IC Layout Design of 16Bit Adder

VLSI Design System

Ahmet Deniz YILMAZ 21028656

Hacettepe University Electrical & Electronics Engg.

*Abstract* -This paper describes the design of an integrated circuit(IC) layout of a 16Bit adder with 2 different methods used for each 8Bits for performance comparison. First 8Bit is a ripple carry adder and second 8Bit is a carry-look ahead adder. The layout was designed by use of an open source software namely Electric

VLSI Design System.Circuit and modules tested in LTspice. The complete layout of the decoder was designed based on its schematic circuit, which consists of NOT gates, 2-input NAND gates, 3-input NAND gates, 2-input OR gates and XOR gates derived from NAND gates.

# 1.INTRODUCTION

Electronic devices have been widely used in many different fields and the size of these devices has been gradually reduced. An example of this is the mobile phone which is made smaller to enhance user’s mobility and usage time. These are the contribution of integrated circuit (IC) technology. With this technology, the modern devices have been reduced to convenient sizes. Besides that, mass production of IC has lowered the cost of production and made most electronic devices affordable. Today, an IC is smaller than a coin and can hold millions of transistors. Hence, further research in the design of IC is important to enhance the production of a more efficient and viable IC.

There are different technologies to construct integrated circuits such as bipolar integrated technology, NMOS technology and CMOS technology. In this project, CMOS technology is used. The main reason in using CMOS technology is due to its scalable high noise immunity and low power consumption. Basically, CMOS technology uses both NMOS and PMOS, which means only either one of both types of transistors will be ON at a time during the operation. Thus, CMOS IC consumes less power as power is used only when the NMOS and PMOS transistors are switching between on and off states.

The main objective of this report is design IC layout of 16Bit adder by Electric VLSI that free open source EDA system that provides service in handling IC layout, schematic drawing, textual hardware description language, and other features [1]

There are different technologies to construct integrated circuits such as bipolar integrated technology, NMOS technology and CMOS technology. In this project, CMOS technology is used. The main reason in using CMOS technology is due to its scalable high noise immunity and low power consumption. Basically, CMOS technology uses both NMOS and PMOS, which means only either one of both types of transistors will be ON at a time during the operation. Thus, CMOS IC consumes less power as power is used only when the NMOS and PMOS transistors are switching between on and off states [2].

# 2. LITERATURE REVIEW

## 2.1 Ripple Carry Adder

A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded.

-**One bit adder**

A one-bit full adder is a combinational circuit that forms the arithmetic sum of three.

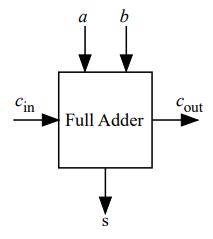
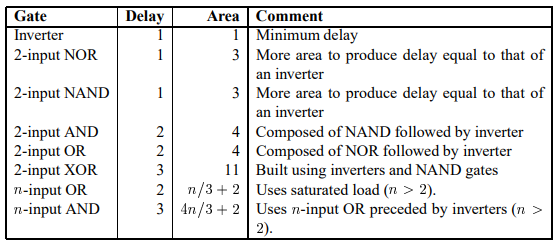


Figure 1. One Bit Adder

In the ripple carry adder, the output is known after the carry generated by the previous stage is produced. Thus, the sum of the most significant bit is only available after the carry signal has rippled through the adder from the least significant stage to the most significant stage. As a result, the final sum and carry bits will be valid after a considerable delay.

Table 1.CMOS gate delays and areas normalized relative to an inverter.



For a 32-bit processor, the carry chain normalized delay would be 131. The ripple carry adder can get very slow when many bits need to be added. In fact, the carry chain propagation delay is the determining factor in most microprocess.

## 2.2 Carry lookahead adder (CLA)

The carry lookahead adder (CLA) solves the carry delay problem by calculating the carry signals

in advance, based on the input signals. It is based on the fact that a carry signal will be generated

in two cases: when both bits and are 1, or when one of the two bits is and the carry-in is.

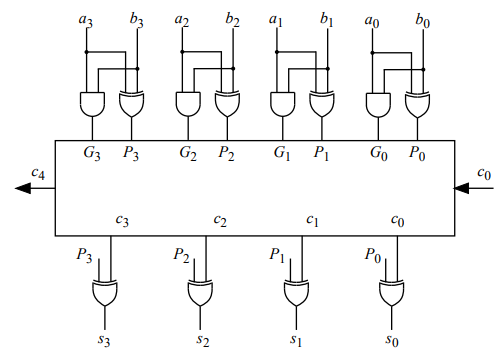


Figure 2. 4Bit carry look ahead implementation detail

The disadvantage of CLA is that the carry logic block gets very complicated for more than 4-bits. For that reason, CLAs are usually implemented as 4-bit modules and are used in a hierarchical structure to realize adders that have multiples of 4-bits. [3]

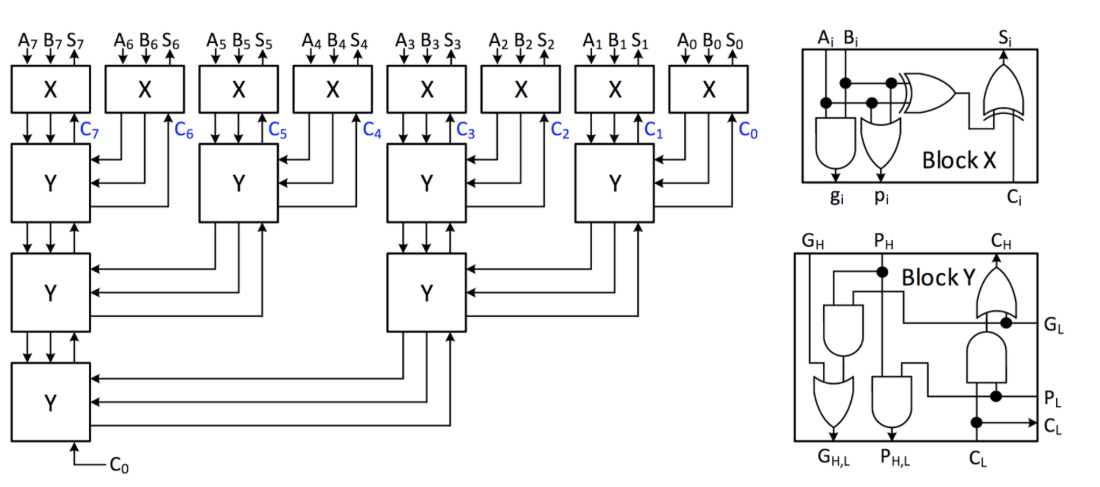


Figure 3. 8Bit Carry-Look Ahead adder

# 3.METHODOLOGY

3.1. Logic Gates

### 3.1.a 2-input NAND Gates

Figure 1 shows schematic diagram and icon view designed in Electric VLSI.

Exports named as Inp0, Inp1, Out and inputs colored dark-red while outputs are dark-green for easy management and readability purposes.

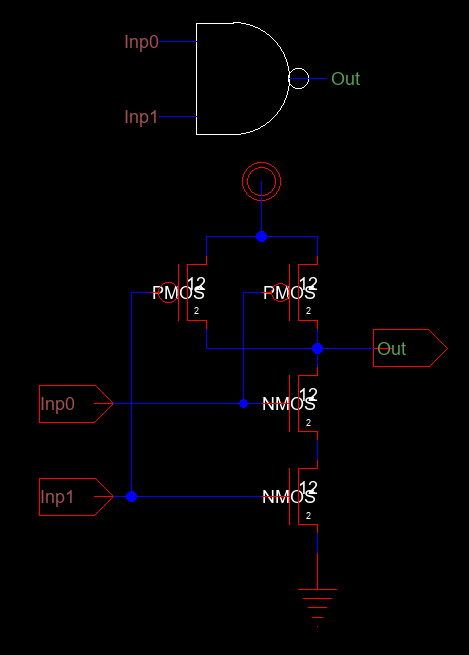
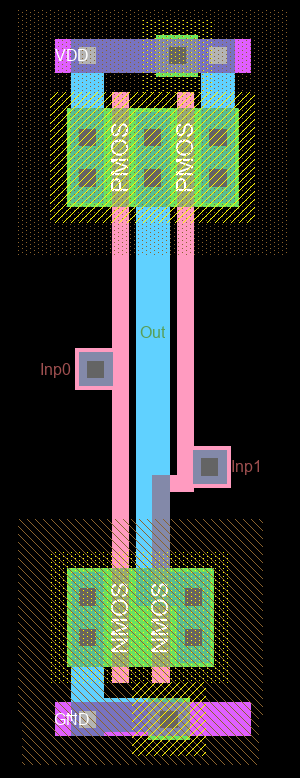
 

Figure 1. NAND2 Gate Schematic and icon view Figure 2. NAND2 Gate Layout Design

We get same correct results for both schematic and layout tests shown in Figure 3 as standart truth table. Applied capacitor to eliminate to spikes caused by states transitions.

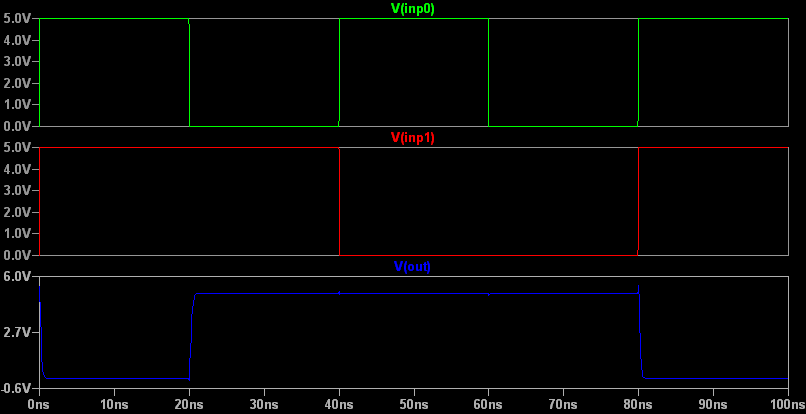


Figure 3. nand2 gate tests via Ltspice. (For both schematic and layout)

**Test Spicecode for nand2 layout and schematic**

\* Spice Code nodes in cell cell 'Nand2Gate{lay}'

\* Spice Code nodes in cell cell 'Nand2Gate{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.1.b 3-input NAND Gates

3-input NAND gate designed and used in XOR gate to avoid using extra transistor for applying nand gate to 3 signals since It’s also possible via 2-input NAND gates that would require more transistors.

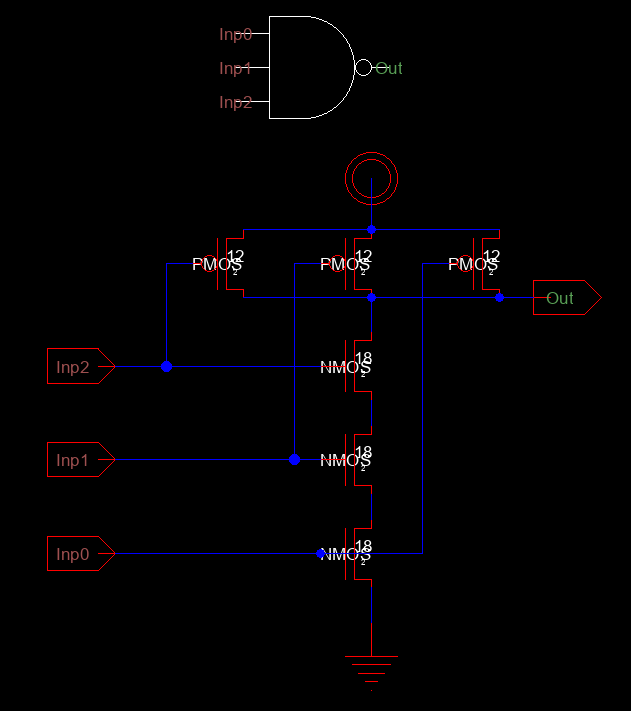
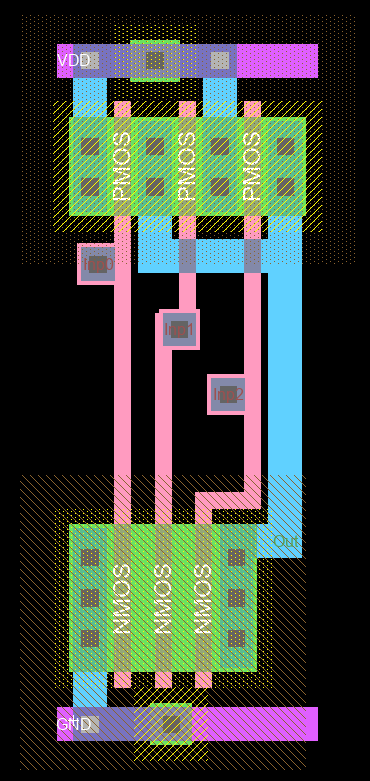
 

Figure 4. NAND3 Gate Schematic and icon view Figure 5. NAND3 Gate Layout Design

We get same correct results for both schematic and layout tests shown in Figure 6 as standart truth table. Applied capacitor to eliminate to spikes caused by states transitions.

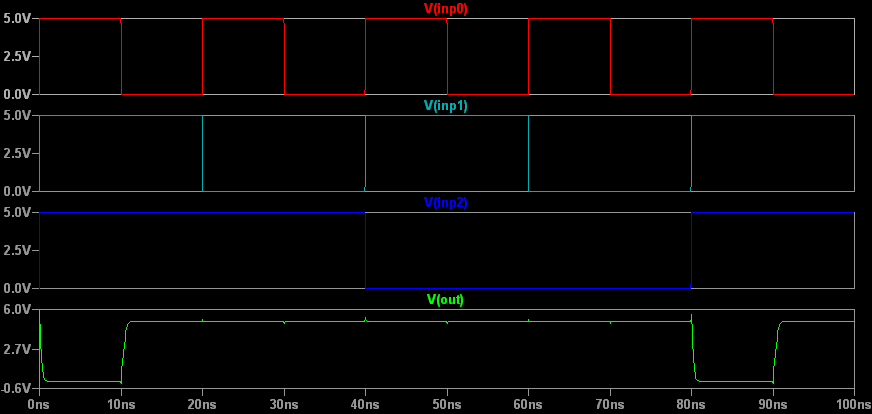


Figure 6. Nand3 gate test plot using Ltspice.(For both schematic and layout)

**Test Spicecode for nand3 layout and schematic**

\* Spice Code nodes in cell cell 'Nand3Gate{lay}'

\* Spice Code nodes in cell cell 'Nand3Gate{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 10n 20n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin3 Inp2 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.1.c Inverter

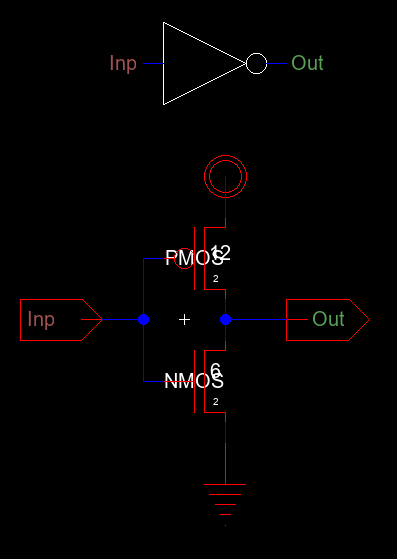
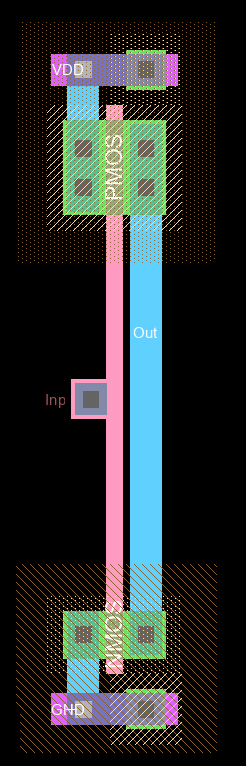
 

Figure 7.Inverter Schematic and icon view Figure 8. Inverter Layout Design

We get same correct results for both schematic and layout tests shown in Figure 3 as standart truth table. Applied capacitor to eliminate to spikes caused by states transitions.

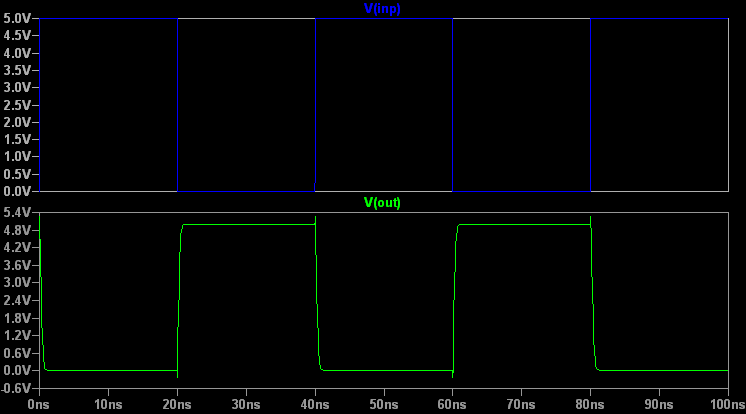


Figure 9.Inverter test plot using Ltspice. (For both schematic and layout).

**Test Spicecode for inverter layout and schematic**

\* Spice Code nodes in cell cell 'Inverter{lay}'

\* Spice Code nodes in cell cell 'Inverter{sch}'

vdd Vdd 0 DC 5

vin Inp 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.1.d 2-Input NOR Gate

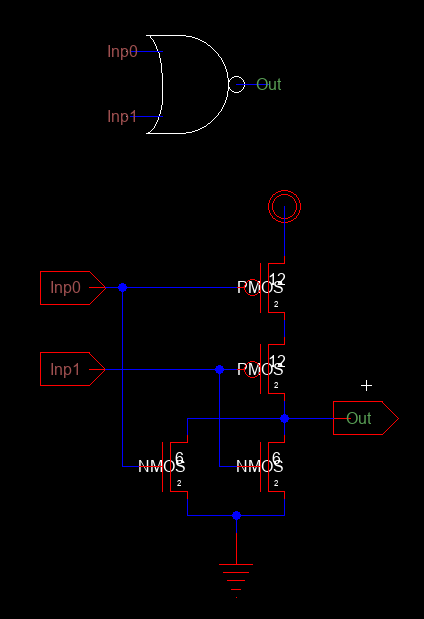
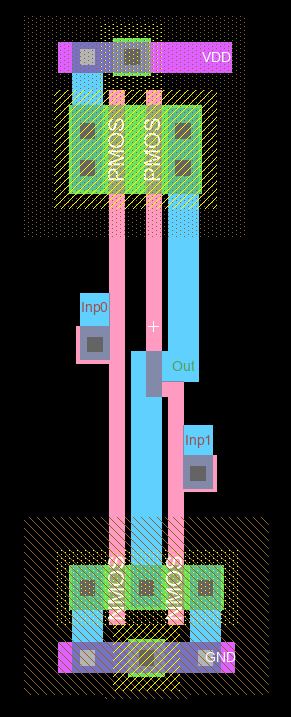
 

Figure 10.Nor2 Gate Schematic and icon view Figure 11.Nor2 Gate Layout Design

We get same correct results for both schematic and layout tests shown in Figure 12.

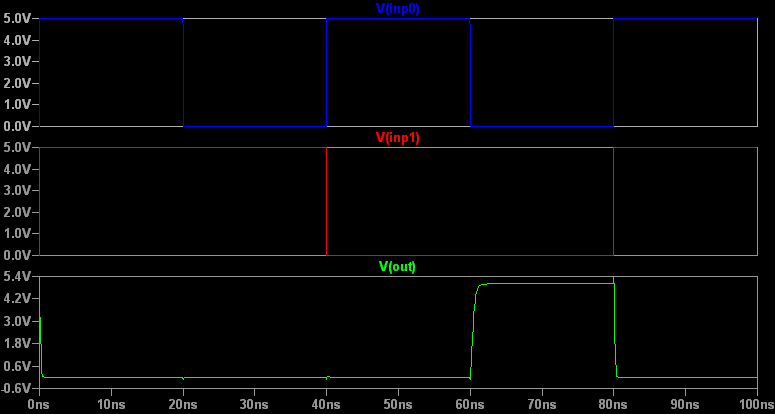


Figure 12.Nor2 gate test plot using Ltspice. (For both schematic and layout)

**Test Spicecode for Nor2 layout and schematic**

\* Spice Code nodes in cell cell 'Nor2Gate{lay}'

\* Spice Code nodes in cell cell 'Nor2Gate{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.1.e 2-Input AND Gate

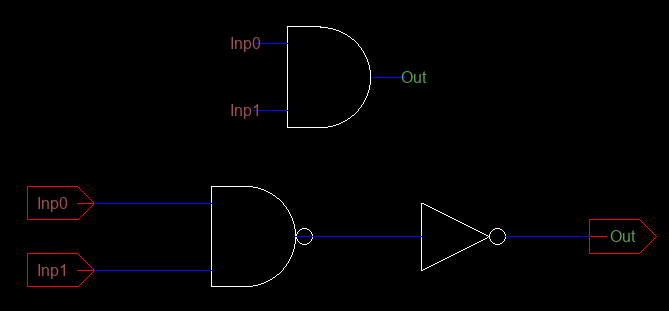
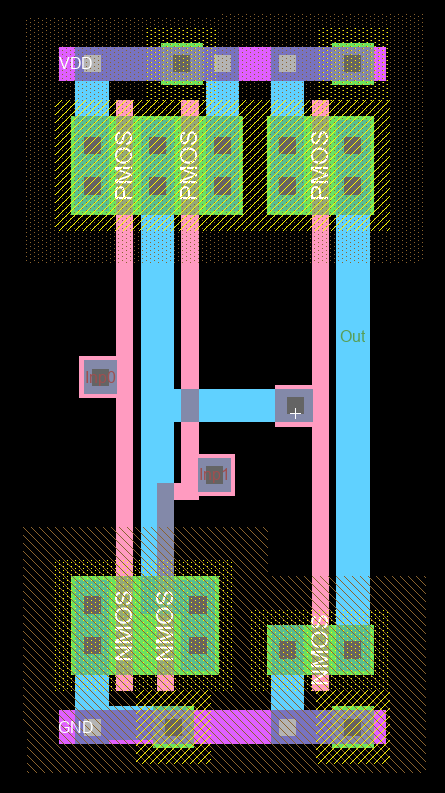
 

Figure 13. And2 Gate Schematic and icon view Figure 14. And2 Gate Layout Design

We get same correct results for both schematic and layout tests shown in Figure 15 as standart truth table. Applied capacitor to eliminate to spikes caused by states transitions.

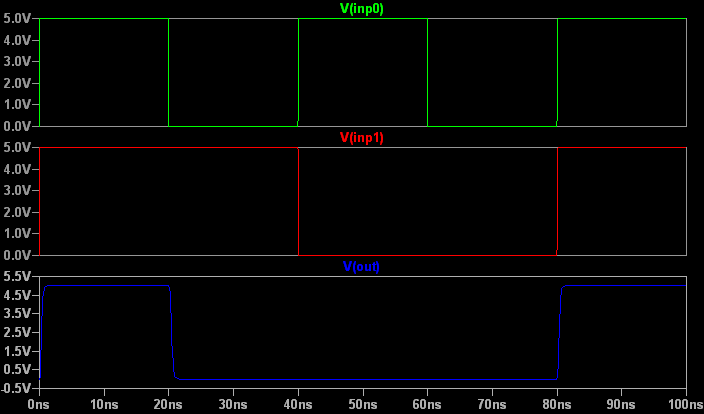


Figure 15. And2 gate test plot using Ltspice. (For both schematic and layout)

**Test Spicecode for and2 layout and schematic**

\* Spice Code nodes in cell cell 'And2Gate{lay}'

\* Spice Code nodes in cell cell 'And2Gate{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.1.f 2-Input OR Gate

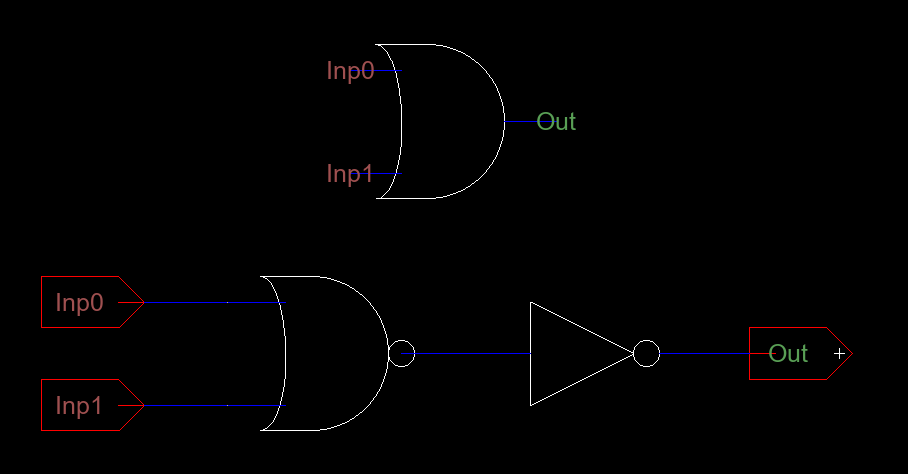
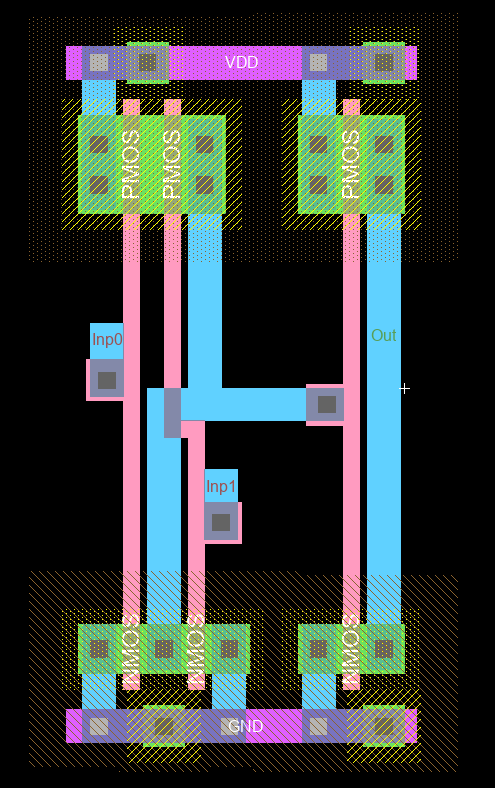
 

Figure 16. Or2 Gate Schematic and icon view Figure 17. Or2 Gate Layout Design

We get same correct results for both schematic and layout tests shown in Figure 18 as standart truth table. Applied capacitor to eliminate to spikes caused by states transitions.

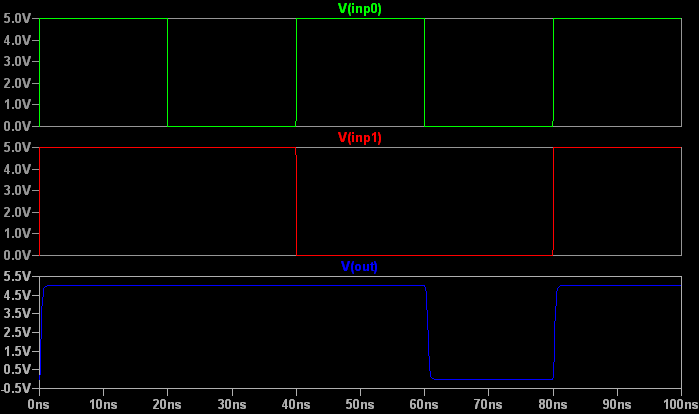


Figure 18. Or2 gate test plot using Ltspice. (For both schematic and layout)

**Test Spicecode for or2 layout and schematic**

\* Spice Code nodes in cell cell 'Or2Gate{lay}'

\* Spice Code nodes in cell cell 'Or2Gate{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.1.g XOR Gate

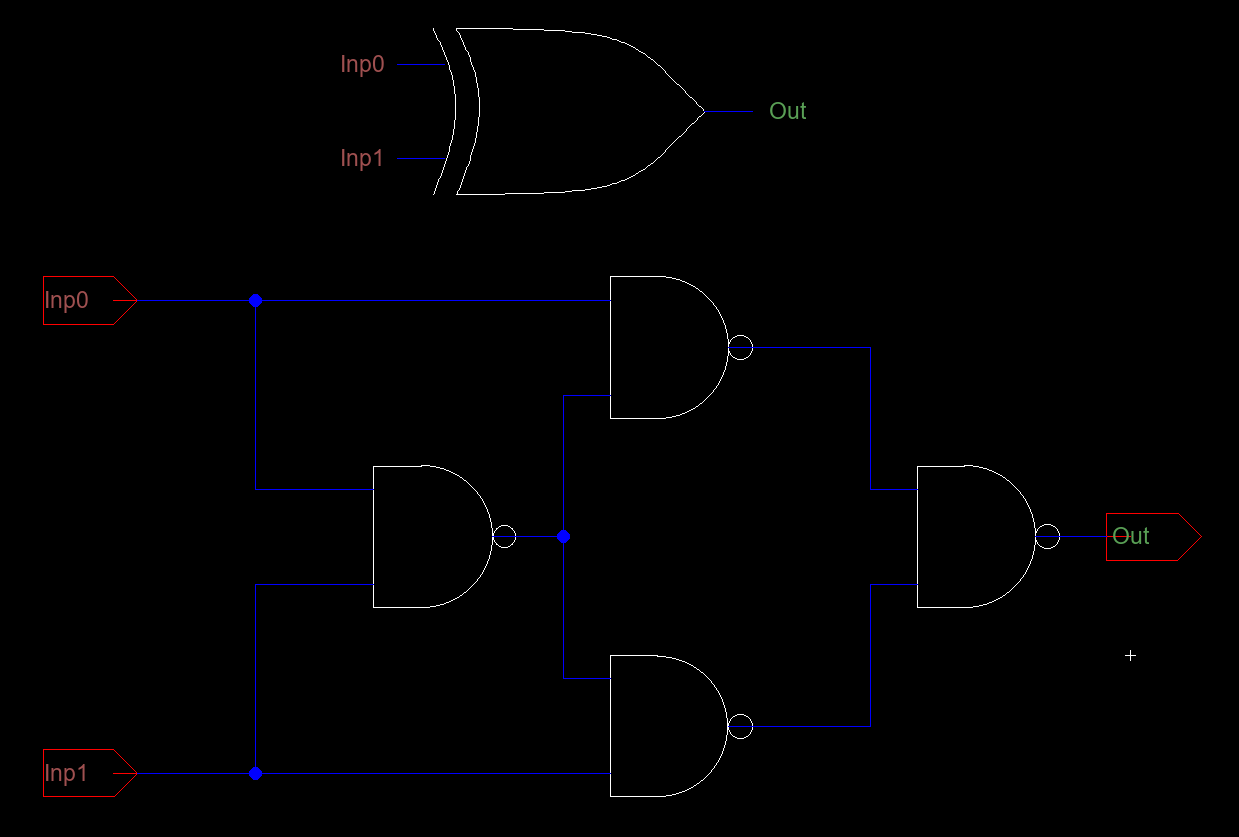


Figure 19. Xor Gate Schematic and icon view

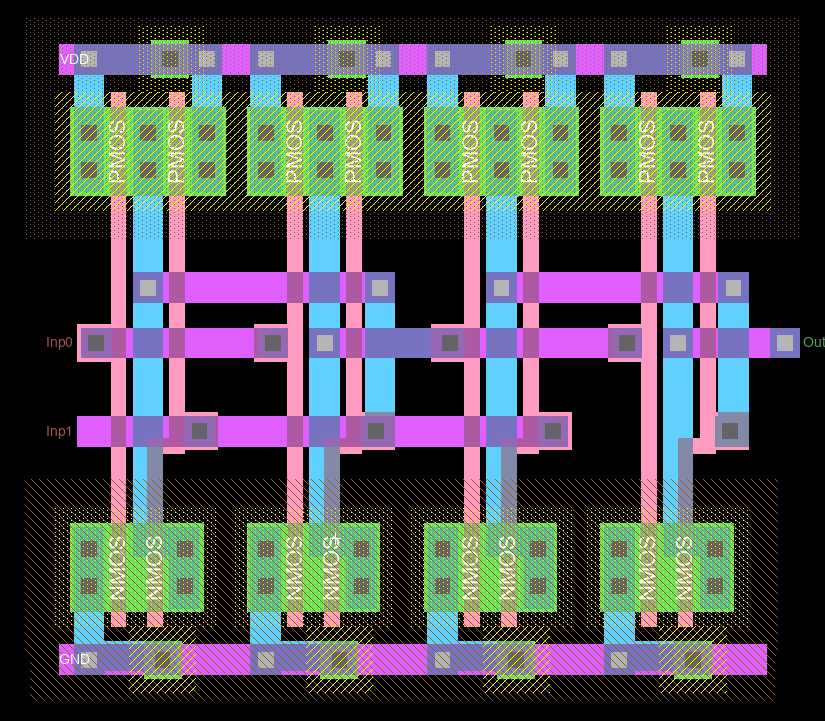


Figure 20. Xor Gate Layout Design

We get same correct results for both schematic and layout tests shown in Figure 21. as standart truth table. Applied capacitor to eliminate to spikes caused by states transitions.

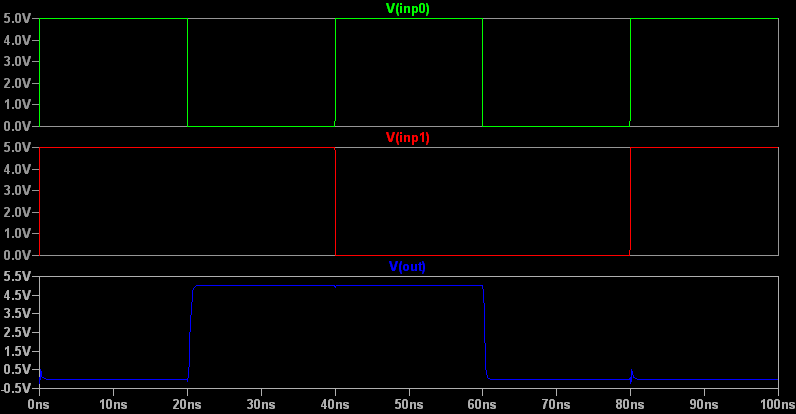


Figure 21. XOr gate test plot using Ltspice. (For both schematic and layout)

**Test Spicecode for xor layout and schematic**

\* Spice Code nodes in cell cell 'Xor2Gate{lay}'

\* Spice Code nodes in cell cell 'Xor2Gate{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

cload Out 0 50fF

.tran 0 100ns

.include C5\_models.txt

.END

## 3.2. Modules

### 3.2.a 1Bit Full Adder

1Bit Full adder( 3 bit adder) with carry in and carry out.

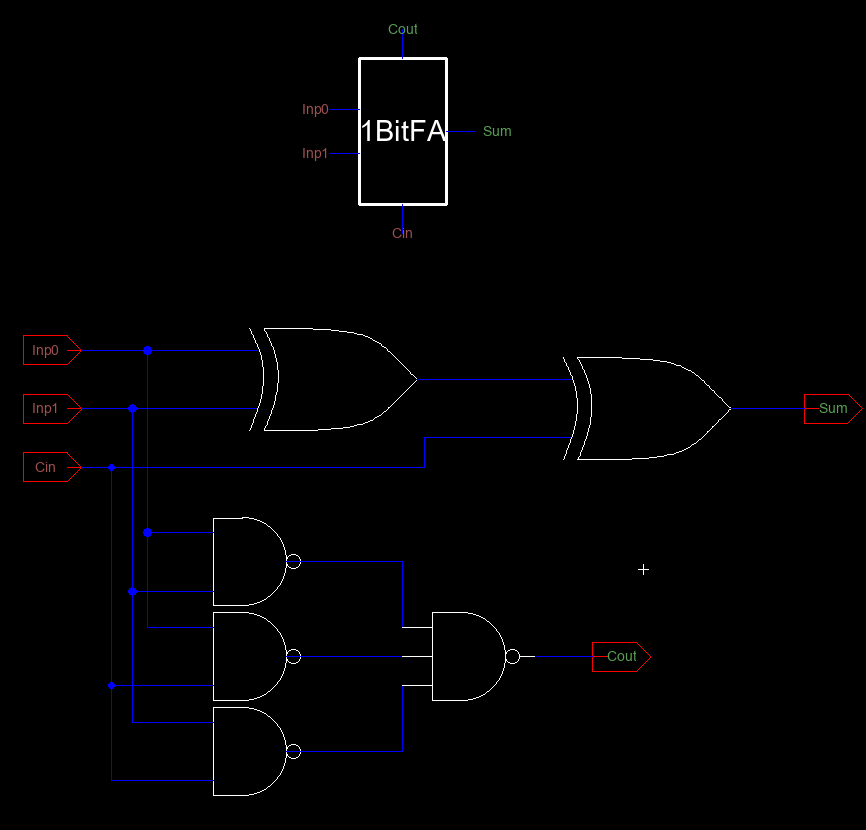


Figure 22. 1Bit FullAdder Schematic and icon view

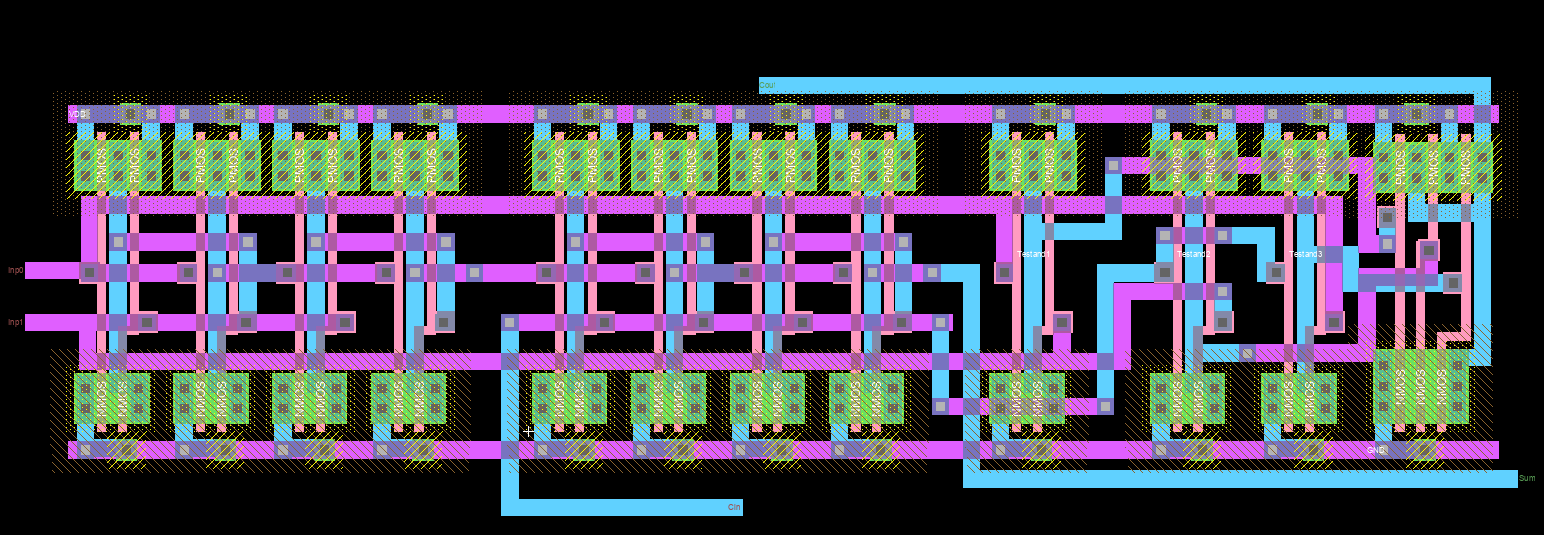


Figure 23. 1Bit FullAdder Layout

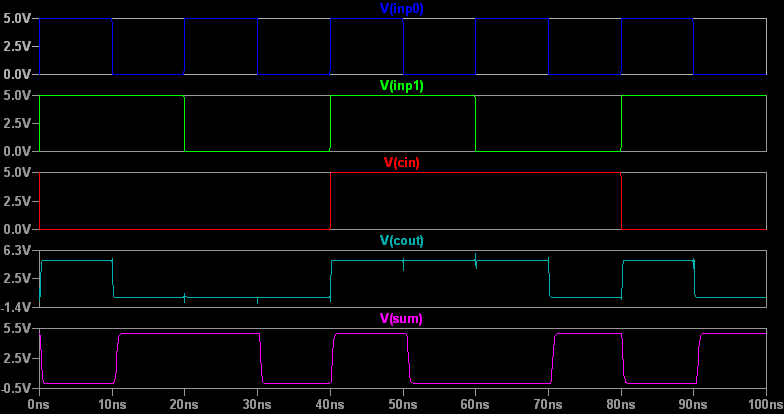


Figure 24.1Bit FullAdder Test Plot

**Test Spicecode for 1Bit Full Adder layout and schematic**

\* Spice Code nodes in cell cell ' 1BitFA{lay}'

\* Spice Code nodes in cell cell '1BitFA{sch}'

vdd Vdd 0 DC 5

vin Inp0 0 DC 5 pulse(0 5 10f 0.5f 0.5f 10n 20n)

vin2 Inp1 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin3 Cin 0 DC pulse(5 0 10f 0.5f 0.5f 40n 80n)

cload Sum 0 50fF

.tran 100ns

.include C5\_models.txt

.END

### 3.2.b 4Bit Full Adder

4Bit Full adder from four 1Bit full adders.

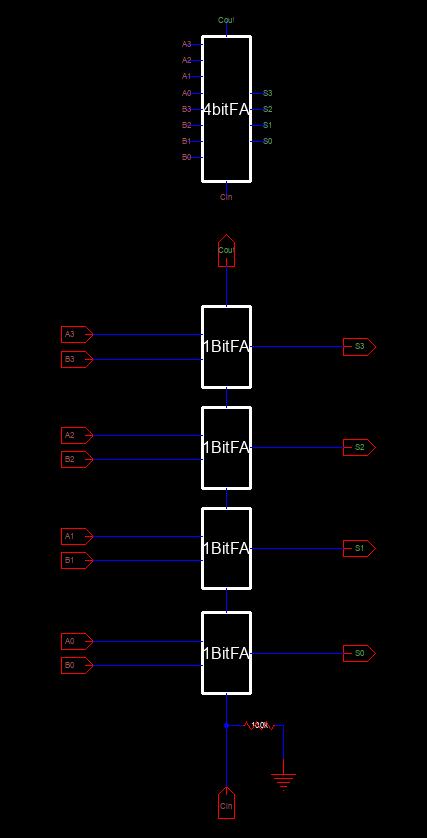


Figure 25. 4Bit FullAdder Schematic and icon view

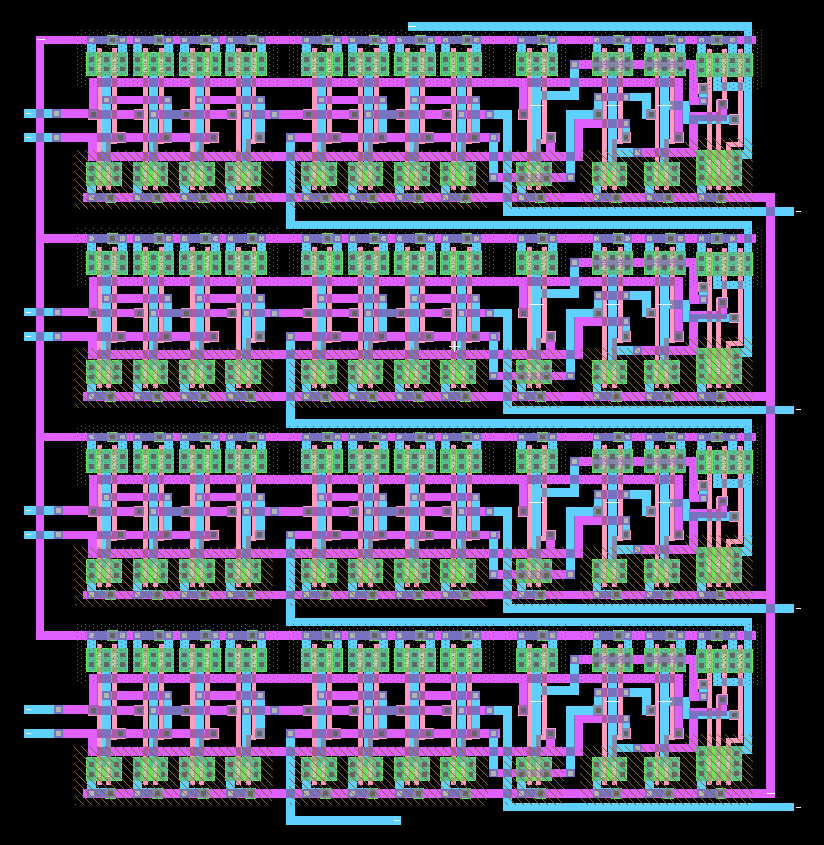


Figure 26. 4Bit FullAdder Layout

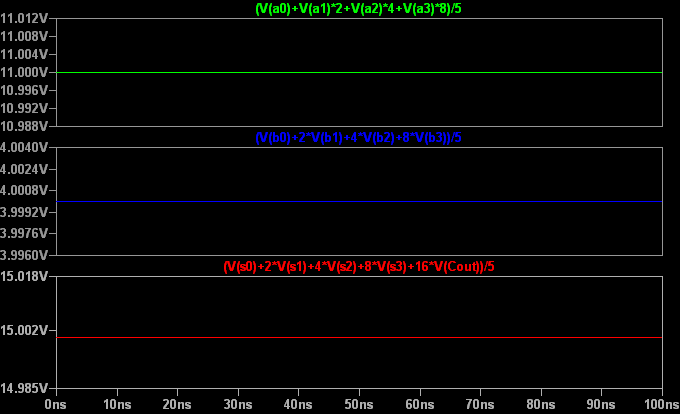


Figure 27.4Bit FullAdder Test Plot

**Test Spicecode for 4Bit Adder layout and schematic**

\* Spice Code nodes in cell cell ' 4BitFA{lay}'

\* Spice Code nodes in cell cell '4BitFA{sch}'

vdd Vdd 0 DC 5

\*A Bits

vin A0 0 DC 5

vin2 A1 0 DC 5

vin3 A2 0 DC 0

vin4 A3 0 DC 5

\*B Bits

vin5 B0 0 DC 0

vin6 B1 0 DC 0

vin7 B2 0 DC 5

vin8 B3 0 DC 0

cload S0 0 50fF

.tran 100ns

.include C5\_models.txt

.END

### 3.2.c 8Bit Adder

8 bit adder with two 4Bit full adders.

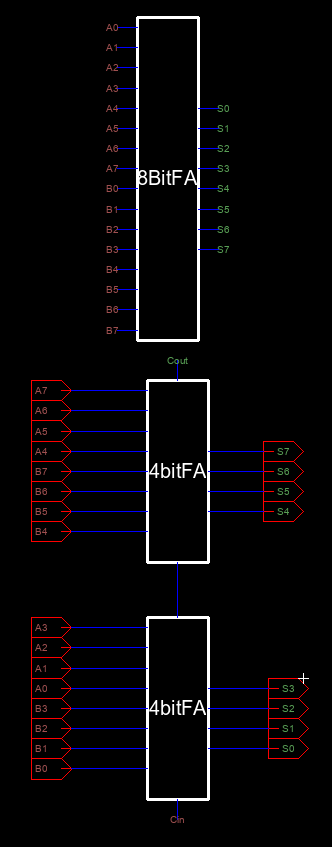


Figure 28. 8Bit Adder Schematic and icon view

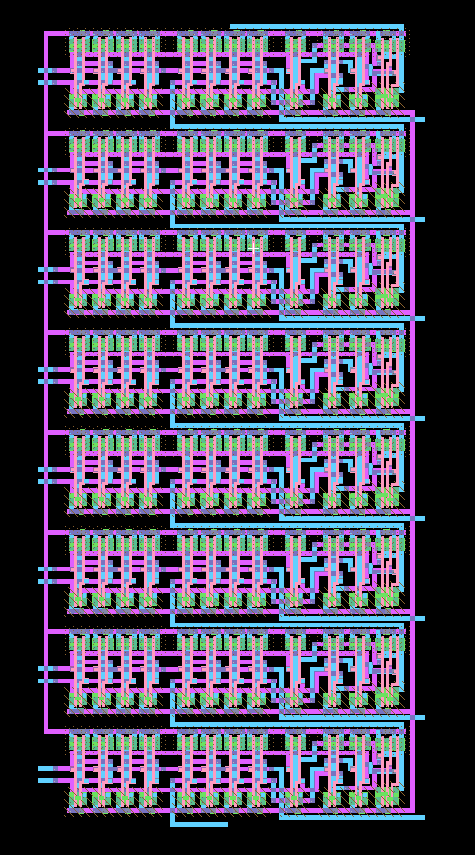


Figure 29. 8Bit Adder Layout

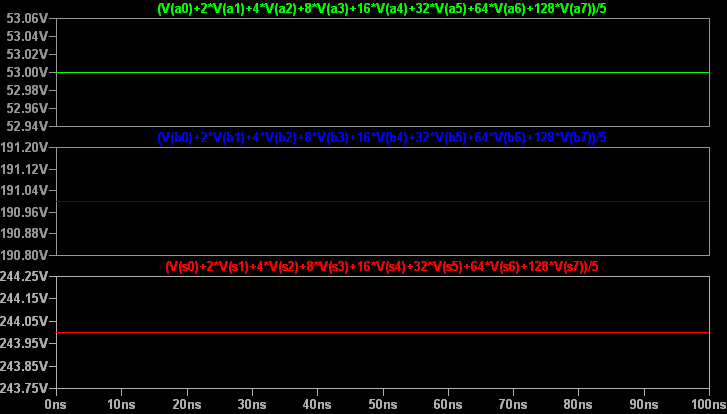


Figure 30. 8Bit Adder Test Plot via LTspice

**Test Spicecode for layout and schematic**

\* Spice Code nodes in cell cell ' 4BitFA{lay}'

\* Spice Code nodes in cell cell '4BitFA{sch}'

vdd Vdd 0 DC 5

\*A Bits

vin A0 0 DC 5

vin2 A1 0 DC 0

vin3 A2 0 DC 5

vin4 A3 0 DC 0

vin5 A4 0 DC 5

vin6 A5 0 DC 5

vin7 A6 0 DC 0

vin8 A7 0 DC 0

\*B Bits

vin9 B0 0 DC 5

vin10 B1 0 DC 5

vin11 B2 0 DC 5

vin12 B3 0 DC 5

vin13 B4 0 DC 5

vin14 B5 0 DC 5

vin15 B6 0 DC 0

vin16 B7 0 DC 5

cload S0 0 5fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.2.d 1Bit CLA Adder

1 Bit Adder for 8 bit carry look a head adder

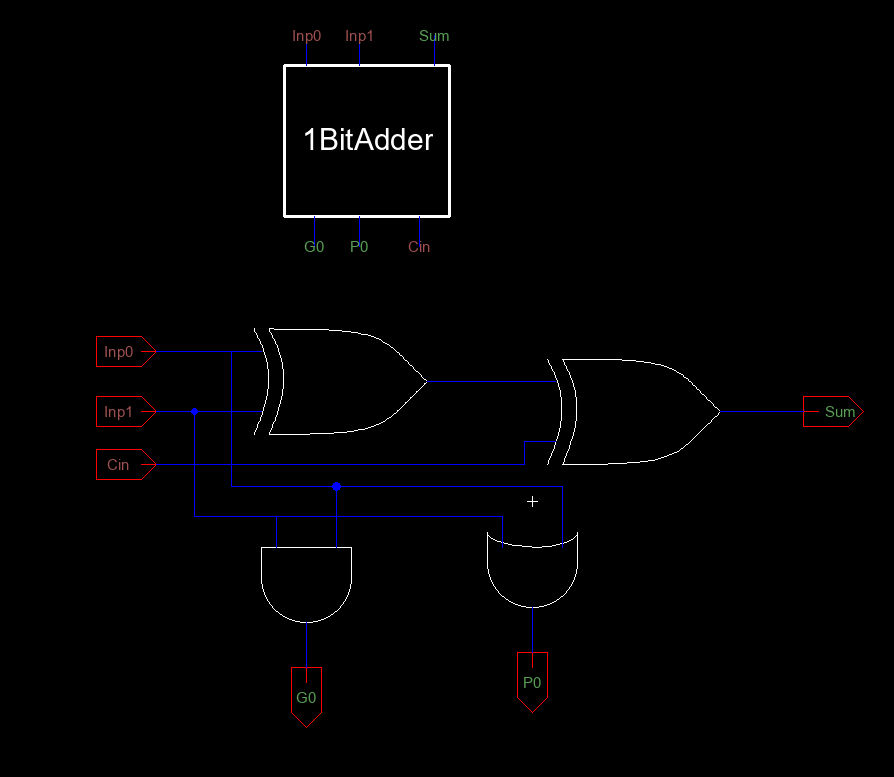


Figure 31. 1Bit CLA Adder Schematic and icon view

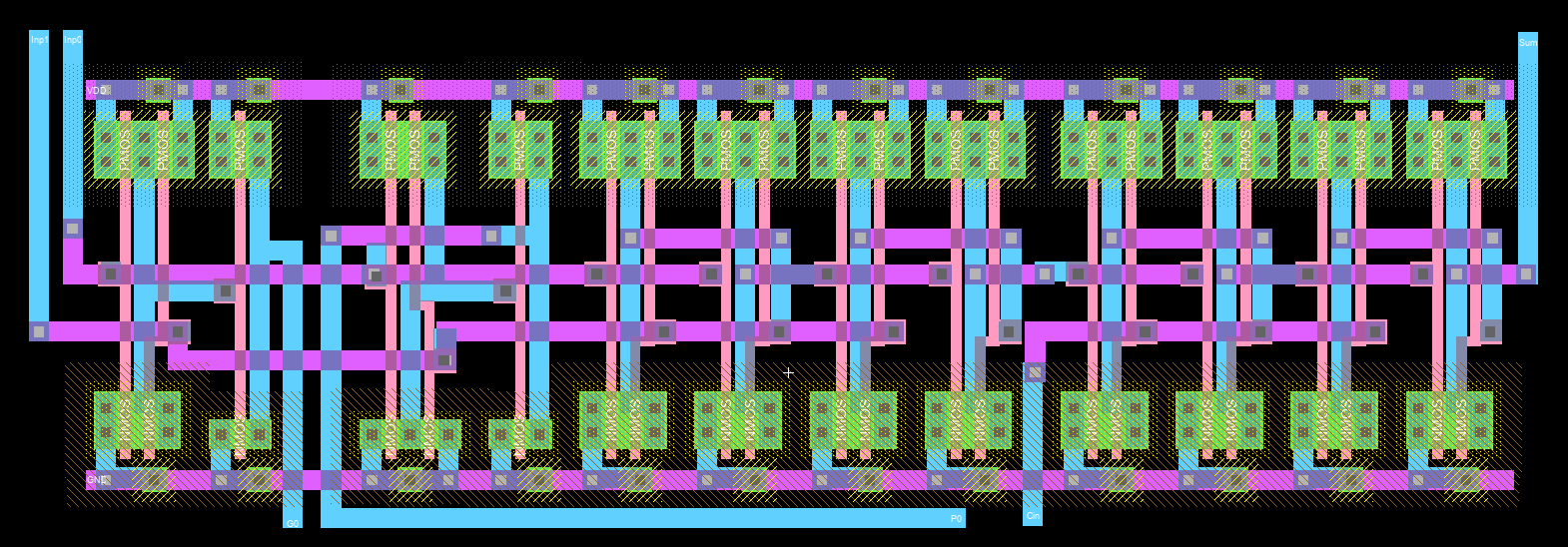


Figure 32. 1Bit CLA Adder Layout

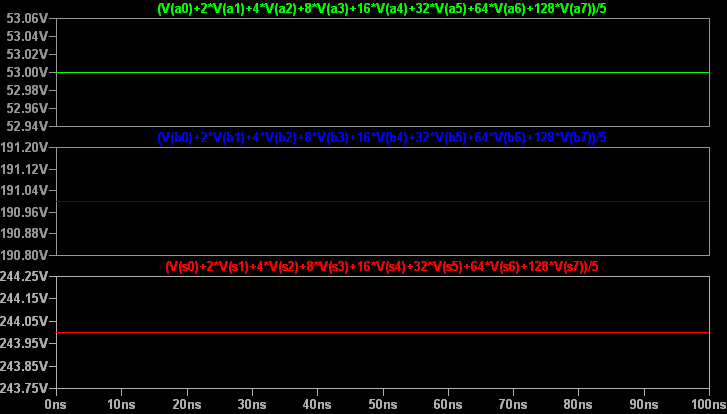


Figure 33. 8Bit Adder Test Plot via LTspice

**Test Spicecode for layout and schematic**

\* Spice Code nodes in cell cell ' 4BitFA{lay}'

\* Spice Code nodes in cell cell '4BitFA{sch}'

vdd Vdd 0 DC 5

\*A Bits

vin A0 0 DC 5

vin2 A1 0 DC 0

vin3 A2 0 DC 5

vin4 A3 0 DC 0

vin5 A4 0 DC 5

vin6 A5 0 DC 5

vin7 A6 0 DC 0

vin8 A7 0 DC 0

\*B Bits

vin9 B0 0 DC 5

vin10 B1 0 DC 5

vin11 B2 0 DC 5

vin12 B3 0 DC 5

vin13 B4 0 DC 5

vin14 B5 0 DC 5

vin15 B6 0 DC 0

vin16 B7 0 DC 5

cload S0 0 5fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.2.e CLA Block

Carry look a head block for CLA network.

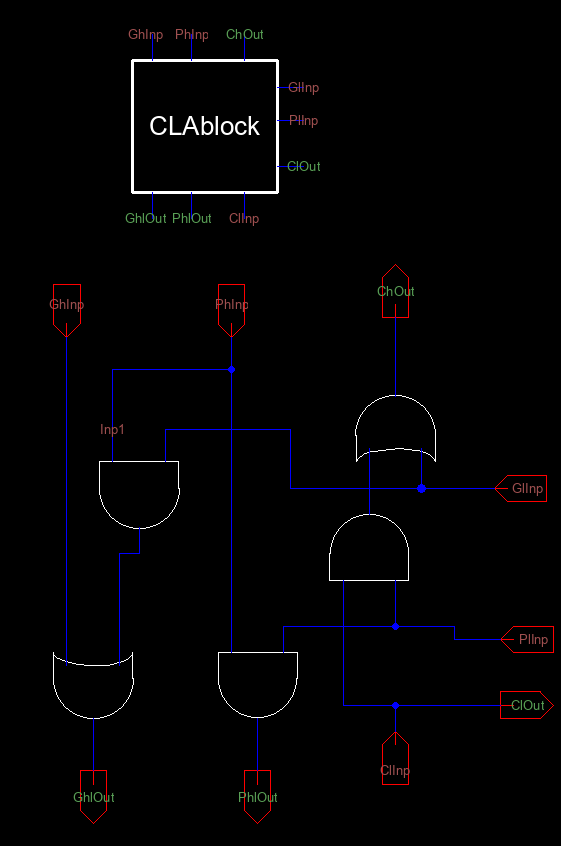


Figure 34. CLA Block Schematic and icon view

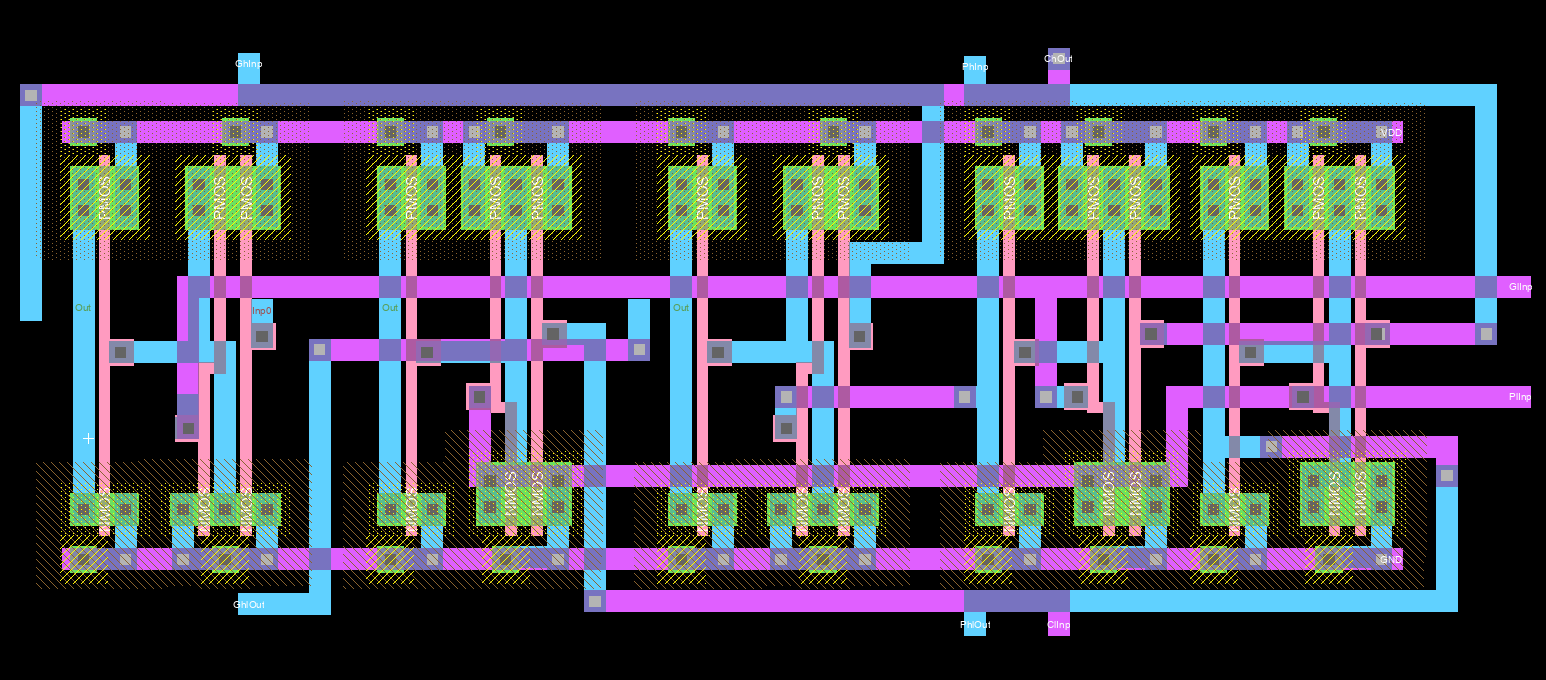


Figure 35. CLA Block Layout

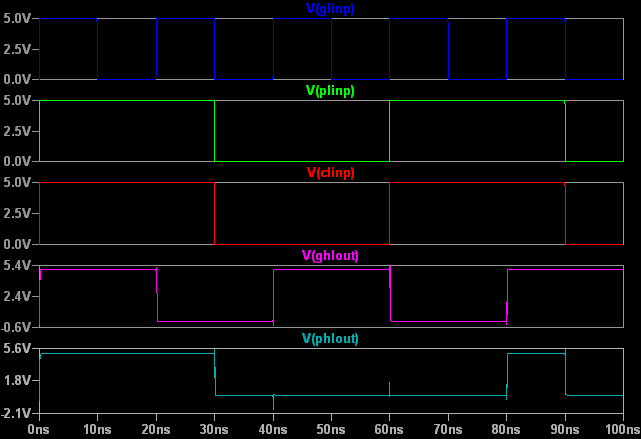


Figure 36. CLA Block Test Plot via LTspice

**Test Spicecode for layout and schematic**

\* Spice Code nodes in cell cell ' 1Bit CLA FA{lay}'

\* Spice Code nodes in cell cell '1Bit CLA FA{sch}'

vdd Vdd 0 DC 5

vdd Vdd 0 DC 5

vin GhInp 0 DC 5 pulse(0 5 10f 0.5f 0.5f 20n 40n)

vin2 PhInp 0 DC 5 pulse(0 5 10f 0.5f 0.5f 40n 80n)

vin3 GlInp 0 DC 5 pulse(0 5 10f 0.5f 0.5f 10n 20n)

vin4 PlInp 0 DC 5 pulse(0 5 10f 0.5f 0.5f 30n 60n)

vin5 ClInp 0 DC 5 pulse(0 5 10f 0.5f 0.5f 30n 60n)

.tran 0 100ns

.include C5\_models.txt

.END

### 3.2.f 8 Bit CLA Adder

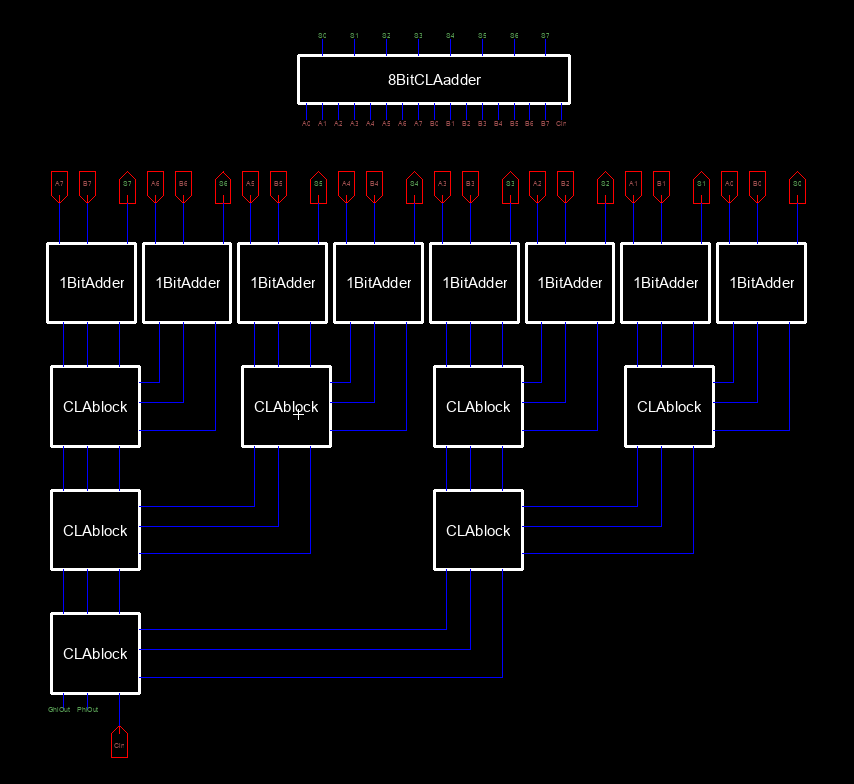


Figure 37. 8Bit CLA Adder Schematic and icon view

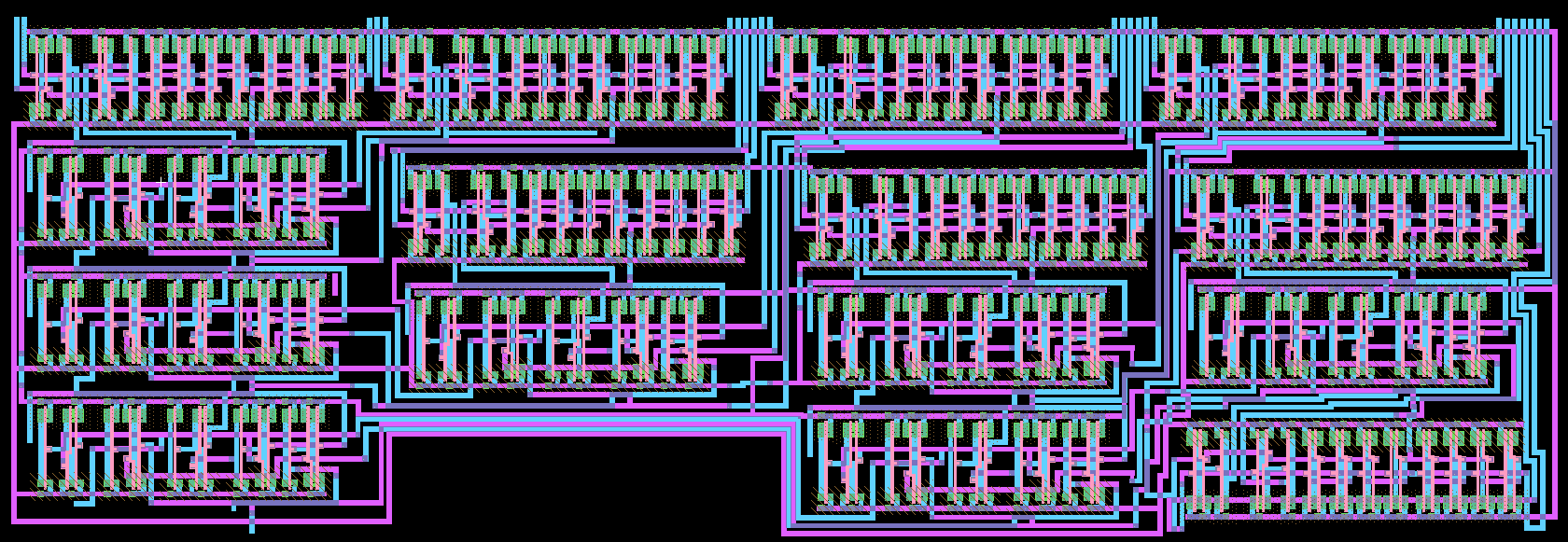


Figure 38. 8Bit CLA Layout

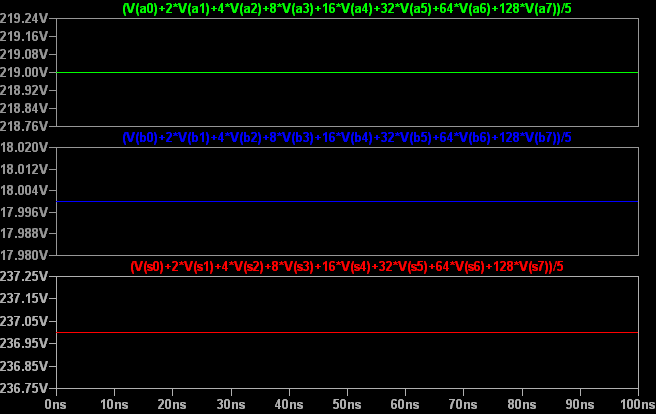


Figure 30. CLA Block Test Plot via LTspice

**Test Spicecode for layout and schematic**

\* Spice Code nodes in cell cell ' 1Bit CLA FA{lay}'

\* Spice Code nodes in cell cell '1Bit CLA FA{sch}'

vdd Vdd 0 DC 5

\*A Bits

vin A0 0 DC 5

vin2 A1 0 DC 5

vin3 A2 0 DC 0

vin4 A3 0 DC 5

vin5 A4 0 DC 5

vin6 A5 0 DC 0

vin7 A6 0 DC 5

vin8 A7 0 DC 5

\*B Bits

vin9 B0 0 DC 0

vin10 B1 0 DC 5

vin11 B2 0 DC 0

vin12 B3 0 DC 0

vin13 B4 0 DC 5

vin14 B5 0 DC 0

vin15 B6 0 DC 0

vin16 B7 0 DC 0

cload S0 0 5fF

.tran 0 100ns

.include C5\_models.txt

.END

### 3.2.g 16 Bit Mixed Adder

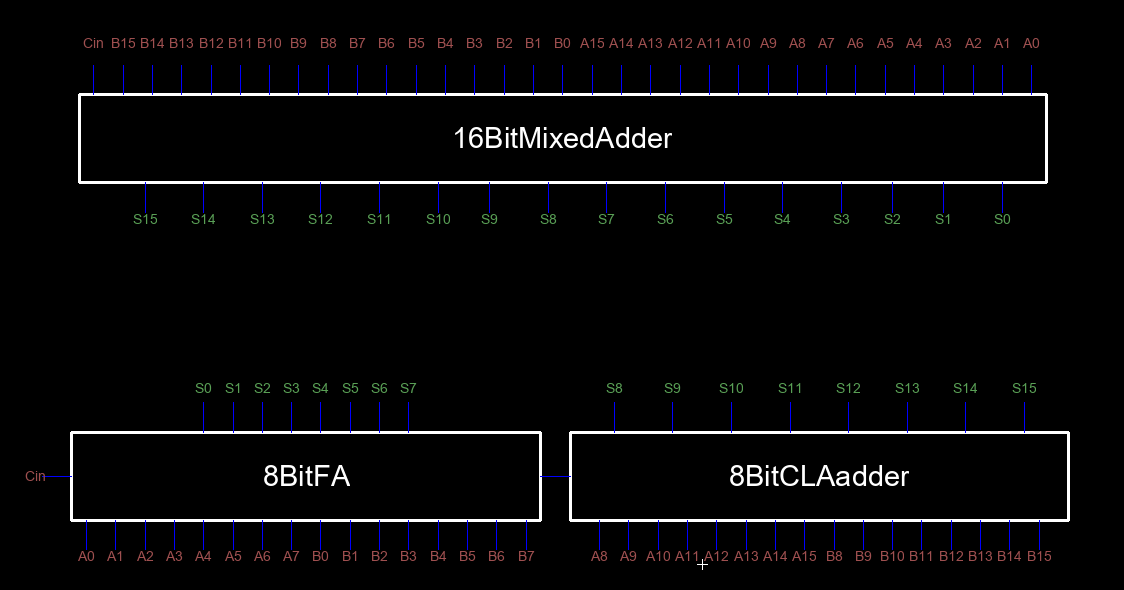


Figure 37. 8Bit CLA Adder Schematic and icon view

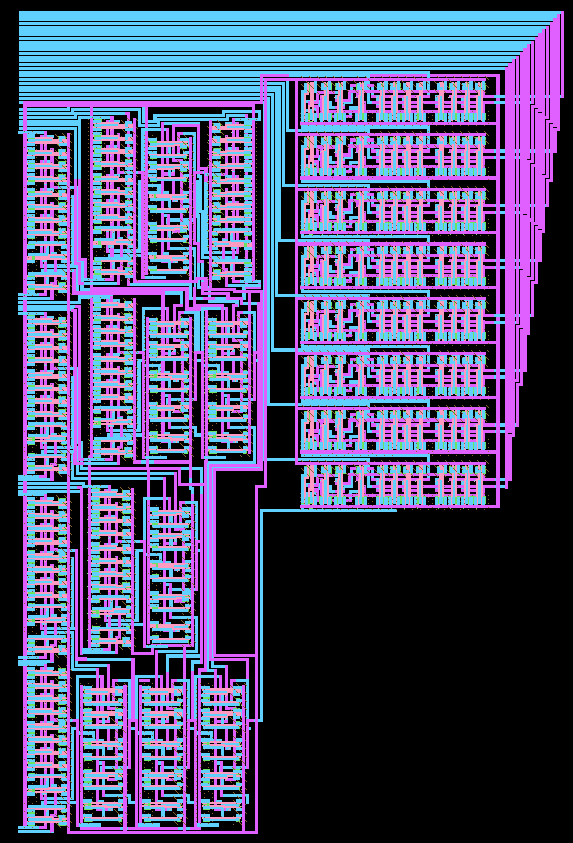


Figure 38. 8Bit CLA Layout

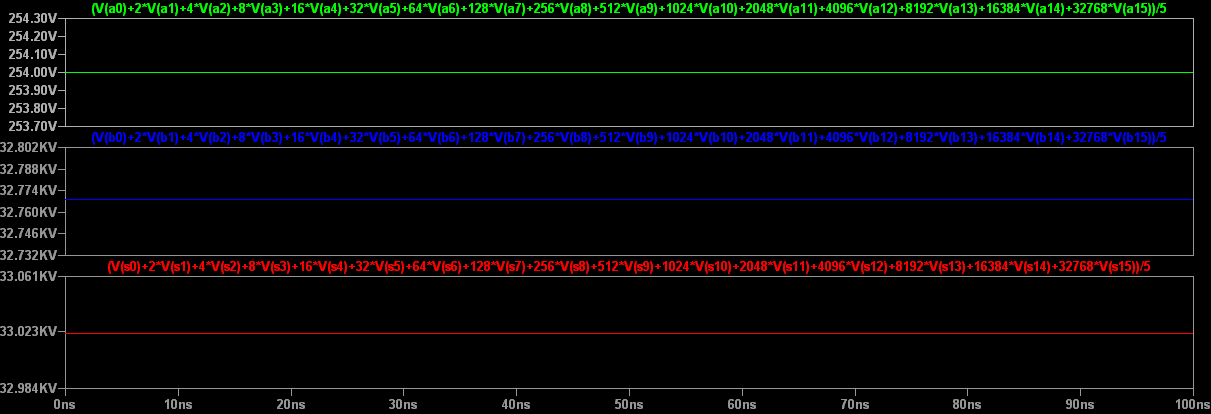


Figure 30. 16Bit adder Test Plot via LTspice

**Test Spicecode for layout and schematic**

\* Spice Code nodes in cell cell ' 1Bit CLA FA{lay}'

\* Spice Code nodes in cell cell '1Bit CLA FA{sch}'

vdd Vdd 0 DC 5

\*A Bits

vin A0 0 DC 0

vin2 A1 0 DC 5

vin3 A2 0 DC 5

vin4 A3 0 DC 5

vin5 A4 0 DC 5

vin6 A5 0 DC 5

vin7 A6 0 DC 5

vin8 A7 0 DC 5

vin9 A8 0 DC 0

vin10 A9 0 DC 0

vin11 A10 0 DC 0

vin12 A11 0 DC 0

vin13 A12 0 DC 0

vin14 A13 0 DC 0

vin15 A14 0 DC 0

vin16 A15 0 DC 0

\*B Bits

vin17 B0 0 DC 0

vin18 B1 0 DC 0

vin19 B2 0 DC 0

vin20 B3 0 DC 0

vin21 B4 0 DC 0

vin22 B5 0 DC 0

vin23 B6 0 DC 0

vin24 B7 0 DC 0

vin25 B8 0 DC 0

vin26 B9 0 DC 0

vin27 B10 0 DC 0

vin28 B11 0 DC 0

vin29 B12 0 DC 0

vin30 B13 0 DC 0

vin31 B14 0 DC 0

vin32 B15 0 DC 5

.tran 10ns 100ns

.include C5\_models.txt

.END

# REFERENCES

[1] About Electric.(May 2013).

[2] R. J. Baker, CMOS: Circuit Design, Layout, and Simulation, John Wiley & Sons, 2010, pp. 6-12.

[3] Professor F. Gebali Ripple Carry and Carry Lookahead Adders

# Figures

[Figure 1. One Bit Adder 2](#_Toc535198332)

[Figure 2. 4Bit carry look ahead implementation detail 3](#_Toc535198333)

[Figure 1. NAND2 Gate Schematic and icon view Figure 2. NAND2 Gate Layout Design 4](#_Toc535198334)

[Figure 3. nand2 gate tests via Ltspice. (For both schematic and layout) 4](#_Toc535198335)

[Figure 4. NAND3 Gate Schematic and icon view Figure 5. NAND3 Gate Layout Design 5](#_Toc535198336)

[Figure 6. Nand3 gate test plot using Ltspice.(For both schematic and layout) 6](#_Toc535198337)

[Figure 7.Inverter Schematic and icon view Figure 8. Inverter Layout Design 6](#_Toc535198338)

[Figure 9.Inverter test plot using Ltspice. (For both schematic and layout). 7](#_Toc535198339)

[Figure 10.Nor2 Gate Schematic and icon view Figure 11.Nor2 Gate Layout Design 7](#_Toc535198340)

[Figure 12.Nor2 gate test plot using Ltspice. (For both schematic and layout) 8](#_Toc535198341)

[Figure 13. And2 Gate Schematic and icon view Figure 14. And2 Gate Layout Design 8](#_Toc535198342)

[Figure 15. And2 gate test plot using Ltspice. (For both schematic and layout) 9](#_Toc535198343)

[Figure 16. Or2 Gate Schematic and icon view Figure 17. Or2 Gate Layout Design 9](#_Toc535198344)

[Figure 18. Or2 gate test plot using Ltspice. (For both schematic and layout) 10](#_Toc535198345)

[Figure 19. Xor Gate Schematic and icon view 11](#_Toc535198346)

[Figure 20. Xor Gate Layout Design 11](#_Toc535198347)

[Figure 21. XOr gate test plot using Ltspice. (For both schematic and layout) 12](#_Toc535198348)

[Figure 22. 1Bit FullAdder Schematic and icon view 13](#_Toc535198349)

[Figure 23. 1Bit FullAdder Layout 13](#_Toc535198350)

[Figure 24.1Bit FullAdder Test Plot 14](#_Toc535198351)

[Figure 25. 4Bit FullAdder Schematic and icon view 15](#_Toc535198352)

[Figure 26. 4Bit FullAdder Layout 16](#_Toc535198353)

[Figure 27.4Bit FullAdder Test Plot 17](#_Toc535198354)

[Figure 28. 8Bit Adder Schematic and icon view 18](#_Toc535198355)

[Figure 29. 8Bit Adder Layout 19](#_Toc535198356)

[Figure 30. 8Bit Adder Test Plot via LTspice 20](#_Toc535198357)

[Figure 31. 1Bit CLA Adder Schematic and icon view 21](#_Toc535198358)

[Figure 32. 1Bit CLA Adder Layout 21](#_Toc535198359)

[Figure 33. 8Bit Adder Test Plot via LTspice 22](#_Toc535198360)

[Figure 34. CLA Block Schematic and icon view 23](#_Toc535198361)

[Figure 35. CLA Block Layout 24](#_Toc535198362)

[Figure 36. CLA Block Test Plot via LTspice 24](#_Toc535198363)

[Figure 37. 8Bit CLA Adder Schematic and icon view 25](#_Toc535198364)

[Figure 38. 8Bit CLA Layout 26](#_Toc535198365)

[Figure 30. CLA Block Test Plot via LTspice 26](#_Toc535198366)

[Figure 37. 8Bit CLA Adder Schematic and icon view 27](#_Toc535198367)

[Figure 38. 8Bit CLA Layout 28](#_Toc535198368)

[Figure 30. 16Bit adder Test Plot via LTspice 29](#_Toc535198369)

Contents

[1.INTRODUCTION 1](#_Toc535199188)

[2. LITERATURE REVIEW 2](#_Toc535199189)

[2.1 Ripple Carry Adder 2](#_Toc535199190)

[2.2 Carry lookahead adder (CLA) 2](#_Toc535199191)

[3.METHODOLOGY 4](#_Toc535199192)

[3.1. Logic Gates 4](#_Toc535199193)

[3.1.a 2-input NAND Gates 4](#_Toc535199194)

[3.1.b 3-input NAND Gates 5](#_Toc535199195)

[3.1.c Inverter 6](#_Toc535199196)

[3.1.d 2-Input NOR Gate 7](#_Toc535199197)

[3.1.e 2-Input AND Gate 8](#_Toc535199198)

[3.1.f 2-Input OR Gate 9](#_Toc535199199)

[3.1.g XOR Gate 11](#_Toc535199200)

[3.2. Modules 13](#_Toc535199201)

[3.2.a 1Bit Full Adder 13](#_Toc535199202)

[3.2.b 4Bit Full Adder 15](#_Toc535199203)

[3.2.c 8Bit Adder 18](#_Toc535199204)

[3.2.d 1Bit CLA Adder 21](#_Toc535199205)

[3.2.e CLA Block 23](#_Toc535199206)

[3.2.f 8 Bit CLA Adder 25](#_Toc535199207)

[3.2.g 16 Bit Mixed Adder 27](#_Toc535199208)

[REFERENCES 30](#_Toc535199209)

[Figures 31](#_Toc535199210)