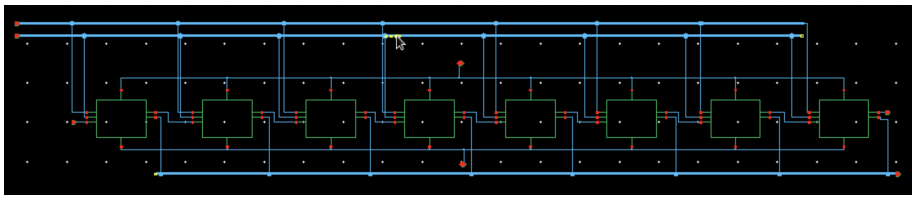
and the layout:



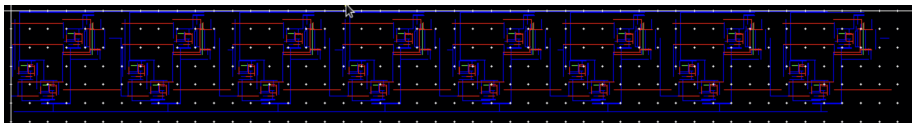
it's verification can be found in the DRC and LVS section.

3.4 8-Bit Adder

The following is the schematic of the Sum Cell, as designed in Cadence Virtuoso:



and the layout:



it's verification can be found in the DRC and LVS section.

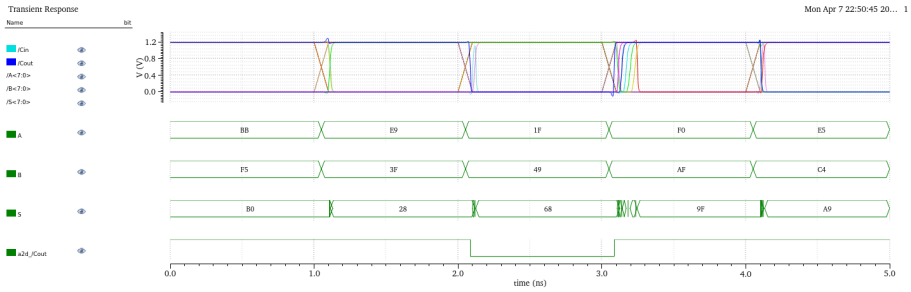
4 Results

The inputs to testbench, and their expected outputs are the following:

A	B	S	C_{out}
BB	F5	B0	1
E9	3F	28	1
1F	49	68	0
F0	AF	9F	1
E5	C4	A9	1

4.1 No-Parasitics

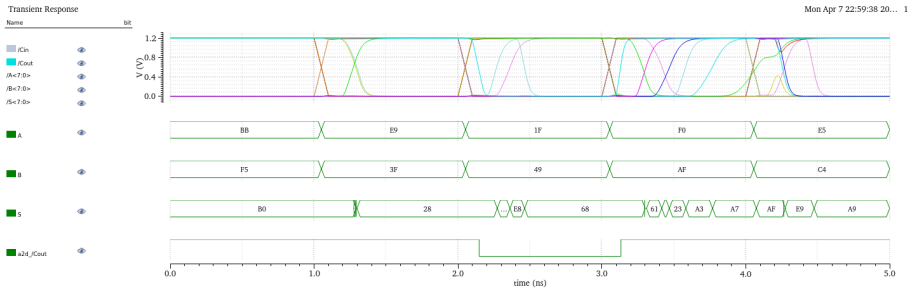
The pre-layout, at 1GHz results are the following:



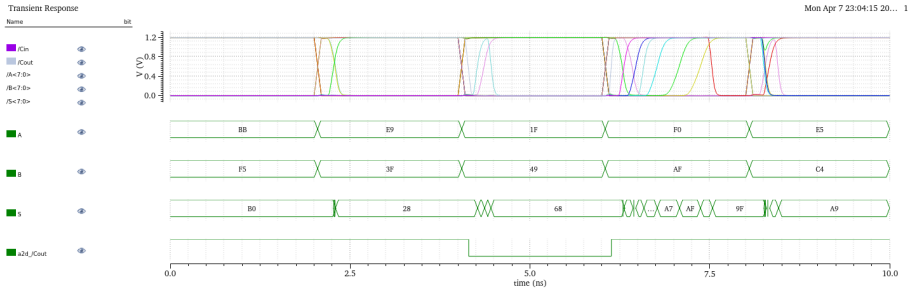
As can be seen, the results are what we expected, proving that the adder works as expected.

4.2 Parasitics

The post-layout, at 1GHz results are the following:



However, this time, it can be seen that in one of the transitions, the desired value is not achieved, that is, it takes too long for the value to evaluate.
When run at 500MHz, the plot is as following:



This time, the desired sum in the previously erroneous bit, was reached, albeit not for a long time.

5 Power Consumption

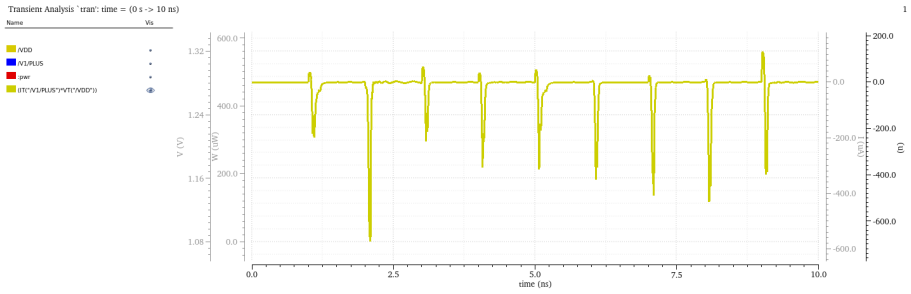
To test power consumption, the following inputs are provided:

A	B
CB	11
EE	33
11	00
1A	A3
C4	6B
1F	F1
55	AC
C3	E2
32	28
FF	FF

5.1 Pre-layout Power Consumption

After defining these as the inputs to the adder, two methods were used to calculate the power consumed:

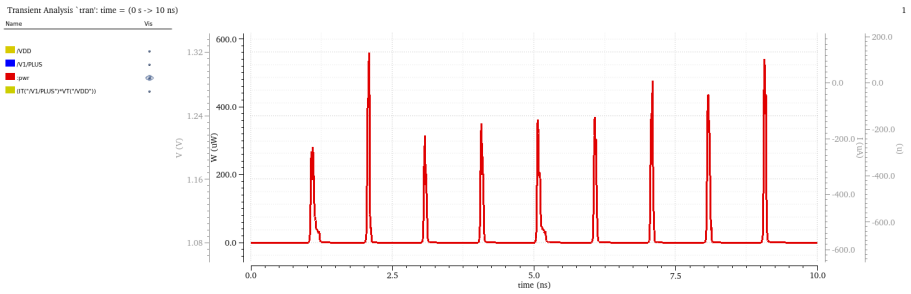
1. $I * V$ of the power supply, which resulted in the following plot:



and when the average was calculated using the calculator, we got a power consumed of $16.85\mu W$ as can be seen:

average(IT("/V1/...	-16.84997E-6

2. Power of the device, as generated by Cadence, which resulted in the following plot:



and when the average was calculated using the calculator, we got a power consumed of $18.165\mu W$ as can be seen:

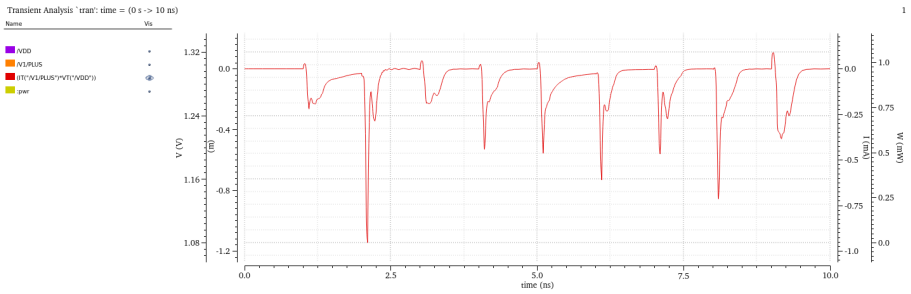
average(getData...	18.16500E-6

Both of these are reasonably similar estimations of the average power consumed.

5.2 Post-layout Power Consumption

Once more the two methods were used:

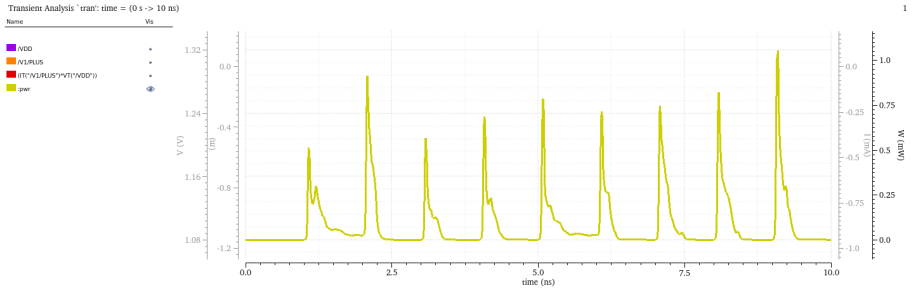
1. $I * V$ of the power supply, which resulted in the following plot:



and when the average was calculated using the calculator, we got a power consumed of $80.26\mu W$ as can be seen:

average(IT("/V1/...	-80.26E-6

2. Power of the device, as generated by Cadence, which resulted in the following plot:



and when the average was calculated using the calculator, we got a power consumed of $83.05\mu W$ as can be seen:

average(getData...	83.05E-6

Once more, both average power results are reasonably similar.

6 Comparisons

We will now compare the Manchester Adder with a few other adders:

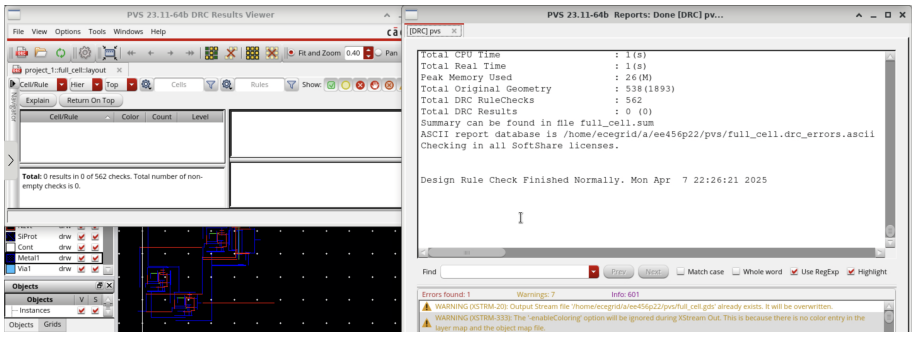
- Han-Carlson (HC) Adder:
  - Layout Complexity: Higher than Manchester, it is similar to KS, but with wiring optimizations.
  - Logic Depth: $\log_2(n)$ vs n in a Manchester adder. Notably lower than a Manchester adder.
  - Fan-Out: Fan-out of up to 2, as opposed to a Manchester adder which has a less consuming fan-out of 1.
  - Area: Since the wiring is simple and minimalistic, the area of a Manchester adder is smaller than that of the HC.
- Brent-Kung (BK) Adder:
  - Layout Complexity: it has a simpler layout than the Manchester adder.
  - Logic Depth: $2\log_2(n)$ vs n in a Manchester adder. Notably lower than a Manchester adder, but still higher than others.
  - Fan-Out: Fan-out of up to 2, as opposed to a Manchester adder which has a less consuming fan-out of 1.
  - Area: Smaller area than the Manchester adder.
- Ladner-Fischer (LF) Adder:

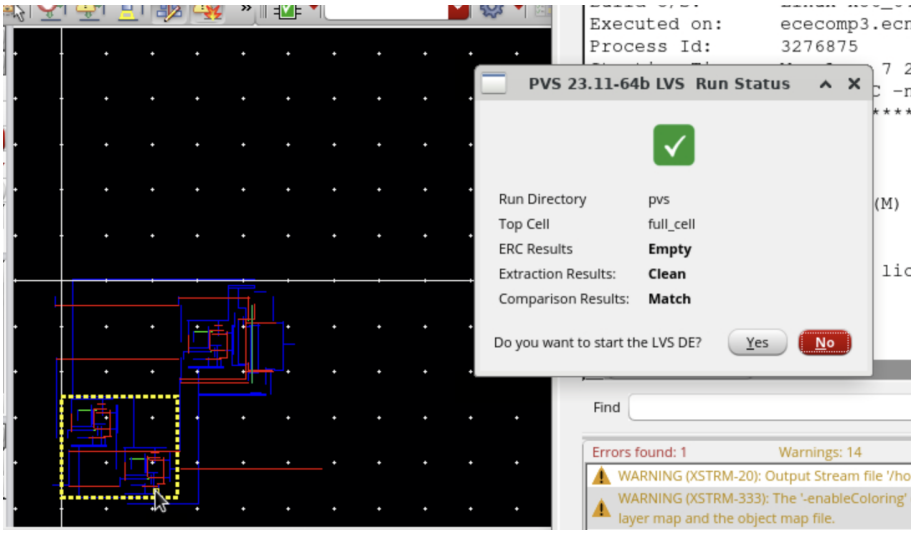
  - Layout Complexity: High Complexity, notably higher than the Manchester adder.
  - Logic Depth: it has a logic depth of $\log_2(n)$ which is much lower than a Manchester adder, which has a complexity of n.
  - Fan-Out: Can have fan-outs larger than 2, as opposed to the 1 of the Manchester adder.
  - Area: Since the layout is so complex, the area of the LF is considerably larger as well.

- Kogge-Stone (KS) Adder:
  - Layout Complexity: Quite high, when compared with the Manchester adder.
  - Logic Depth: it has a logic depth of $\log_2(n)$ which is much lower than a Manchester adder, which has a complexity of n.
  - Fan-Out: Both adders have similar fan-outs.
  - Area: Since the layout is so complex, the area of the KS is considerably larger as well.

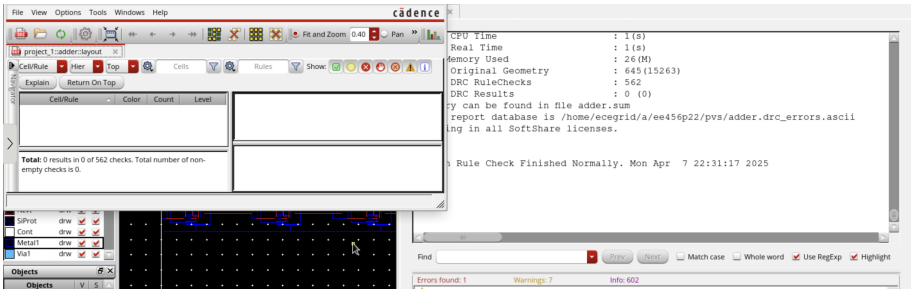
7 DRC and LVS

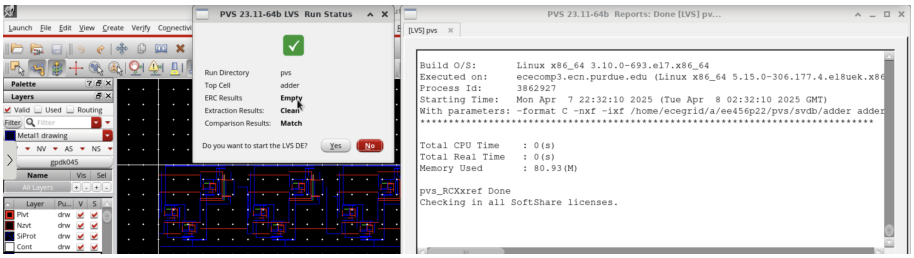
1-Bit Adder




PVS 23.11-64b LVS Run Status
Run Directory pvs
Top Cell full_cell
ERC Results Empty
Extraction Results: Clean
Comparison Results: Match
Do you want to start the LVS DE?
Yes
No
Executed on: ececomp3.ecr
Process Id: 3276875
Find
Errors found: 1
Warnings: 14
WARNING (XSTRM-20): Output Stream file '/ho
WARNING (XSTRM-333): The '-enableColoring'
layer map and the object map file.

8-Bit Adder


cadence
CPU Time : 1(s)
Real Time : 1(s)
Memory Used : 26(M)
Original Geometry : 645(15263)
DRC RuleChecks : 562
DRC Results : 0 (0)
report database is /home/ecegrid/a/ee456p22/pvs/adder.drc_errors.ascii
Rule Check Finished Normally. Mon Apr 7 22:31:17 2025


PVS 23.11-64b LVS Run Status
Run Directory pvs
Top Cell adder
ERC Results Empty
Extraction Results: Clean
Comparison Results: Match
Do you want to start the LVS DE?
Yes
No
Build O/S: Linux x86_64 3.10.0-693.el7.x86_64
Executed on: ececomp3.ecn.purdue.edu (Linux x86_64 5.15.0-306.177.4.el8uek.x86
Process Id: 3862927
Starting Time: Mon Apr 7 22:32:10 2025 (Tue Apr 8 02:32:10 2025 GMT)
With parameters: -format C -nxf -ixf /home/ecegrid/a/ee456p22/pvs/svdb/adder adder
Total CPU Time : 0(s)
Total Real Time : 0(s)
Memory Used : 80.93(M)
pvs_RCXxref Done
Checking in all SoftShare licenses.