8-bit Manchester Adder

MOS VLSI DESIGN (ECE 55900)

Carlet Pierrelouis

11/14/2024

Dr. Mohammadi

Abstract

This project presents the design of an 8-bit Manchester adder using transmission gate static logic for the accumulator section of a high-performance chip. The design emphasizes minimal propagation delay, high throughput, and efficient power dissipation. The methodology includes transistor-level implementation, layout creation, DRC/LVS checks, and functional validation through pre- and post-layout simulations. The performance is compared against Han-Carlson, Brent-Kung, Ladner-Fischer, and Kogge-Stone adders, referencing the papers of Knowles (1999) and Harris (2003). The results demonstrate the effectiveness of the transmission gate approach for high-speed arithmetic operations.

Introduction

High-speed and power-efficient adders are fundamental to the performance of modern digital systems. This project focuses on designing an 8-bit Manchester adder using transmission gate static logic for integration into Letni Corporation's accumulator. Transmission gate logic was chosen for its superior performance in minimizing delay and optimizing throughput, meeting the specified rise and fall times of 0.1 ns.

The design process involved transistor-level implementation, layout generation, and verification through DRC and LVS. Input functionality was validated using predefined hexadecimal samples, and power dissipation was analyzed pre- and post-layout. Performance comparisons with parallel prefix-tree adders, including Han-Carlson, Brent-Kung, Ladner-Fischer, and Kogge-Stone designs, are made using metrics such as logic depth, wiring complexity, and area, with references to Knowles (1999) and Harris (2003). This report highlights the design's efficiency and suitability for high-performance applications.

Design Approach

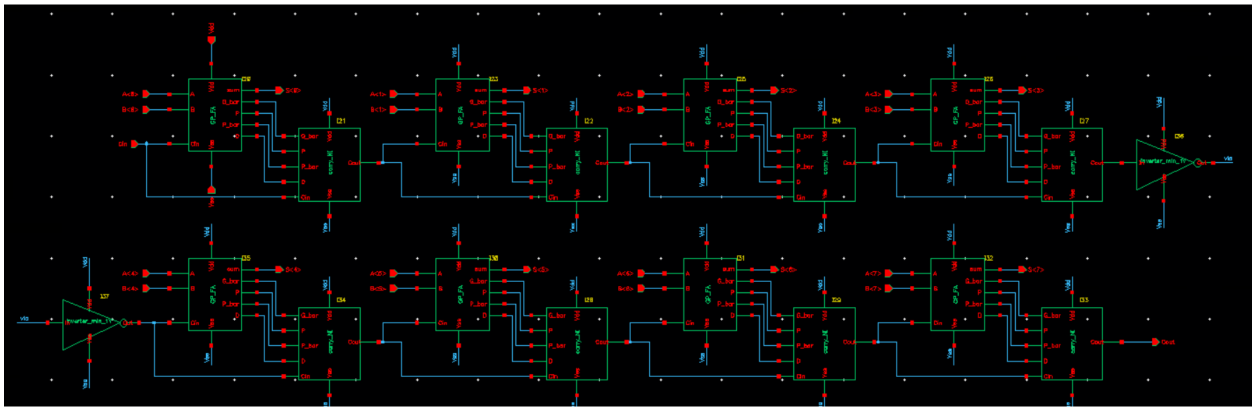


Figure 1. Manchester adder schematic of top-level module.

The design of the 8-bit Manchester adder (figure 1.) was implemented using transmission gate static logic to achieve high performance and minimize propagation delay. The following steps outline the approach used:

Transmission Gate Design

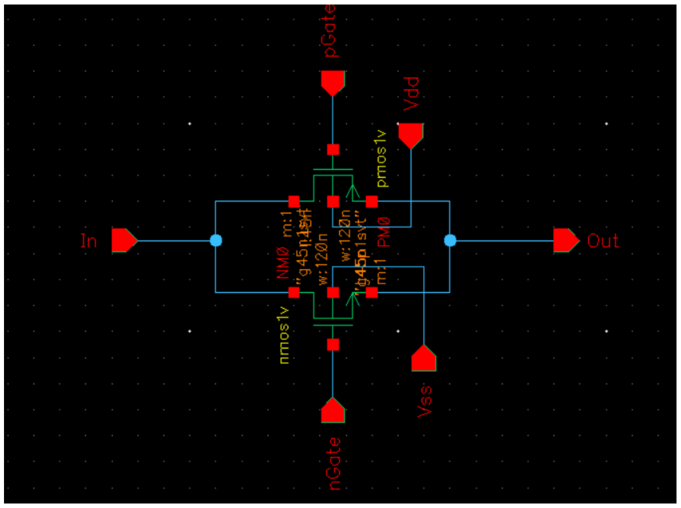


Figure 2. Transmission gate schematic.

The transmission gates (figure 2) were designed with minimum size transistors, using a width of 120 nm and a length of 45 nm for both PMOS and NMOS devices. This choice minimizes power consumption and area while maintaining sufficient speed.

XOR Gate Design

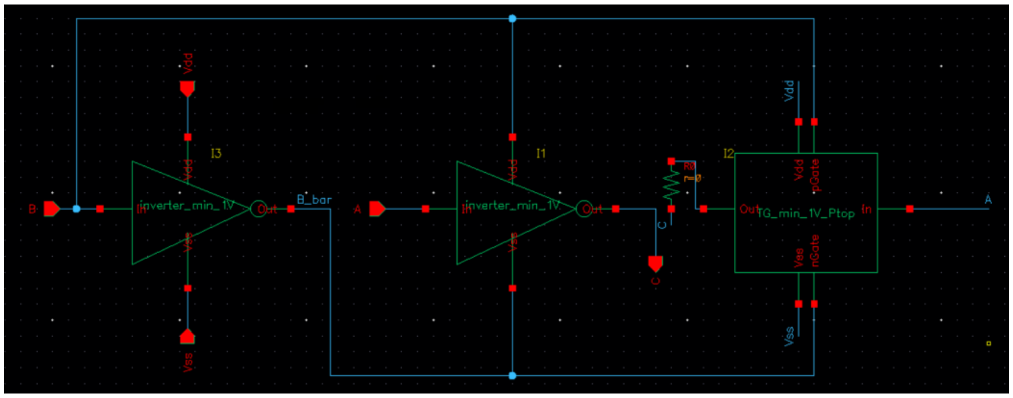


Figure 3. XOR gate schematic design.

Using the minimum-sized transmission gates, the XOR gate was designed (figure 3.) as the foundational component for the Manchester adder. The XOR gate is a critical building block for generating the sum output of the full adder.

AND Gate Design

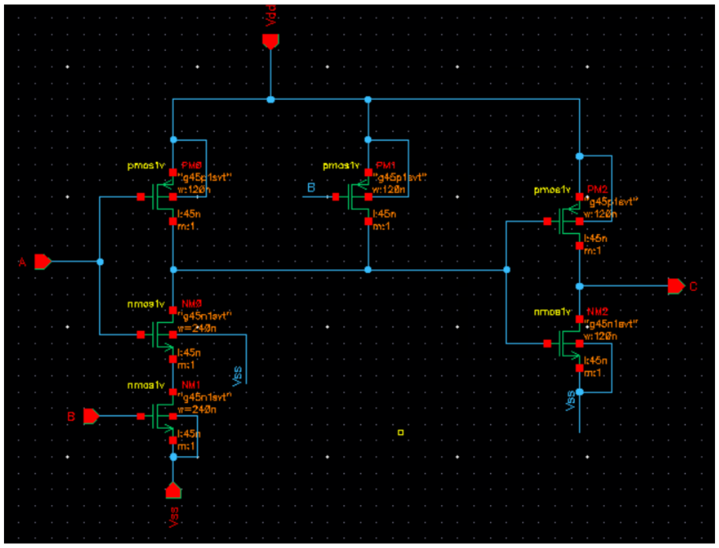

Vdd
pmos1v PM0 "g45p1svt" w:120n l:45n m:1
pmos1v PM1 "g45p1svt" w:120n l:45n m:1
B
pmos1v PM2 "g45p1svt" w:120n l:45n m:1
A
C
nmos1v NM0 "g45n1svt" w=240n l:45n m:1
nmos1v NM2 "g45n1svt" w:120n l:45n m:1
Vss
nmos1v NM1 "g45n1svt" w=240n l:45n m:1
B
Vss
Vss

Figure 4. AND gate transistor level schematic

For the AND gate (figure 4.), a CMOS NAND-based implementation was chosen. The pull-down network (PDN) of the NAND gate was sized with NMOS devices having double the width (2-finger NMOS transistors) since the two NMOS transistors were connected in series. This reduced effective resistance and sped up the gate. The NAND gate's output was then inverted using a minimum-sized inverter to complete the AND functionality.

Full Adder Design

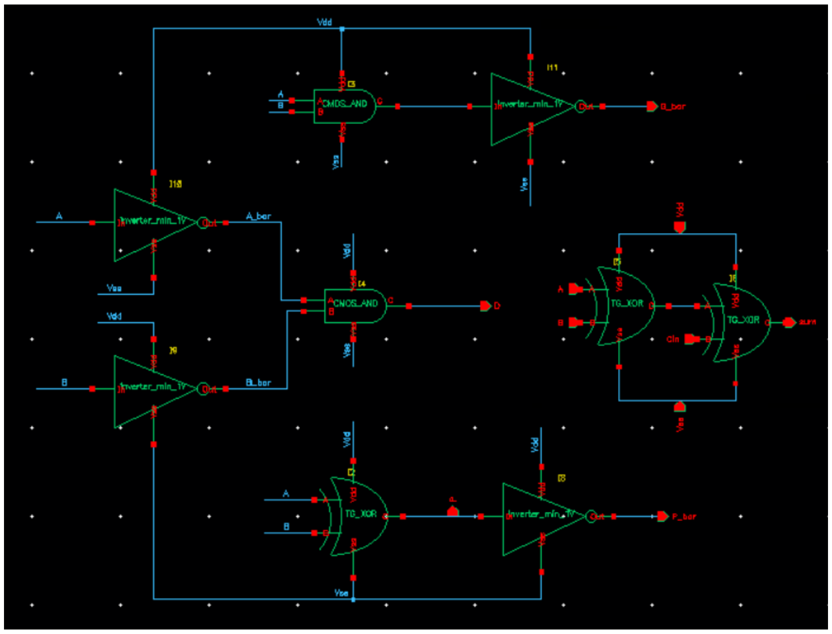


Figure 5. Full adder schematic

The full adder design (figure 5.) was built using two 2-input XOR gates for generating the sum output. Internally, the delete, propagate, and generate signals were calculated to optimize carry propagation. The full adder module outputs these signals to facilitate integration into the chain module.

Chain Module Design

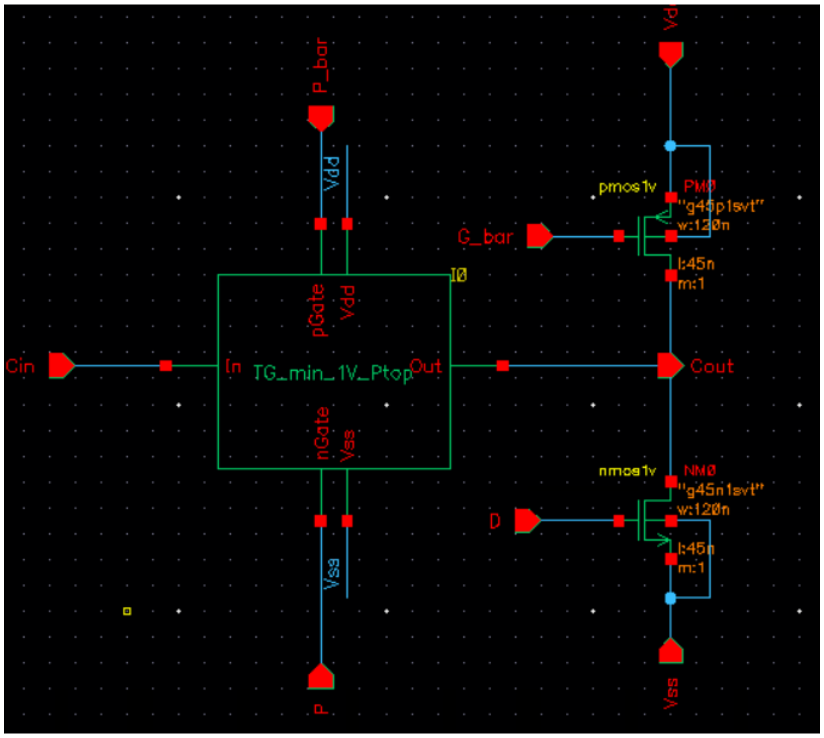


Figure 6. Chain module schematic.

The chain module (figure 6.) uses the propagate and generate signals from each full adder, carry propagation was efficiently managed across the adder chain, ensuring minimal delay and consistent functionality.

This hierarchical design approach ensures modularity and simplifies testing and verification. The use of minimum-sized transmission gates and careful transistor sizing for critical gates helped achieve the desired performance objectives of minimal delay and high throughput.

Layout Design

During the layout, I encountered a DRC "soft check" for stamped body connections in the NMOS transistors of the XOR gate, despite being correctly tied to V_{ss}(Ground). A similar issue occurred in a prior shift register design, where using the analog library's "gnd" pin resolved the error but may have caused post-layout simulation issues, as flagged by the TA. This issue not only affects the XOR gate but also prevents the entire 8-bit adder from passing LVS, even though all connections are properly made. Additional images and discussion of this issue are included in the appendix A for clarity and further investigation.

Transmission Gate Layout

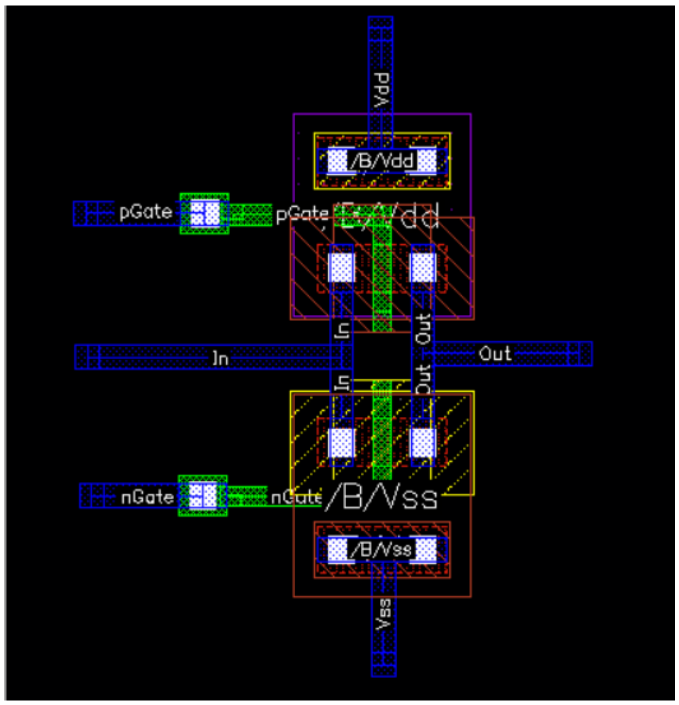


Figure 7. Tranmission gate layout.

XOR Gate Layout

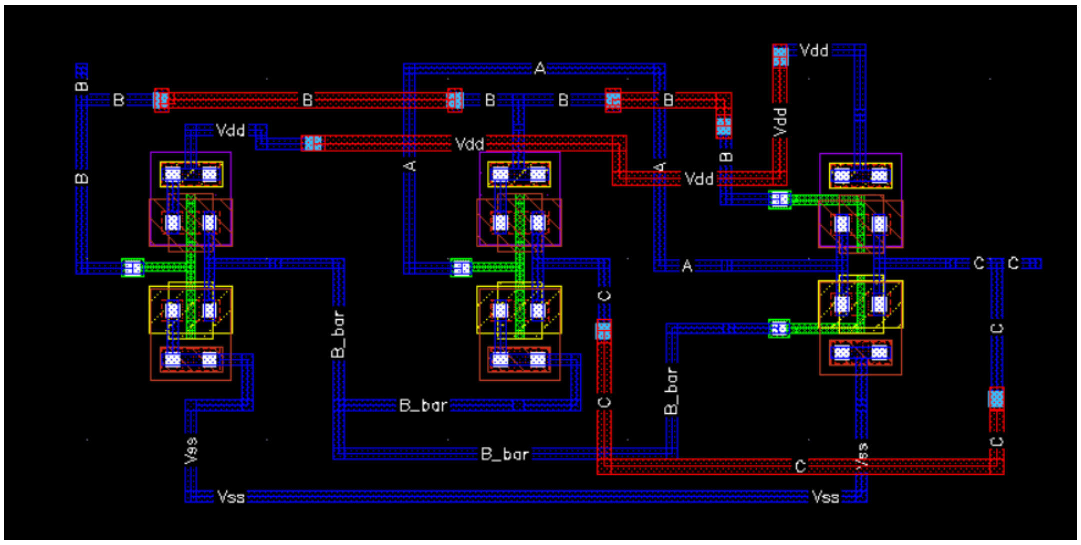


Figure 8. XOR gate layout design.

AND Gate Layout

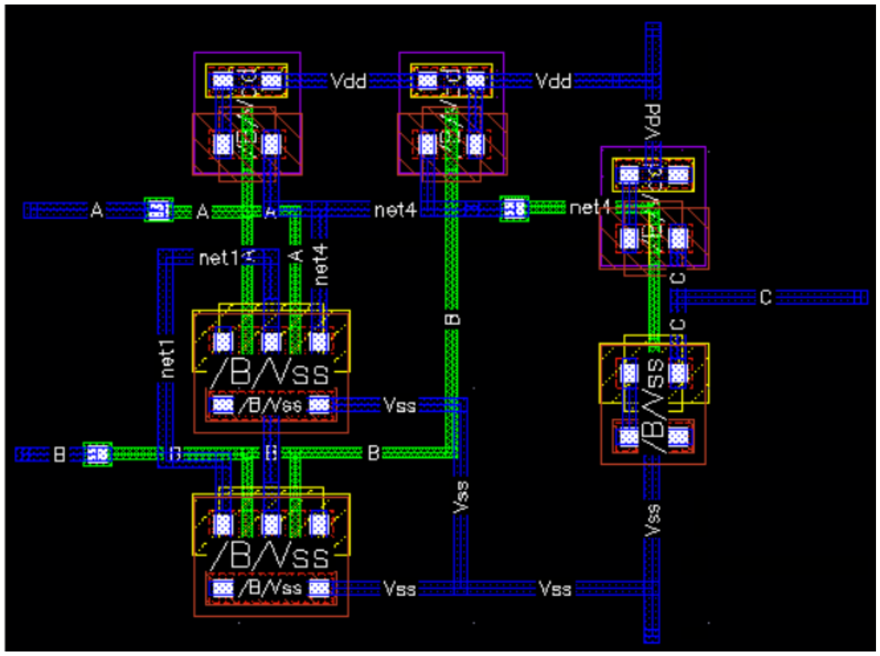


Figure 9. AND gate layout design.

Full Adder Layout (8-bit)

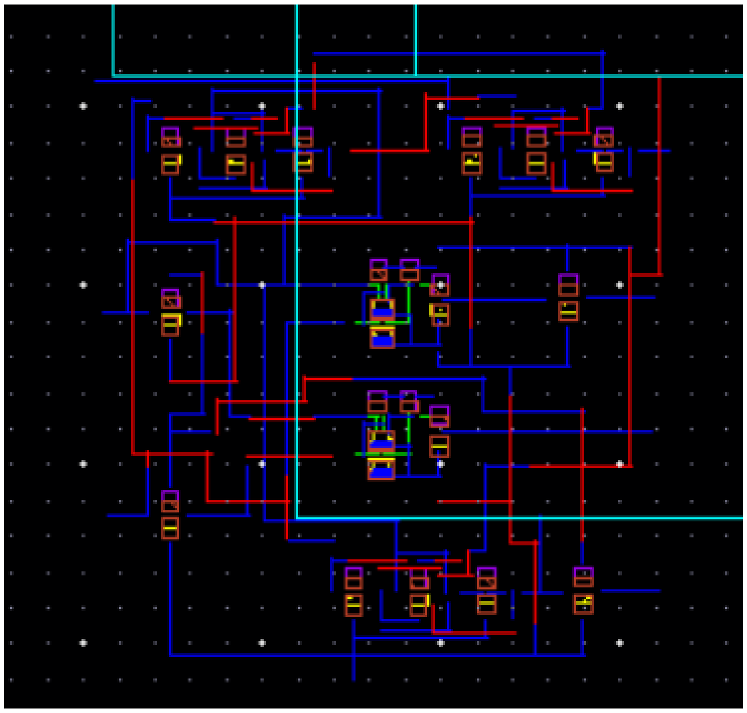


Figure 10. Full adder layout design.

Chain Module Layout

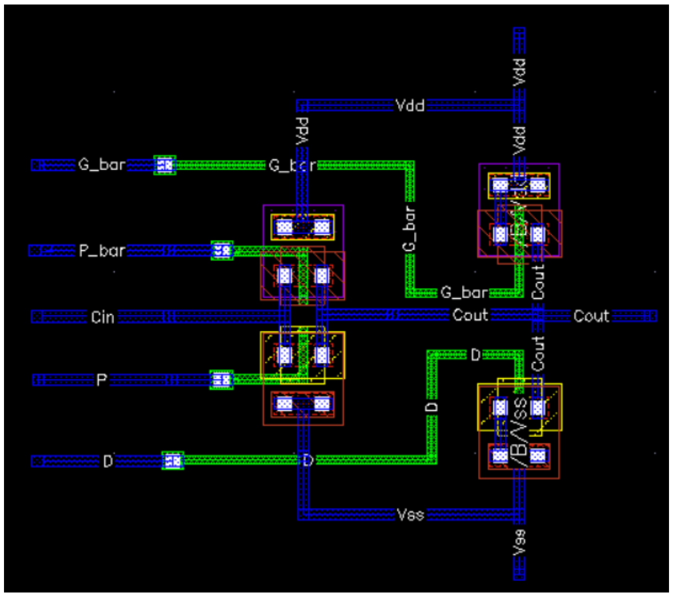


Figure 11. Chaain module layout design.

Manchester Adder Layout

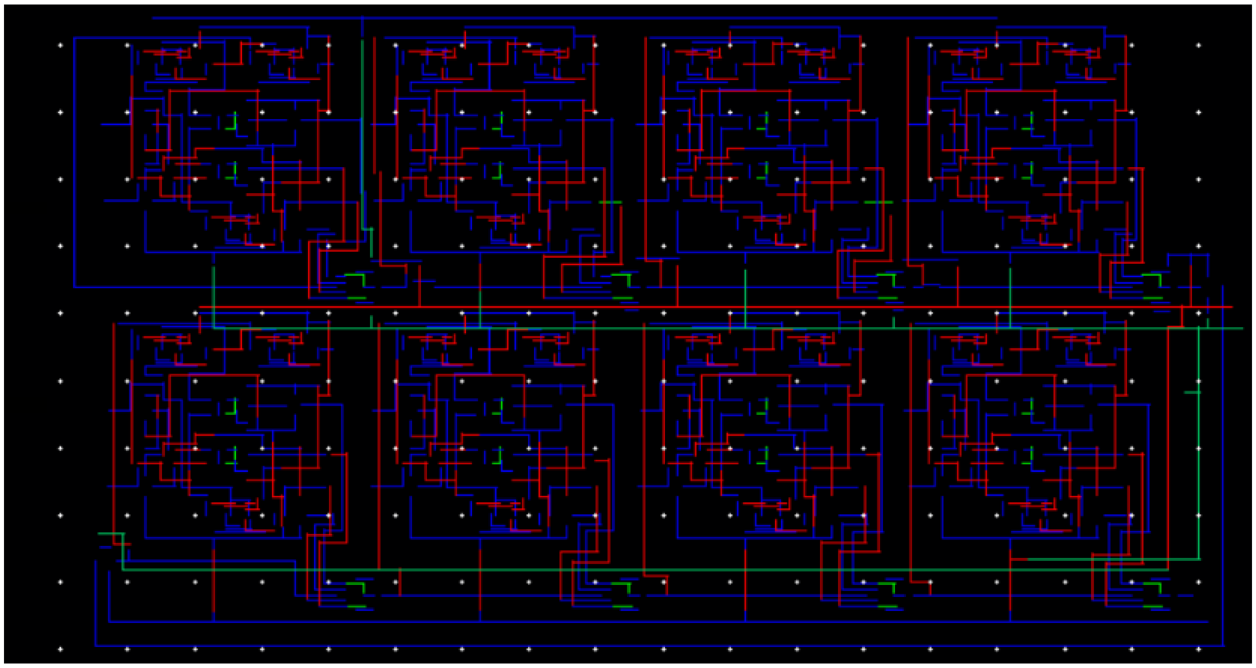
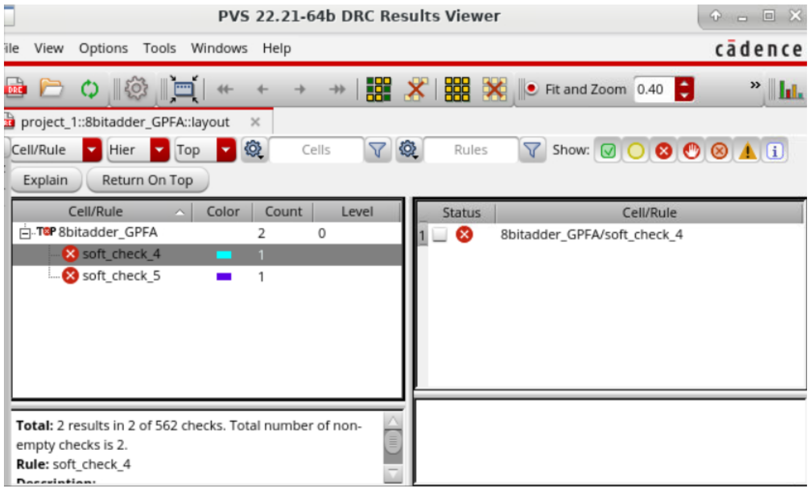


Figure 12. 8-bit Manchester adder layout design.

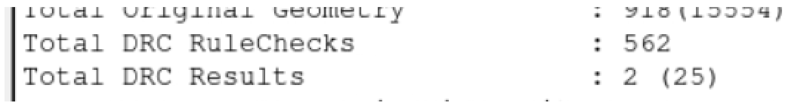
DRC and LVS (Manchester Only)


PVS 22.21-64b DRC Results Viewer
ile View Options Tools Windows Help
cādence
Fit and Zoom 0.40
project_1::8bitadder_GPFA::layout
Cell/Rule Hier Top Cells Rules Show:
Explain Return On Top

Cell/Rule	Color	Count	Level
TOP 8bitadder_GPFA		2	0
soft_check_4	🟦	1	
soft_check_5	🟪	1	

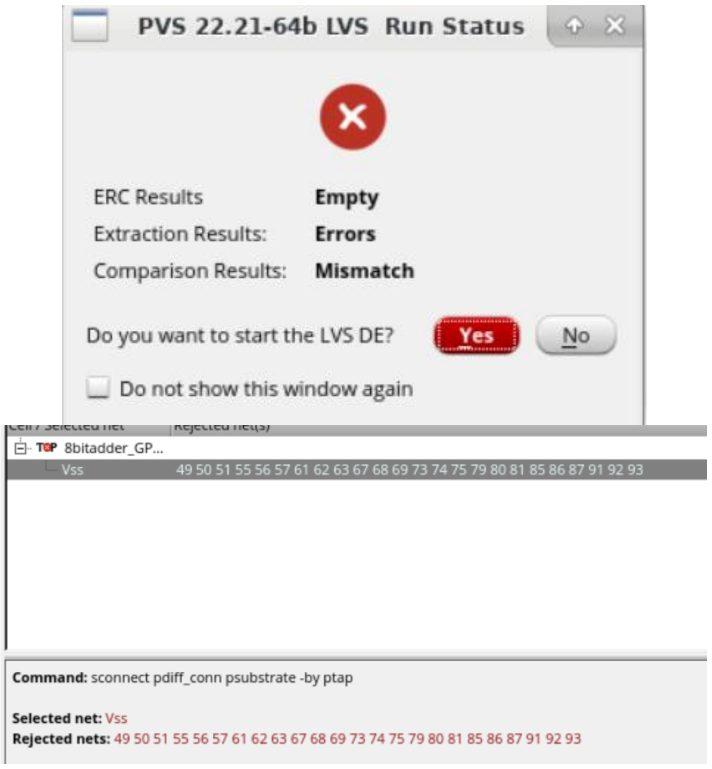
	Status	Cell/Rule
1	✗	8bitadder_GPFA/soft_check_4

Total: 2 results in 2 of 562 checks. Total number of non-empty checks is 2.
Rule: soft_check_4



```
Total Original Geometry          : 918(15554)
Total DRC RuleChecks             : 562
Total DRC Results                : 2 (25)
```

Figure 13. DRC results.



```
ERC Results           Empty
Extraction Results:   Errors
Comparison Results:   Mismatch

Do you want to start the LVS DE?   Yes   No

Do not show this window again
```

```
Cell / Selected net     Rejected net(s)
TOP 8bitadder_GP...
   Vss                  49 50 51 55 56 57 61 62 63 67 68 69 73 74 75 79 80 81 85 86 87 91 92 93

Command: sconnect pdiff_conn psubstrate -by ptap

Selected net: Vss
Rejected nets: 49 50 51 55 56 57 61 62 63 67 68 69 73 74 75 79 80 81 85 86 87 91 92 93
```

Figure 14. LVS results.

It is evident from the DRC and LVS results (Fig. 13 and Fig. 14) that a soft-checking error prevents proper post-layout results. Specifically, Fig. 14 shows that the LVS is rejecting all the V_{ss} nets from the XOR gate.

Simulation Results

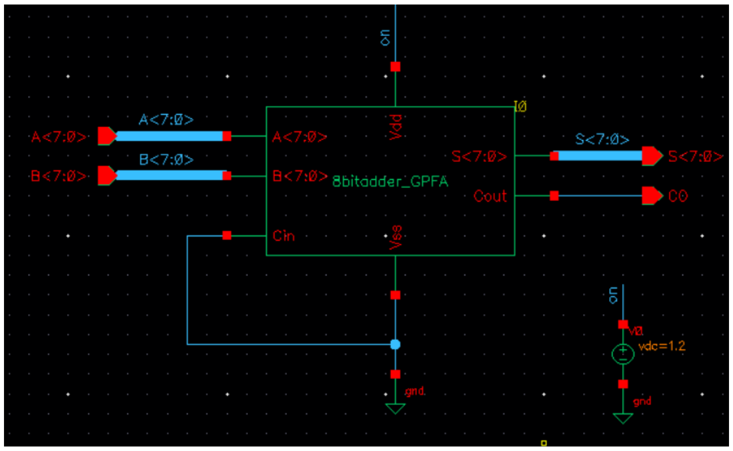


Figure 15. Schematic of testbench.

Post-layout simulation was not possible due to the DRC and LVS issues discussed in the previous section. The vector file used for the pre-layout simulation is included in the appendix B for reference.

Input	A1-A0 (Hex)	B1-B0(Hex)	Expected Output
1	BB	F5	1B0
2	E9	3F	128
3	1F	49	068
4	F0	AF	19F
5	E5	C4	1A9

Table 1. Simulation input signals.

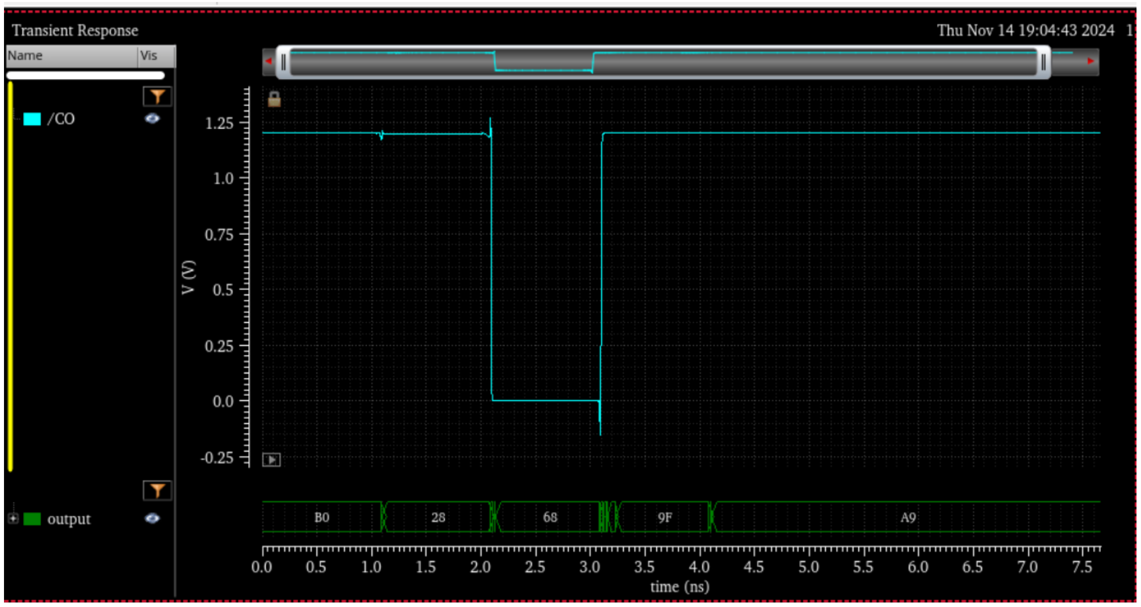


Figure 16. Waveform results.

The provided waveform (fig. 16) clearly demonstrates the functionality of the design, as the digital result combined with the carry-out signal matches the expected values from table 1. for all input vectors, confirming the accuracy of the pre-layout simulation.

Power Estimation

Input	A1-A0 (Hex)	B1-B0(Hex)
1	CB	11
2	EE	33
3	11	00
4	1A	A3
5	C4	6B
6	1F	F1
7	55	AC
8	C3	E2
9	32	28
10	FF	FF

Table 2. Power estimation inputs.

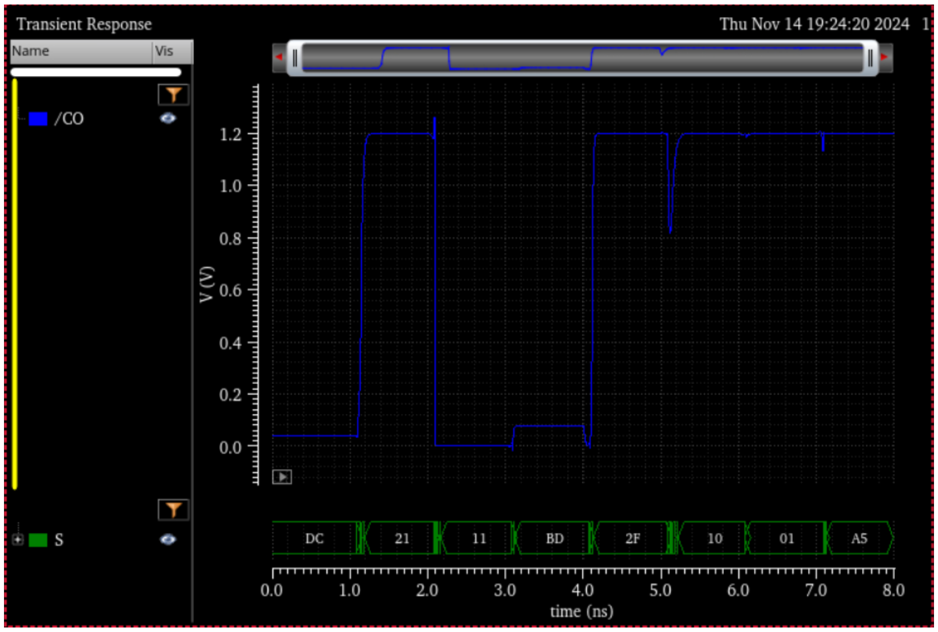


Figure 17. Power estimation input adder results.

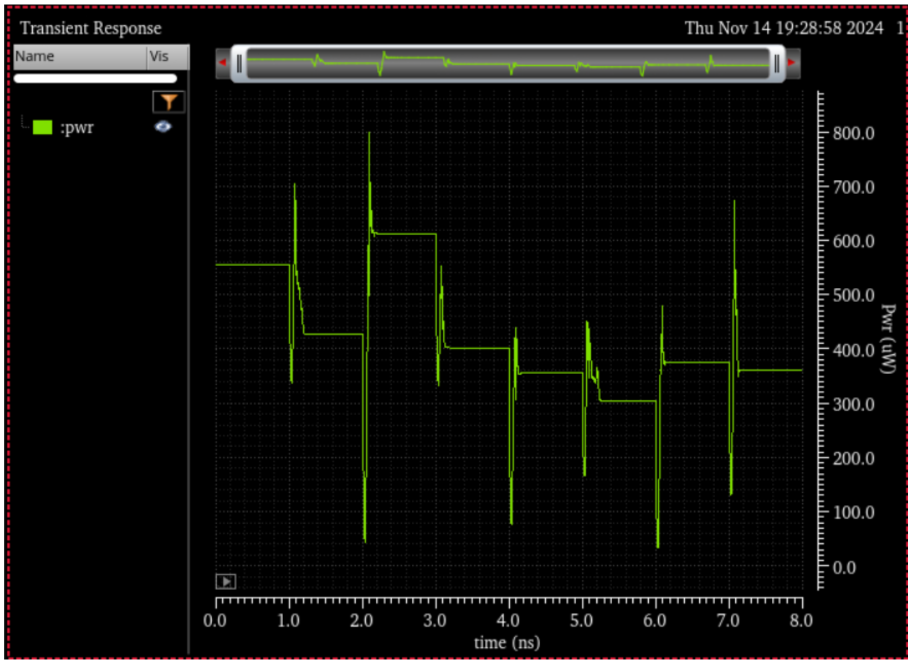


Figure 18. Power waveform.

,

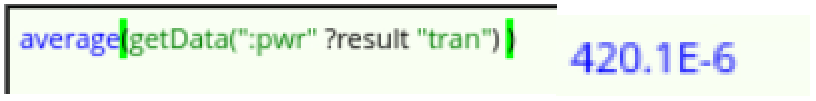

```
average(getData(":pwr" ?result "tran"))    420.1E-6
```

Figure 19. Calculator average function and result.

For power estimation, we utilized the input vectors shown in Table 2. To calculate the power, the output waveforms were saved in Cadence, and the "pwr" waveform (fig. 18) was extracted from the "psf" folder. By importing this waveform into the calculator tool in Cadence, we calculated the average power dissipation (fig. 19), which was determined to be approximately 420 µW.

Comparison with Other Adders

To evaluate the performance of our Manchester adder, we compared its power dissipation and efficiency with two widely used parallel prefix adders: the Kogge-Stone (KS) adder and the Brent-Kung (BK) adder.

Kogge-Stone Adder:

- The Kogge-Stone adder is known for its minimal logic depth and high-speed operation. However, it comes with increased wiring complexity and overhead area. In terms of power, the Kogge-Stone adder typically dissipates more power due to its extensive interconnects and high activity factor. For an 8-bit implementation, typical power dissipation values range between **450–500 µW** as detailed in [3], making it slightly higher than our Manchester adder's **420 µW**.

Brent-Kung Adder:

    - The Brent-Kung adder strikes a balance between logic depth and wiring complexity, offering a more power-efficient design compared to the Kogge-Stone adder. For an 8-bit design, its power dissipation is generally around **400 µW**, making it slightly more efficient than our Manchester adder. However, the Manchester adder's simpler design and reduced area requirements provide advantages in scenarios with stringent layout constraints.

Conclusion

In this project, an 8-bit Manchester adder was successfully designed using transmission gate static logic, emphasizing minimal propagation delay and efficient throughput. The design process included optimizing gate sizes for performance and ensuring proper connections for delete, propagate, and generate signals within the full adder and chain modules. While pre-layout simulations confirmed the functionality of the design, DRC and LVS issues, particularly with the XOR gate's ground connections, prevented successful post-layout simulations. This limitation also impacts the overall 8-bit adder's verification.

The Manchester adder's power dissipation, estimated at **420 µW**, was compared to other adders such as the Kogge-Stone and Brent-Kung adders. The results demonstrate that while the Manchester adder offers a simpler layout, its power efficiency is competitive, particularly in applications with moderate performance demands. Further refinements, including resolving DRC/LVS issues and optimizing the layout, could enhance the design's robustness and enable comprehensive post-layout verification.

References

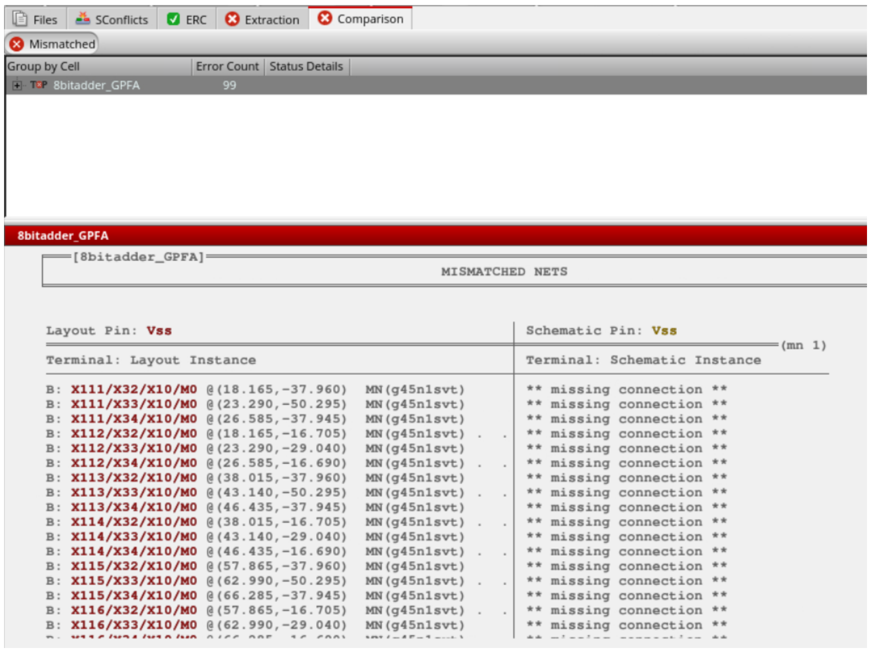
[1] S. Knowles, “A family of adders,” in Proc. 14th IEEE Symp. Computer Arithmetic, Apr. 1999, pp. 277–281.

[2] D. Harris, “A Taxonomy of Parallel Prefix Networks,” in Proc. 37th Asilomar Conf. Signals Systems and Computers, pp. 2213–2217, 2003.

[3] A. Sindhu and A. Bhatia, "8-Bit Kogge-Stone Adder," *EE 619 Course Project Report*. [Online]. Available: https://www.cl.cam.ac.uk/research/srg/han/ACS-P35/8-bit-KoggeStone-Adder.pdf

---

Appendix A: DRC and LVS Results



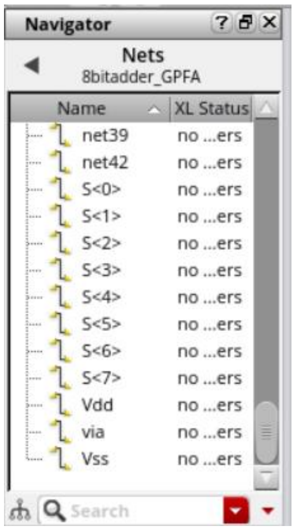
Group by Cell	Error Count	Status Details
TOP 8bitadder_GPFA	99	

```
8bitadder_GPFA
[8bitadder_GPFA]
                                   MISMATCHED NETS

Layout Pin: Vss                                        | Schematic Pin: Vss
                                                       |                        (mn 1)
Terminal: Layout Instance                              | Terminal: Schematic Instance

B: X111/X32/X10/M0 @(18.165,-37.960)  MN(g45n1svt)     | ** missing connection **
B: X111/X33/X10/M0 @(23.290,-50.295)  MN(g45n1svt)     | ** missing connection **
B: X111/X34/X10/M0 @(26.585,-37.945)  MN(g45n1svt)     | ** missing connection **
B: X112/X32/X10/M0 @(18.165,-16.705)  MN(g45n1svt) . . | ** missing connection **
B: X112/X33/X10/M0 @(23.290,-29.040)  MN(g45n1svt)     | ** missing connection **
B: X112/X34/X10/M0 @(26.585,-16.690)  MN(g45n1svt) . . | ** missing connection **
B: X113/X32/X10/M0 @(38.015,-37.960)  MN(g45n1svt)     | ** missing connection **
B: X113/X33/X10/M0 @(43.140,-50.295)  MN(g45n1svt) . . | ** missing connection **
B: X113/X34/X10/M0 @(46.435,-37.945)  MN(g45n1svt)     | ** missing connection **
B: X114/X32/X10/M0 @(38.015,-16.705)  MN(g45n1svt) . . | ** missing connection **
B: X114/X33/X10/M0 @(43.140,-29.040)  MN(g45n1svt)     | ** missing connection **
B: X114/X34/X10/M0 @(46.435,-16.690)  MN(g45n1svt) . . | ** missing connection **
B: X115/X32/X10/M0 @(57.865,-37.960)  MN(g45n1svt)     | ** missing connection **
B: X115/X33/X10/M0 @(62.990,-50.295)  MN(g45n1svt) . . | ** missing connection **
B: X115/X34/X10/M0 @(66.285,-37.945)  MN(g45n1svt)     | ** missing connection **
B: X116/X32/X10/M0 @(57.865,-16.705)  MN(g45n1svt) . . | ** missing connection **
B: X116/X33/X10/M0 @(62.990,-29.040)  MN(g45n1svt)     | ** missing connection **
```

Figure A1. LVS comparison errors.



Name	XL Status
net39	no ...ers
net42	no ...ers
S<0>	no ...ers
S<1>	no ...ers
S<2>	no ...ers
S<3>	no ...ers
S<4>	no ...ers
S<5>	no ...ers
S<6>	no ...ers
S<7>	no ...ers
Vdd	no ...ers
via	no ...ers
Vss	no ...ers

Figure A2. Navigator Nets connection window.

In this section, we discuss the issues encountered during the LVS (Layout vs. Schematic) comparison, specifically regarding the Vss net. The LVS report indicated errors, as shown in **Figure A1**, where the layout PIN for Vss is being rejected and multiple connections are reported as missing. This suggested an issue with the connections of the Vss net in the layout.

However, further inspection using the **Navigator tool** (referenced in **Figure A2**) shows that all nets, including Vss, are properly connected, and there are no errors associated with the Vss net in the layout. This discrepancy indicates that the connections are intact, yet LVS continues to flag the Vss net as problematic.

Despite this, I have reviewed the issue with TA Dali Lai, but no immediate solution has been identified. There is no obvious fix for this issue at present, and further investigation is required to resolve the LVS rejection.

Appendix B: Input Vector Files for Simulation

```
; radix specifies the number of bits of the vector.
radix 44 44 44 1

; io defines the vector as an input or output vector.
io   ii ii oo o

; vname assigns the name to the vector.
vname A<[7:0]> B<[7:0]> S<[7:0]> CO

; tunit sets the time unit.
tunit ns

; trise specifies the rise time of each input vector.
trise 0.1

; tfall specifies the fall time of each input vector.
tfall 0.1

; vih specifies the logic high voltage of each input vector.
vih 1.2

; vil specifies the logic low voltage of each input vector.
vil 0.0

; voh specifies the logic high voltage of each output vector.
voh 1.2

; vol specifies the logic low voltage of each output vector.
vol 0.0
```

```
; time per bit
period 1.0

; sets the window for when to check the output data
; this example checks for 200ps prior to each clock edge
; chk_window 0 0.2 1 period=1 first=0.8

; data
BB    F5    B0    1
E9    3F    28    1
1F    49    68    0
F0    AF    9F    1
E5    C4    A9    1
```