Design of 16 – Bit Comparator using three

Mariam Turk[1] *– 1211115 , Noura Manassra*[2] *1212359 , Yara Mukho*[3]1211269

Department of Electrical and Computer Engineering, Faculty of Engineering and Technology Birzeit University

8 / 18 / 2024

KEYWORD

Nand
Nor
4-bit And
4-bit Or
1-bit compartor
4-bit compartor
16-bit compartor
NMOS
PMOS
CMOS
Schematic
Layout
Wires
Power
Delay
Sizing
Area
Length
Width
Complexity
Microprocessor
Kmap
Performance
Invertor
Gate Level
Transistor Level
Moore's Law
P-Type
N-Type
Contact
Drain
Source
Gate
Spacing
Simulations
Codes

ABSTRACT

This research paper designs and implements a 16-bit comparator using complementary metal-oxide-semiconductor (CMOS) devices, PMOS and NMOS transistors, together with basic logic gates. CMOS technology is preferred because it has several advantages that include low power consumption, high noise immunity and scalability, thus making it the best choice for modern digital systems . The design of a 1-bit comparator using basic logic gates like AND, OR and NOT gates is described by the paper. It is from this 1-bit comparator that a 4-bit by 1-bit comparator circuit is constructed. Lastly the 4-bit by 1 bit comparator is expanded to form a 16-bit by 4 bit comparator.

The purpose of this sixteen bits comparator will be to compare two sixteen bits binary numbers and give outputs indicating relations between them such as "greater than," "less than" or "equal." The paper provides an in-depth look at designing, implementing and performance analysis of one-, four- and sixteen bits comparators . According to the results obtained it can be concluded that the use of CMOS devices combined with basic logic gates can successfully implement a six-teen bit comparator which demonstrates that this approach is feasible and effective. This paper presents a comparative design suitable to different digital systems including microprocessors .

1- Theory : Integrated Circuits

Definition

Integrated Circuits: The Pervasive Foundations of Modern Electronics

Nowadays, electronics have been widely integrated into almost every part of our lives. From our kitchens' microwaves to the satellites revolving around our globe, electronic devices are everywhere. Even when we are asleep, digital technologies in devices track our acoustics, haptics and analytics.[1]

Although the specific systems that power, connect and mobilize our daily lives can be very different, almost all electronic devices share a common basic building block called integrated circuit (IC), commonly known as "chip". [1]

The ICs are manufactured from silicon which is a semiconductor material where small electrical components called transistors are created. These transistors are then connected together by layers of wires that are laid on top of the surface of silicon. [1]

This combination of semiconductor materials, transistors and interconnects allows ICs to perform an astonishingly wide range of complex functions within a remarkably small and efficient area. Starting from data processing to governing electromechanical systems, the all-powerful IC represents the foundation upon which our contemporary world packed with electronics is founded. [1]

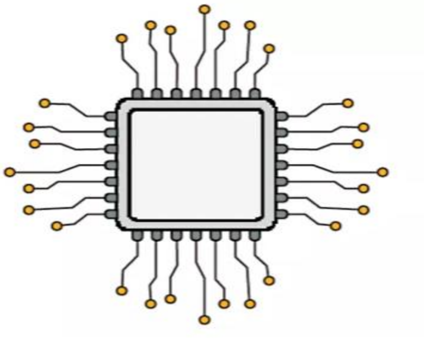


Figure 1 : IC

What integrated circuits do ?!

For a range of present-day electronic appliances and systems, Integrated Circuits (ICs) serve as the basic components that empower them. They are located at the core of computers, forming the computational power. In mobile phones, they represent the communicative part that allows for uninterrupted communication. Besides, ICs also act as controlling units in automation systems thus enhancing accuracy and efficiency. [2]

Nonetheless, ICs are not just about these well-known applications. They equally function as signal amplifiers, manage power distribution and convert various energy types. As a matter of fact, it is ICs that lay foundation for rapid execution of tasks by electronic devices. [2]

Integrated circuits enable our modern world to operate through processing data, transmitting information or regulating electrical systems among others. But those tiny things do more than meet the eye and that has constantly made them a must component in many other electronics which keep expanding every other time. [2]

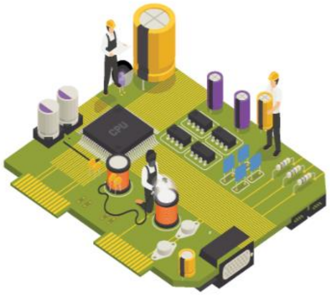


Figure 2 : How IC work

History of IC

The history of ICs is marked by milestone inventions, which have been improved upon relentlessly. It all started when John Ambrose Fleming invented the vacuum tube in 1897, hence setting the stage for today's electronics.[3]

1947 was an important date when William Shockley and other team members at Bell Labs invented the transistor. It is a compact, solid-state device, replacing effectively the bulky and power-hungry vacuum tubes, hence marking the beginning of a new electronic advancement. [3]

The next big breakthrough was in 1959 with the invention of the integrated circuit. This was a new concept where an entire array of electronic devices, from transistors to resistors, would be made on a single semiconductor wafer. This used less power; more importantly, they gave very smooth outputs; and thus, they found their presence everywhere in a lot of electronic devices. [3]

From the vacuum tube through the transistor, culminating with the integrated circuit, each of these innovations changed everything. As the years went by and ICs kept getting smaller, more efficient, and more capable, they became the basic building blocks powering the vast array of today's electronic devices, spanning from computers and smartphones to industrial automation systems and more. [3]

This development in integrated circuit technology is a testimonial of the human genius and man's unending quest to push the envelope in the world of electronics. [3]

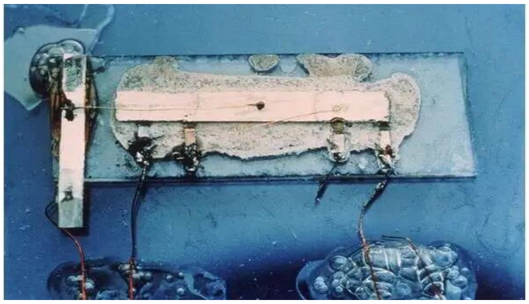


Figure 3 : The First IC

What is an Integrated Circuit?

Integrated circuits integrate large numbers of components, including transistors, capacitors, resistors, and others, on one chip of semiconductor material. This makes the IC perform a wide range of complicated operations, especially with oscillations, flip-flopping, and amplification. [3]

ICs can either be analog or digital. Analog ICs would have continuous valued outputs, which would vary with the input signals. Digital ICs operate with discrete valued binary signals. Some ICs include both parts that are analog and others that are digital. [3]

ICs have, therefore, been of paramount importance in the progress made by many electronic devices, like computers and mobile phones. With increased integration and capability, changes were brought about in their application; thus, driving changes in technology. [3]

The journey of ICs has witnessed giant strides, revolutionizing the realm of electronics. This evolution could be used to be summarized in the following phases:

Small-Scale Integration (SSI)

The appearance of SSI technology greeted the dawn of the 1960s when 10 to 100 transistors were allowed to be fabricated on a single chip. This thus enabled the pulsations for the progressive evolution of basic logic gates and flip-flops. [4]

By the end of the 1960s, an alternative to MSI already existed, which included 100 to 1,000 transistors on a chip, which allowed one to develop more complex devices, including multiplexers, counters, and decoders. [3]

Large-Scale Integration (LSI)

Developed in the 1970s, LSI, as the name implies, has much greater capacity. It would allow including 1,000 to 20,000 transistors on one single chip. It allowed the manufacturer to develop considerably more complex devices, including RAM, ROM, and microprocessors. [4]

Very-Large Scale Integration

The 1980s saw the rise of VLSI, where 20,000 to 50,000 transistors could be placed on one chip. It became a gigantic platform upon which it was possible to build RISC processors, 16-bit and 32-bit microprocessors, and digital signal processors (DSPs). [4]

Ultra-Large Scale Integration (ULSI)

After 1985, the industry finally achieved a true breakthrough in the manufacturing of over 50,000 to even billions of transistors on a chip. Such a development created a path to instituting 64-bit processors and other technologies related to advanced computing. [4]

Each of the above phases of this evolution has expanded the capabilities of integrated circuits and accelerated further electronic device and system development that currently characterized the modern world around us. [4]

Moore’s Law and the future of IC

In the 1960s, the number of components developed on a single chip started to increase exponentially with the introduction of the integrated circuits. The trend that has now become famously known as Moore's Law was first noticed by Gordon Moore, a co-founder of Intel. He commented that the number of transistors on a chip had doubled in number about every one to two years, increasing the computational power and capacities of the devices in extraordinarily huge amounts. [5]

But in recent years, Moore's Law has started to face some challenges. As of 2006 integrated circuits could hold over 300 million transistors, while, chips today only carry about 1 billion, off the predicted 4-5 billion if applying Moore's Law. [5]

Space Problem: When too many components are squished together in the same physical space, eventually the question becomes not whether a rogue atom will come along and cause a chip to fail but when, with reliability problems as a consequence. [5]

Overheating: The high transistor density creates a lot of heat, which can be difficult to manage — an issue that can be exacerbated by the small size of modern devices. [5]

Weird behavior: The tighter the transistors are packed, the more the quantum mechanical effects kick in—they make the electrons behave erratically, simultaneously risking data corruption. [5]

The scientists are also developing an approach through other materials/technologies instead of silicon-based typical electronics. Some of the salient solutions are enlisted below:

Graphene: This is a very conductive material; it can replace the silicon one day, but so far, mass-producing it reliably has been more of a challenge. [5]

Photonics: A technique which may very soon provide for faster computing with much more efficiency, as it uses photons—light particles—for carrying data instead of electrons. [5]

Then it is yet to be said how these alternative approaches are going to scale up the computing power needed for computers in the future, enabling the exponential growth envisioned by Moore's Law. These questions are answered by how breakthrough and new innovative computing technology continues to overcome these technical challenges and finds new ways of packing in more and more computational power into smaller, more efficient devices. [5]

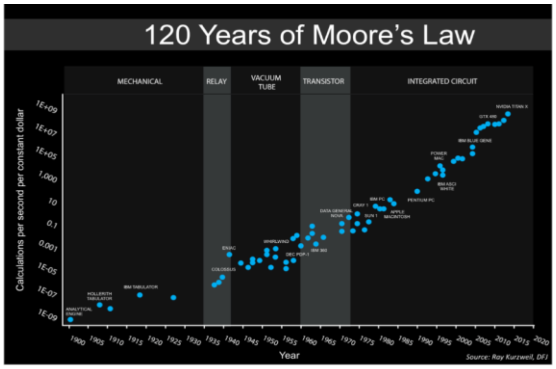


Figure 4 : Moore's Law

How an IC Is Made

At the heart of an integrated circuit, IC, lie layers of silicon wafers, the semiconductors, combined with copper interconnections that result in the miniature size of electronic components in use today—transistors, resistors, diodes, among others. [5]

The secret to the development of such integrated circuits lies in the process of doping semiconductor materials. Doping is the intentional introduction of impurities into a semiconductor, normally silicon, to modify its electrical properties. [5]

There are two major kinds of semiconductor doping:

N-Type Doping:

It means that the addition of an element, such as antimony, to intrinsic silicon introduces extra electrons in the semiconductor material. In this way, n-type silicon is formed with an abundance of negatively charged electrons that can move and conduct electric current. [5]

P-Type Doping:

On the other hand, introducing an element like boron to the silicon removes some of its electrons, thereby creating "holes" that act as positively charged carriers. This will yield p-type silicon, which has an excess of positive charge carriers. [5]

By precisely controlling the placement of these n-type and p-type doped regions within the silicon wafer, it lays the base for a wide variety of electronic components. The interface between these regions of n-type and p-type doped silicon provides the means for electrons to flow, in effect serving as the basis for the on/off, 1/0 binary function that underlies the transistor, the integrated circuit, and modern digital electronics. [5]

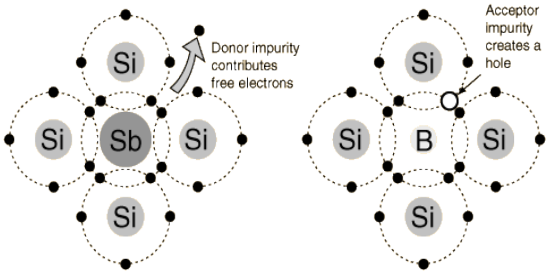

Si
Si
Sb
Si
Si
Donor impurity contributes free electrons
Si
Si
B
Si
Si
Acceptor impurity creates a hole

Figure 5 : n-type (left) and p-type (right) doping

This complex structure of doped semiconductor layers, plus the copper interconnections, makes for the complete die—the functional unit of an integrated circuit. The die forms the climax of a labyrinthine and intricate process of manufacture, which by tight control over the doping, layering, and patterning of these materials realizes the miniaturization of electronic elements beyond what is possible on a breadboard. [5]

The capability to manage semiconductor doping has been the most significant innovation that provided impetus to the extraordinary advancement of integrated-circuit technology, which has powered the ever-ubiquitous electronic devices in our lives today. [5]

Manufacturing

Integrated circuit (IC) design and manufacturing process involves several steps to create complex electronic circuits on a small semiconductor chip:

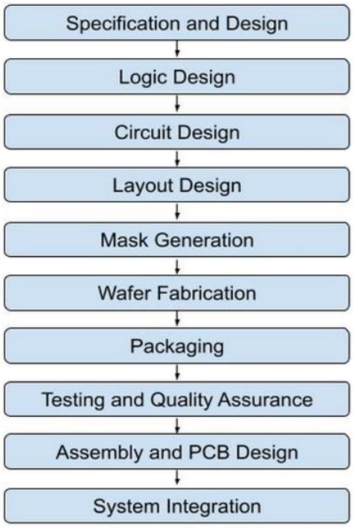


Figure 6 : Manufacturing

The development of an integrated circuit is complex and involves a couple of highly meticulous processes in steps, each building on the other. Let's delve into the step-by-step journey of bringing an IC from concept to reality:

Specification and Design:

It begins with the definition of specifications and requirements for the desired IC. A detailed circuit diagram would be drawn where the components of the device, such as transistors, resistors, capacitors, and all interconnections are identified. Normally, EDA software is used for such purpose. [6]

Logic Design

This stage would involve conversion of the circuit diagram to logic design, using a hardware description language, for example, VHDL or Verilog. In this stage, the behavior and function of the circuit are specified. [6]

Circuit Design:

The description of the logic design is translated into a detailed description of a circuit design, clearly showing what components are used and how they are connected. This circuit is then simulated to determine whether it works and if it operates at the desired speed. [6]

Layout Design:

The circuit is then translated into a Layout, which specifies the geometric pattern for exact placement and dimensions of all the components and their interconnections to be formed on the chip. Special layout tools are used to perform this layout. [6]

Mask Generation:

Masks are designed based on layout design. These masks are patterns that define the shapes and locations of different features on the chip. So, the fabricated circuit design is being printed on the silicon wafer, at time of fabrication process. [6]

Wafer fabrication :

These are masks placed in the silicon wafer to form the required pattern and for the deposited and etched layers, doping, and for making the required transistor structures; there are several ICs in one wafer. [6]

Packaging :

After the fabrication of wafers, it involves that the fabricated individual ICs be separated and then packaged into appropriate housings that protect them and realize their electrical connection. This packaging stage includes attaching leads or balls that shall be used to make the external connections. [6]

Testing and Quality Assurance

The ICs are taken to receive electrical tests to specify their conformance to the specifications. The test shall be a functional verification, case performance evaluation, and reliability estimation. [6]

Assembly and Design of the PCB:

These ICs are then mounted onto printed circuit boards that provide electrical connectivity and mechanical support. Component placement, routing of the electrical connections, signal integrity, power distribution, and thermal management constitute the design process of a PCB. [6]

System Integration:

Finally, the ICs mounted on the PCB are integrated with other components and subsystems to make the complete electronic product. [6]

Schematic & Layout

It alone does not capture how the electronic components are physically laid out on a circuit board, nor does it account for the traces connecting the components. It also ignores layout regarding ground and power planes, all of which can be critical to the maintaining of signal integrity in a PCB. [7]

Understanding the basic concepts of electromagnetism allows one to appreciate why there is a limitation to a schematic diagram's ability to fully model how circuits will actually behave when laid out on a real PCB. Simulations run based on schematics often fail to predict actual behavior that is seen in a finished PCB. [7]

Even if simulation results do not turn up a single integrity problem, it may still result in crosstalk, unwanted resonances, and ringing. In this section we will present why these differences occur and how common layout problems can be dealt with to ensure proper signal integrity. [7]

Any differences between the behavior predicted by your schematic simulations and the actual performance of your PCB have their roots in the physical geometry of the circuits on the board. In fact, a PCB design always starts with circuit diagrams that are further developed into a schematic diagram. It is an important document that will provide the core structure for your circuit board, giving a reference for the layout of your PCB. The schematic displays each circuit and its components as they would appear in a normal 2D CAD assembly drawing. [7]

Now that you have completed the capture of the schematic, you can use a schematic import tool to translate the circuits and interconnections into a physical layout of your board. This tool creates an initial placement so you can start to place elements. This is the view of how your circuits are going to work in real life, defining the distribution of an electromagnetic field throughout your system. It defines how traces, planes in every layer, components, and other conductive elements are arranged—those that can create signal and power integrity problems, not noticed by a schematic. [7]

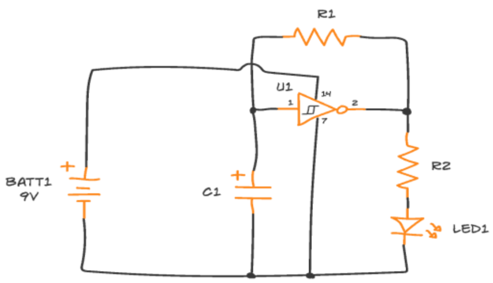


Figure 7 : Schematic

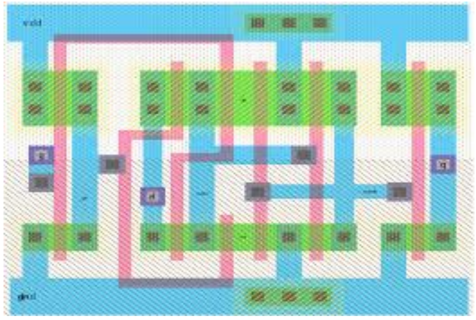
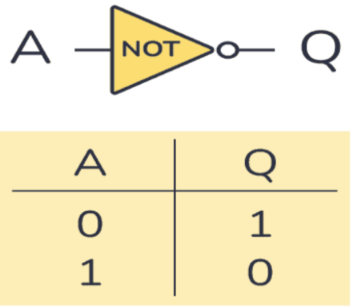


Figure 8 : Layout

2- Introduction

Inverter

A NOT gate (or inverter) is a logic gate where the output is the opposite of the input. So you can say that the output is NOT the same as the input. It's often called an *inverter* since it inverts the input.



A	Q
0	1
1	0

Figure 9 : Inverter gate and truth table

Schematic for inverter

To design an inverter in ElectricBinary 9.07, first, series connect one PMOS and one NMOS transistor. The PMOS is connected to VDD, which is set equal to 5V, and the NMOS is connected to ground as shown in Figure 10. Fix the dimensions for the transistors: PMOS of width 8 and height 2 and NMOS of width 12 and height 2. [8]

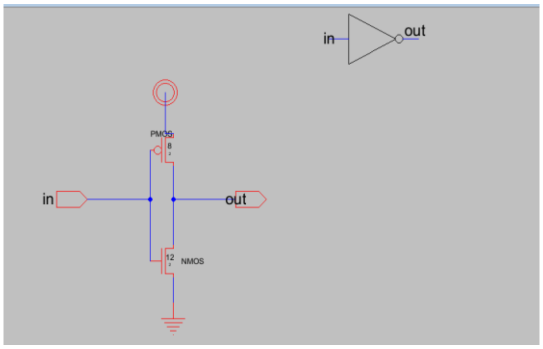


Figure 10 : Schematic for inverter

Simulation for Inverter

As shown in figure 11 , the simulation is true since , the V(out) is as expected .

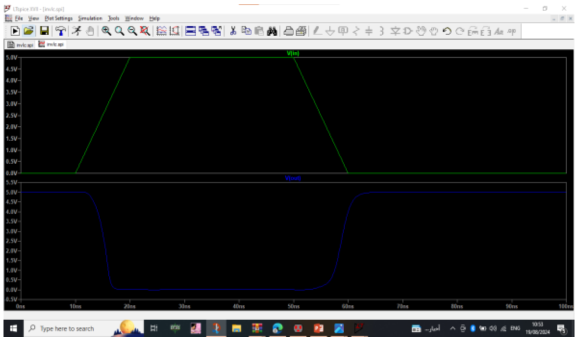


Figure 11 : Simulation for Inverter

Layout for Inverter

While designing the layout for an inverter, the PMOS and NMOS transistor's width and height are set according to the schematic so that proper functionality would not be altered. More emphasis is paid on the spacing between elements so that design rules will be followed to prevent parasitic effects. The input and output connections are adjusted to touch layout elements so that the design will fit the schematic precisely. With this method, not only will the layout function correctly, but there will also be consistency with the schematic, enabling both effective production and reliable operation .

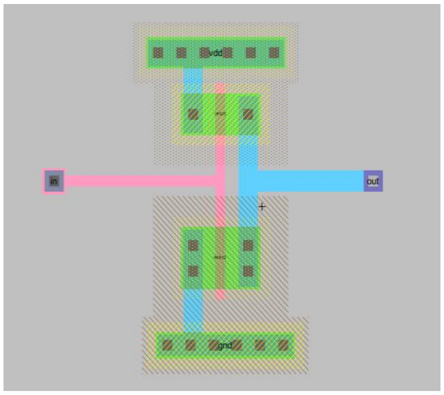


Figure 12 : layout for Inverter

NOR

A NOR gate is a logic gate where the output goes HIGH (or “1”) only if all its inputs are LOW (or “0”). The schematic symbol for a NOR gate is like the OR gate, just with a circle at the output to indicate that it’s an inverted version of OR. [9]

Schematic for NOR

To design an NOR in Electric Binary 9.07, first, series connect tow PMOS with tow Parallel NMOS transistor. The PMOS is connected to VDD, which is set equal to 5V, and the NMOS is connected to ground and tow input one output as shown in Figure 13. Fix the dimensions for the transistors: PMOS of width 4 and height 2 and NMOS of width 12 and height 2 .

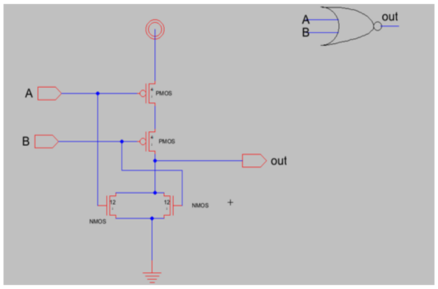


Figure 13 : Schematic for NOR

Layout for NOR

While designing the layout for a NOR gate, the width and height of the PMOS and NMOS transistors are set according to the schematic to maintain proper functionality. Special attention is given to the spacing between elements to ensure that design rules are followed, preventing parasitic effects. The input and output connections are adjusted to align with the layout elements, ensuring that the design fits the schematic precisely. This approach ensures that the NOR gate layout is not only functional but also consistent with the schematic, enabling effective production and reliable operation.

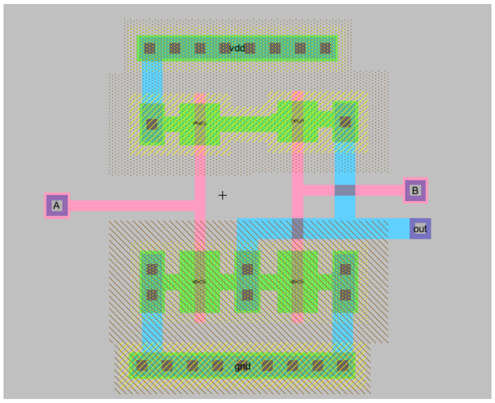
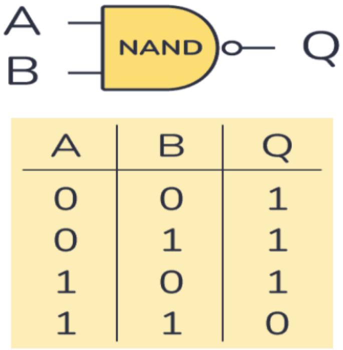


Figure 14 : layout for NOR

NAND

A NAND gate is a logic gate where the output goes LOW (or “0”) only if all its inputs are HIGH (or “1”). The schematic symbol for a NAND gate is like the AND gate, just with a circle at the output to indicate that it’s an inverted version of AND. [10]



A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Figure15 : NAND gate and truth table

Schematic for NANR

To design an NAND in Electric Binary 9.07, first, parallel connect tow PMOS with tow series NMOS transistor. The PMOS is connected to VDD, which is set equal to 5V, and the NMOS is connected to ground and tow input one output as shown in Figure 16. Fix the dimensions for the transistors: PMOS of width 4 and height 8 and NMOS of width 6 and height 2 .

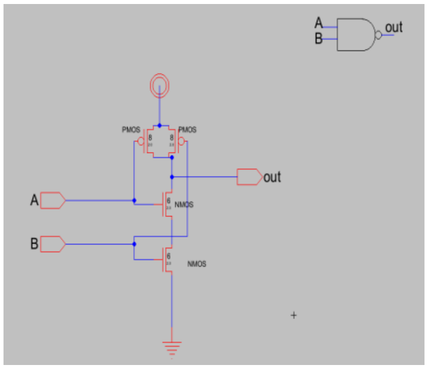


Figure 16 : Schematic for NAND

Layout for NAND

While designing the layout for a NAND gate, the width and height of the PMOS and NMOS transistors are set according to the schematic to ensure proper functionality. Special attention is given to the spacing between elements to adhere to design rules and prevent parasitic effects. The input and output connections are adjusted to align with the layout elements, ensuring that the design matches the schematic precisely. This approach ensures that the NAND gate layout is both functional and consistent with the schematic, facilitating effective production and reliable operation.

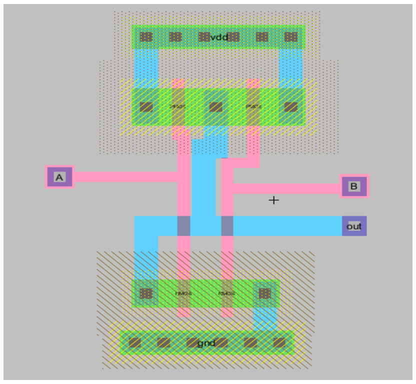
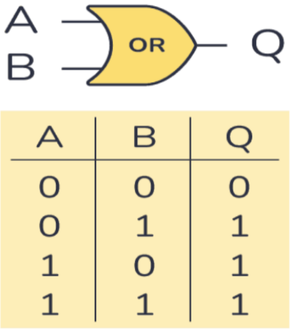


Figure17 : layout for NAND

4-bit OR

An OR gate is a logic gate where the output goes HIGH (or "1") if any of its inputs are HIGH. So if A **OR** B is HIGH, the output Q also becomes HIGH. [11]

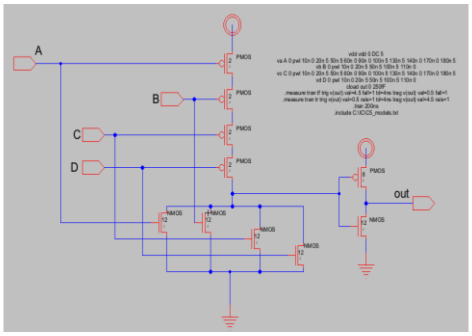


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Figure 18 : OR gate and truth table

Schematic for 4-bit OR

To design an 4-bit OR in Electric Binary 9.07, first, series connect four PMOS with four parallel NMOS transistor. The PMOS is connected to VDD, which is set equal to 5V, and the NMOS is connected to ground and four input one output as shown in Figure 19. Fix the dimensions for the transistors: PMOS of width 2 and height 2 and NMOS of width 12 and height 2 .



```
vdd vdd 0 DC 5
va A 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5
vb B 0 pwl 10n 0 20n 5 50n 5 100n 5 110n 0
vc C 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5
vd D 0 pwl 10n 0 20n 5 50n 5 100n 5 110n 0
cload out 0 250fF
.measure tran tf trig v(out) val=4.5 fall=1 td=4ns trag v(out) val=0.5 fall=1
.measure tran tr trig v(out) val=0.5 rais=1 td=4ns trag v(out) val=4.5 rais=1
.tran 200ns
.include C:\IC\C5_models.txt
```

Figure 19 : Schematic for 4-bit NOR

Layout for 4-bit OR

While designing the layout for a 4-bit OR gate, the width and height of the PMOS and NMOS transistors are set according to the schematic to ensure proper functionality. Special attention is given to the spacing between elements to adhere to design rules and prevent parasitic effects. The input and output connections are adjusted to align with the layout elements, ensuring that the design matches the schematic precisely. This approach ensures that the 4-bit OR gate layout is both functional and consistent with the schematic, facilitating effective production and reliable operation.

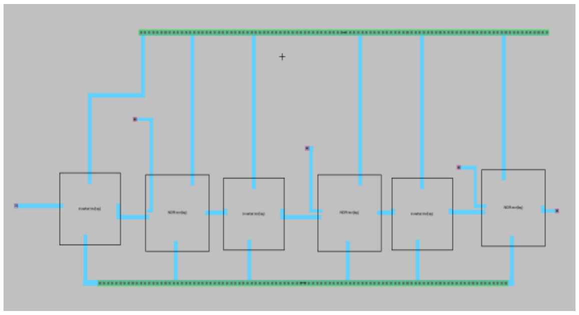
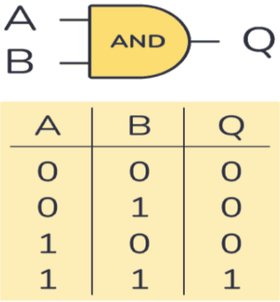


Figure 20 : layout for 4-bit OR

4-bit AND

An AND gate is a logic gate where the output goes HIGH (or "1") only if all its inputs are HIGH. So if the inputs A **AND** B are HIGH, the output Q will also be HIGH. [12]

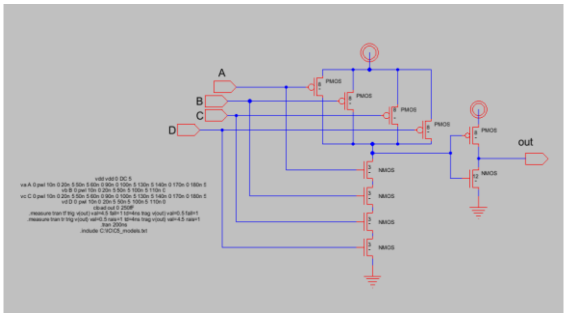


A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Figure 21 : AND gate and truth table

Schematic for 4-bit AND

To design an 4-bit AND in Electric Binary 9.07, first, parallel connect four PMOS with four series NMOS transistor , and parallel connect one PMOS one NMOS . The PMOS is connected to VDD, which is set equal to 5V, and the NMOS is connected to ground and four input one output as shown in Figure 22. Fix the dimensions for the transistors: PMOS of width 8 and height 2 and NMOS of width 12 for one NOMS for 4 NMOS width is 3 and height 2 .



```
vdd vdd 0 DC 5
va A 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5
vb B 0 pwl 10n 0 20n 5 50n 5 100n 5 110n C
vc C 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5
vd D 0 pwl 10n 0 20n 5 50n 5 100n 5 110n 0
cload out 0 250fF
.measure tran tf trig v(out) val=4.5 fall=1 td=4ns trag v(out) val=0.5 fall=1
.measure tran tr trig v(out) val=0.5 rais=1 td=4ns trag v(out) val=4.5 rais=1
.tran 200ns
.include C:\IC\C5_models.txt
```

Figure 22 : Schematic for 4-bit AND

Layout for 4-bit AND

While designing the layout for a 4-bit AND gate, the width and height of the PMOS and NMOS transistors are set according to the schematic to ensure proper functionality. Special attention is given to the spacing between elements to adhere to design rules and prevent parasitic effects. The input and output connections are adjusted to align with the layout elements, ensuring that the design matches the schematic precisely. This approach ensures that the 4-bit AND gate layout is both functional and consistent with the schematic, facilitating effective production and reliable operation.

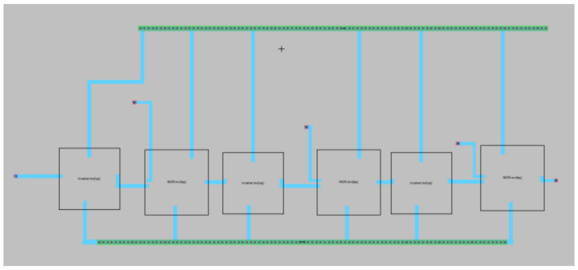


Figure 23 : layout for 4-bit AND

1-bit Comparator

A digital magnitude comparator is a combinational circuit designed to compare two digital or binary numbers in finding out whether one is less than, equal to, or greater than another. The circuit has two input numbers, A and B, and gives three output signals: one if A > B, another if A = B, and the third if A < B. [13]

Comparisons start from the MSB of the two numbers and move towards the LSB. The comparison for each bit position involves extracting the bits at that position for A and B and comparing them. If the bit of A is 1 while the bit of B is 0, A > B is asserted and the circuit concludes that A is greater. Contrarily, if the bit from B is greater, then A < B is set to 1, and the circuit concludes that A is less than B. [13]

If they are equal at that position, the comparator circuit moves on to the next bit position and continues the comparison. This continues until all bits have been compared. If at any moment a decision is reached, A is greater than B or A is less than B, the process is terminated and an appropriate output is produced. In case all the bits are equal, the circuit outputs A equating to B, stating that the two numbers are equal. [13]

Realizations of a magnitude comparator can be done by combining XOR, AND, and OR gates or even by a cascaded arrangement of full adders. The implementation will depend on factors such as speed, complexity, and power efficiency. [13]

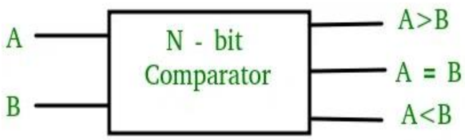
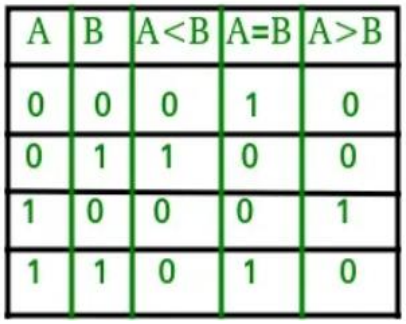


Figure 24 : N – bit Comparator

A comparator used to compare two bits is called a single-bit comparator. It consists of two inputs each for two single-bit numbers and three outputs to generate less than, equal to, and greater than between two binary numbers. [13]

The truth table for a 1-bit comparator is given below.



A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

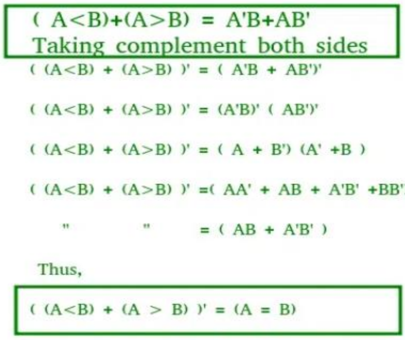
Figure 25 : 1-Bit Magnitude Comparator

From the above truth table logical expressions for each output can be expressed as follows. [13]

A>B: AB'

A<B: A'B

A=B: A'B' + AB



(A<B)+(A>B) = A'B+AB'
Taking complement both sides

((A<B) + (A>B))' = (A'B + AB')'

((A<B) + (A>B))' = (A'B)' (AB')'

((A<B) + (A>B))' = (A + B') (A' +B)

((A<B) + (A>B))' =(AA' + AB + A'B' +BB')

" " = (AB + A'B')

Thus,

((A<B) + (A > B))' = (A = B)

Figure 26 : Derivation of 1-Bit Magnitude Comparator

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below . [13]

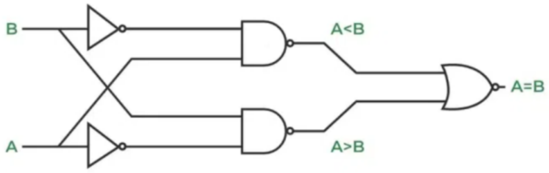


Figure 27 : Logic Circuit

Schematic for 1-bit comparator

Designing a 1-bit comparator in Electric Binary 9.07:

Inverter (first stage of the circuit):

An inverter is used to form the first part of the circuit. This includes a PMOS transistor connected in series with an NMOS transistor. The PMOS transistor forms the connection to VDD (5V), while the NMOS transistor connects to ground. The input to the inverter comes from the bit under comparison.

Dimension: Make the PMOS transistor 2 in width and 2 in height, and the NMOS transistor 12 in width and 2 in height.

NAND Gate State 2:

Now, at the output of the inverter, connect a NAND gate. The NAND gate input will be connected from the output of the inverter and another from the compared bit.

Dimensions: The NAND will be composed of two PMOS devices in parallel and two NMOS devices in a series. The PMOS devices will have a width of 2 and a height of 2, and the NMOS devices will have a width of 12 and a height of 2.

Inverter: Stage 3:

This output is then passed on to another inverter. This inverter inverts the output from the NAND gate to contribute to the logic needed for the comparator.

Dimension: This should be the same dimensions as the first inverter. The dimension for the PMOS transistor is 2 width and 2 height, and for the NMOS transistor, it is 12 width and 2 height.

NOR Gate (Stage 4):

Finally, attach a NOR gate at the output of the second inverter. The NOR gate will also have two inputs: one line coming from the output of the second inverter and another line coming from the bit being compared.

Dimensions: A NOR gate can be created by connecting two PMOS transistors in series and two NMOS transistors in parallel. The width of the PMOS transistor could be 2, and the height would be 2. For the NMOS transistor, the width would be 12, and the height would be 2.

Using this setup, design a comparator circuit that compares two 1-bit inputs using only inverters, NAND, and NOR gates in Electric Binary 9.07. It is with careful choice of transistor dimensions that the circuit works properly to conform to the desired schematic.

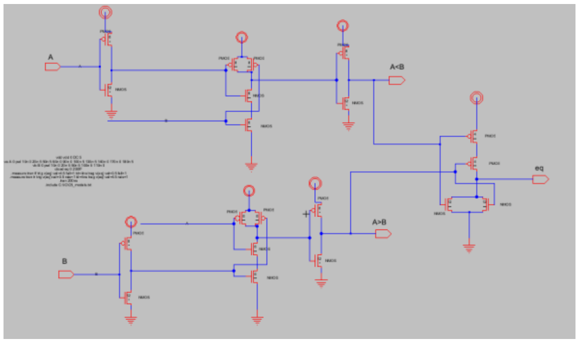


Figure 28 : Schematic for 1-bit comparator

Layout for 1-bit comparator

To do the layout , use the layout of inverter , layout for NAND , layout for NOR , then connect the inverter , inverter with NAND and NAND with inverter and finaly Inverter with NOR .

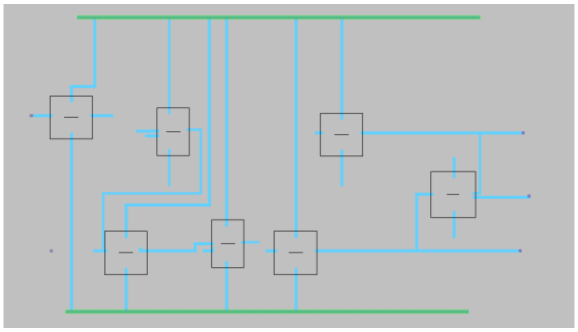


Figure 29 : layout for 1-bit comparator

4-bit Comparator

A magnitude comparator with 4 bits is a digital circuit used to compare two binary numbers, each consisting of four bits. It has eight input lines, which present two 4-bit numbers: A3, A2, A1, A0 and B3, B2, B1, B0, and three output lines indicating whether the first number is less than, equal to, or greater than the second number. [13]

The following are conditions under which A will be greater than B for a 4-bit comparator. [13]

1. A3 = 1 and B3 = 0

2. A3 = B3, but A2 = 1 and B2 = 0

3. A3 = B3, A2 = B2, but A1 = 1 and B1 = 0

4. A3 = B3, A2 = B2, A1 = B1, but A0 = 1 and B0 = 0

Next test A < B occurs in the following cases:

1. When A3 = 0 and B3 = 1

2. When A3 = B3, but A2 = 0 and B2 = 1

3. When A3 = B3, A2 = B2, but A1 = 0 and B1 = 1

4. When A3 = B3, A2 = B2, A1 = B1, but A0 = 0 and B0 = 1

The condition A = B is true only when all corresponding bits of the two numbers are equal, that is A3 = B3, A2 = B2, A1 = B1, and A0 = B0.

Using these conditions, the Boolean expressions for each output can be worked out as follows:

- **A > B**:

  - \((A3 \text{ AND } \overline{B3}) \)

- OR \((A3 \text{ XNOR } B3 \text{ AND } A2 \text{ AND } \overline{B2}) \)

  - OR \((A3 \text{ XNOR } B3 \text{ AND } A2 \text{ XNOR } B2 \text{ AND } A1 \text{ AND } \overline{B1}) \)

- OR \((A3 \text{ XNOR } B3 \text{ AND } A2 \text{ XNOR } B2 \text{ AND } A1 \text{ XNOR } B1 \text{ AND } A0 \text{ AND } \overline{B0}) \)

- **A < B**:

  - \((\overline{A3} \text{ AND } B3) \)

  - OR \((A3 \text{ XNOR } B3 \text{ AND } \overline{A2} \text{ AND } B2) \)

- OR \((A3 \text{ XNOR } B3 \text{ AND } A2 \text{ XNOR } B2 \text{ AND } \overline{A1} \text{ AND } B1) \)

  - OR \((A3 \text{ XNOR } B3 \text{ AND } A2 \text{ XNOR } B2 \text{ AND } A1 \text{ XNOR } B1 \text{ AND } \overline{A0} \text{ AND } B0) \)

- **A = B**:

-(A3 XNOR B3) AND (A2 XNOR B2) AND (A1 XNOR B1) AND (A0 XNOR B0)

These Boolean expressions can be used to design a logic circuit which implements the 4-bit comparator. The circuit will correctly tell whether one 4-bit binary number is less than, equal to or greater than another. [13]

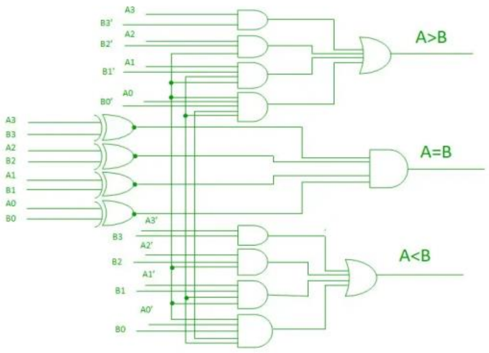


Figure 30 : 4-Bit Magnitude Comparator

Schematic for 4-bit comparator

4-bit comparator is build by using 4 of the 1-bit comparator and 7 AND , 2 OR .

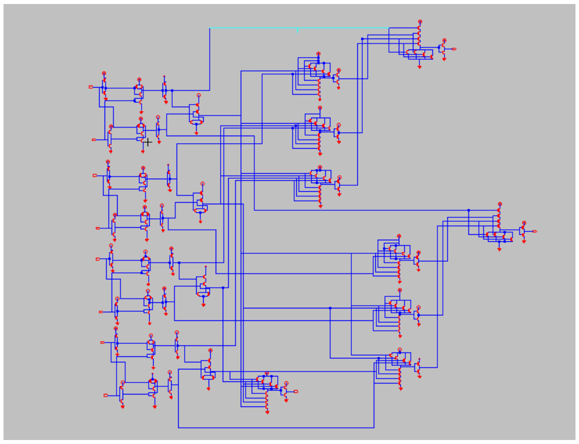


Figure 31 : Schematic for 4-bit comparator

Layout for 4-bit comparator

To do the layout , use the layout of inverter , layout for NAND , layout for 1-bit comparator , then connect 4 of the 1-bit comparator , 1-bit comparator with 7 AND and 7 AND with 2 Or .

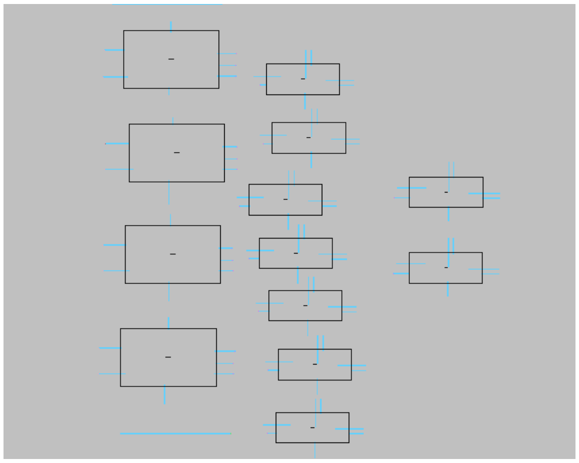


Figure 32 : Layout for 4-bit comparator

16-bit comparator

A 16-bit comparator is designed in behavioral style of modeling and also in structural style of modeling. In behavioral style, a comparator can be designed by knowing only the output characteristics of a comparator. Hence it is very simple to design comparators of any data width without knowing the logic inside .

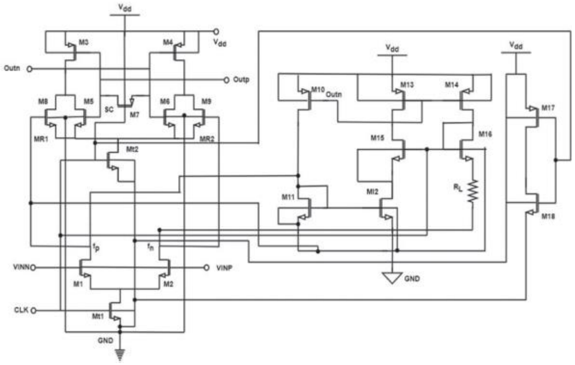


Figure 33 : 16-bit comparator

Schematic for 4-bit comparator

16-bit comparator is build by using 4-bit comparator and 7 AND , 2 OR .

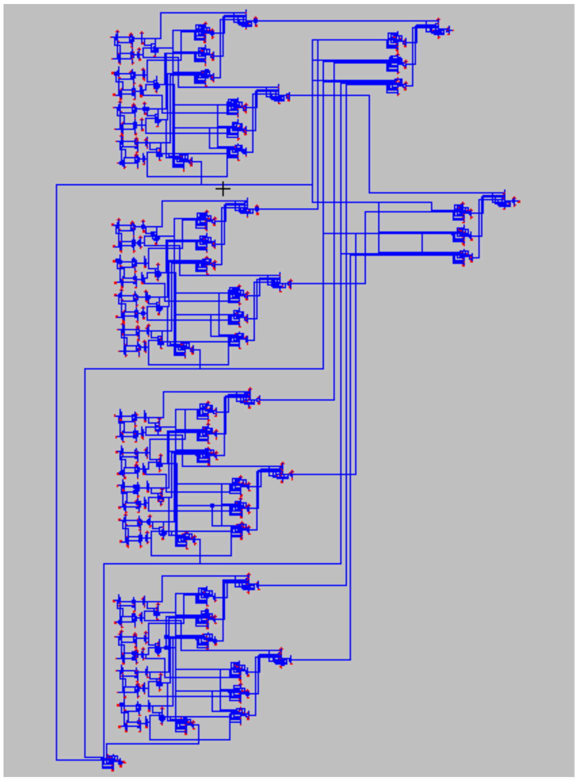


Figure 34 : Schematic for 4-bit comparator

Layout for 16-bit comparator

To do the layout , use the layout of inverter , layout for NAND , layout for 4-bit comparator , then connect 4 of the 4-bit comparator , 1-bit comparator with 7 AND and 7 AND with 2 Or .

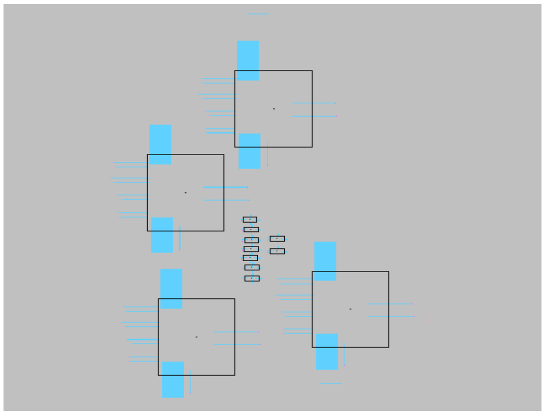


Figure 35 : Layout of 16-bit comparator

Simulations

Inverter :

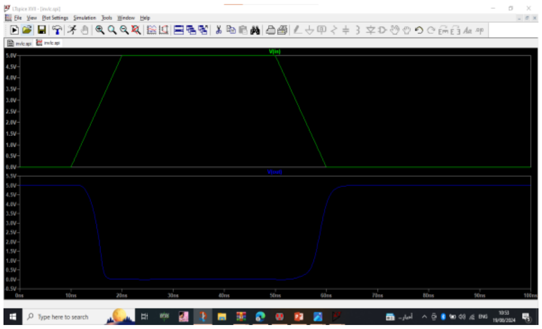


Figure 36 : simulation for Inverter

Nor :

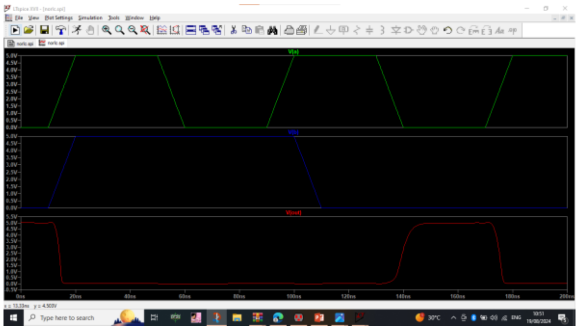


Figure 40 : simulation for NOR

NAND

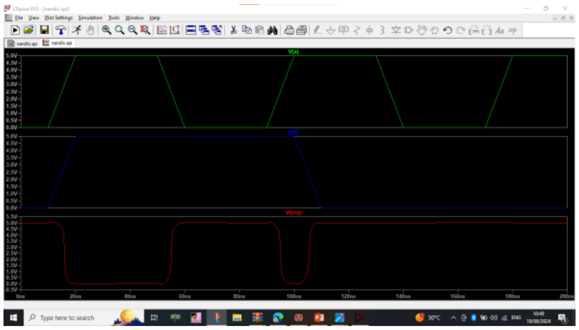


Figure 41 : simulation for NAND

4-bit OR

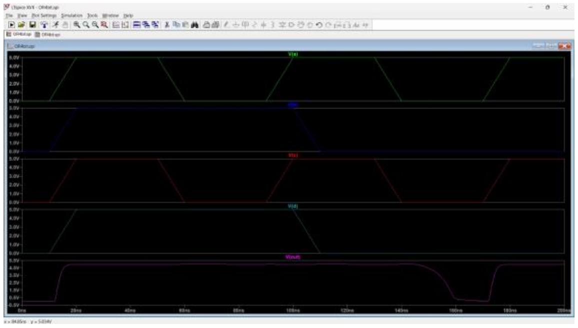


Figure 42 : simulation for 4-bit OR

4-bit AND

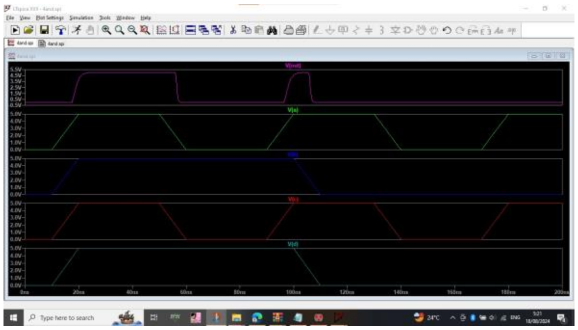


Figure 43 : 4-bit AND

1-bit comparator

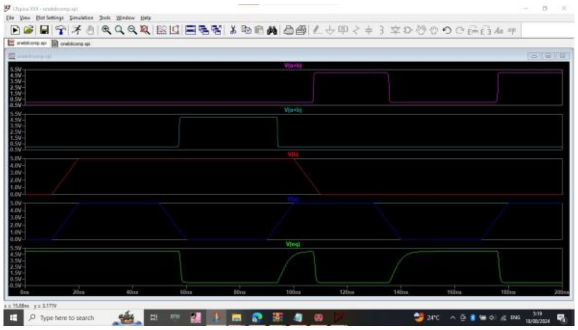


Figure 44 : 1-bit Comparator

4-bit Comparator

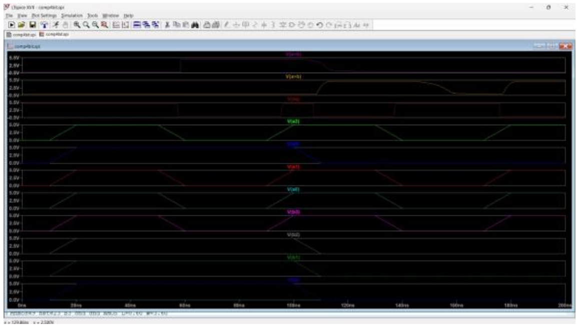
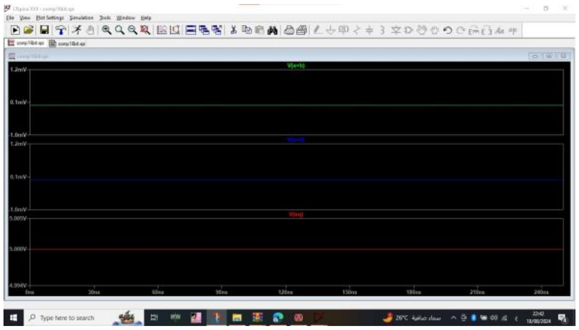


Figure 45 : simulation for 4-bit comparator



7Figure 46 : simulation for 16-bit comparator

Calculations

- **Area**=15105µm×20828.5µm=314,579,842.5µm
- **Delay**= 74.915392-73.582126=1.333266
- **P**=V×I = V * I=V×I.
- **avg_power**=5V×(−2.00862×10^−9 A)=−1.00431×10−8W=−10.0431nW

Note that the area is got from the window of the LTspice. And the Power, the current and the voltage were found using the code. For the delay, it’s got using the cursor.

Codes:

NOT:

```
vdd vdd 0 DC 5

vin in 0 pwl 10n 0 20n 5 50n 5 60n 0

cload out 0 250fF

.measure tran tf trig v(out) val=4.5 fall=1 td=8ns trag v(out) val=0.5 fall=1

.measure tran tr trig v(out) val=0.5 rais=1 td=50ns trag v(out) val=4.5 rais=1

.tran 0 0.1us

.include C:\IC\C5_models.txt
```

NAND & NOR:

```
vdd vdd 0 DC 5

Va A 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5  140n 0 170n 0 180n 5

Vb B 0 pwl 10n 0 20n 5 100n 5 110n 0

cload Y 0 250fF

.measure tran tf trig v(Y) val=4.5 fall=1 td=8ns trag v(Y)=0.5 fall=1

.measure tran tr trig v(Y) val=0.5 rise=1 td=50ns trag v(Y)=4.5 rise=1

.tran 200ns

.include "C:\IC\C5_models.txt"
```

4 bits AND & OR:

```
vdd vdd 0 DC 5

va A 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5

vb B 0 pwl 10n 0 20n 5 50n 5 100n 5 110n 0

vc C 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5

vd D 0 pwl 10n 0 20n 5 50n 5 100n 5 110n 0

cload out 0 250fF

.measure tran tf trig v(out) val=4.5 fall=1 td=4ns trag v(out) val=0.5 fall=1

.measure tran tr trig v(out) val=0.5 rais=1 td=4ns trag v(out) val=4.5 rais=1

.tran 200ns

.include C:\IC\C5_models.txt
```

1 bit Comp:

```
vdd vdd 0 DC 5

va A 0 pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5

vb B 0 pwl 10n 0 20n 5 50n 5 100n 5 110n 0

cload eq 0 250fF

.measure tran tf trig v(eq) val=4.5 fall=1 td=4ns trag v(eq) val=0.5 fall=1

.measure tran tr trig v(eq) val=0.5 rais=1 td=4ns trag v(eq) val=4.5 rais=1

.tran 200ns

.include C:\IC\C5_models.txt
```

16bits Comp:

```
* Power supply

vdd vdd 0 DC 5

* Piecewise Linear sources for 16-bit A

va0 A0 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va1 A1 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va2 A2 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va3 A3 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va4 A4 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va5 A5 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va6 A6 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va7 A7 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va8 A8 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va9 A9 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va10 A10 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va11 A11 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va12 A12 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va13 A13 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va14 A14 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

va15 A15 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

* Piecewise Linear sources for 16-bit B

vb0 B0 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb1 B1 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb2 B2 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb3 B3 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb4 B4 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb5 B5 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb6 B6 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb7 B7 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb8 B8 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb9 B9 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb10 B10 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb11 B11 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)
```

```
vb12 B12 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb13 B13 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb14 B14 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

vb15 B15 0 PWL(0n 0 50n 0 100n 0 150n 0 200n 0)

* Load capacitor

cload x0eqb 0 250fF

* Measurements

.measure tran tf1 TRIG v(x0eqb) val=0.5 rise=1 td=4ns TARG v(x0eqb) val=4.5 rise=1

.measure tran tf2 TRIG v(x0eqb) val=4.5 fall=1 td=4ns TARG v(x0eqb) val=0.5 fall=1

* Transient analysis

.tran 250ns

* Include model file

.include C:\IC\C5_models.txt
```

3 **Challenges:**

A number of issues arose during the layout phase. First, there was the integrity of the signal, which was a big issue since there are so many lines carrying the signal to increase the effects of noise and reflection on performance. Component selection and the correct component type for the design was another. Finally, the nuisances of LTSpice mastering was a challenge, since, although very helpful in features, it often led to errors in simulation with either failure or results that are not correct.

Also critical in the process was ensuring all the connections were properly connected with correct grounding and Vdd connections. Designing and laying out a complicated circuit like a 16-bit comparator proved to be time-consuming. The layout phase was very demanding and required a lot of attention to detail. Further complicating the layout process was the need to make sure wires did not touch each other, thus requiring a better understanding of LTSpice idiosyncrasies to manage these issues..include C:\IC\C5_models.txt

4**References:**

https://www.ansys.com/blog/what-is-an-integrated-circuit [1] 10:19 PM

https://www.ninjaone.com/it-hub/endpoint-management/what-is-an-integrated-circuit/ [2] 10:55

https://tresbizz.com/blog/evolution-of-integrated-circuits/ [3] 11:35

https://www.tutorialspoint.com/history-of-the-integrated-circuit [4] 12:25am

https://www.autodesk.com/products/fusion-360/blog/integrated-circuit-moores-law/ [5] 12:50 PM

https://resources.pcb.cadence.com/blog/2023-ic-design-and-manufacturing-process [6] 1:34 Pm

https://www.nwengineeringllc.com/article/schematic-vs-layout-pcb-geometry-parasitics-and-signal-integrity.php [7] 2:03PM

https://www.build-electronic-circuits.com/not-gate/ [8] 2:45 PM

https://www.build-electronic-circuits.com/nor-gate/ [9] 3:56PM

https://www.build-electronic-circuits.com/nand-gate/ [10] 4:12PM

https://www.build-electronic-circuits.com/or-gate/ [11] 4:44PM

https://www.build-electronic-circuits.com/and-gate/ [12] 4:57PM

https://www.geeksforgeeks.org/magnitude-comparator-in-digital-logic/ [13] 5:13PM

5Reference Paper:

Design of a CMOS Comparator for Low Power
and High Speed. R. S. Gamad & Sumit Kale

Table Of Content

Table Of Figures