1. **INTRODUCTION**
   1. **Introduction to vlsi**

Very large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by the Combining Millions are billions of MOS integrated onto a single chip. VLSI was introduced in the 1970s when MOS integrated circuit (Metal Oxide Semiconductor) chips were developed which were then widely adopted. Their adoption led to More developments which allowed us to develop complex semiconductors and telecommunication technologies.

* 1. **Overview**

General Microelectronics introduced the first commercial MOS integrated circuit in 1964. In the early 1970s, MOS integrated circuit technology allowed the integration of more than 10,000 transistors in a single chip. This allowed the researchers, in VLSI during the 1970s and 1980s, to fabricate tens of thousands of MOS transistors on a single chip (later hundreds of thousands, then millions, and now billions).

The first semiconductor chip was made of two transistors each. Subsequent advances allowed to fabricate more transistors, and consequently, more individual functions or systems were integrated over time. The first integrated circuit was able to hold only a few devices, perhaps as many as ten diodes, transistors, resistors, and capacitors, making it possible to fabricate one or more logic gates on a single device which, retrospectively is known as the Small-Scale Integration (SSI). The improvements in this technique led us to add more devices. With hundreds of logic gates, the integration technique is called Medium-Scale Integration (MSI). Further improvements led us to add more devices which created the Large-Scale Integration (LSI) technique where systems consist of at least a thousand logic gates. At present, the technology has moved far past this mark and today's microprocessors have many millions of gates and billions of individual transistors.

Once, there used to be more work to name and calibrate various levels of large- scale integration. Above VLSI, terms like ultra-large-scale integration (ULSI) were used. But the huge number of gates and transistors available on common devices has

* 1. **What is VLSI?**

VLSI, which stands for Very Large-Scale Integration, is a process used in the design and fabrication of integrated circuits (ICs), which are electronic circuits that are made up of many transistors and other components that are integrated into a single chip. the technology allows for the creation of highly complex and compact ICs that can be used in a wide range of applications, from consumer electronics and computers to communication systems and medical devices.

One of the key advantages of this technology is its ability to pack many transistors and other components onto a single chip, which allows for the creation of highly complex and powerful ICs. This makes it possible to create ICs that can perform a wide range of functions, from simple logical operations to complex algorithms. It also allows for the creation of ICs with high levels of performance, power efficiency, and reliability, which are essential for many applications.

Another advantage of VLSI technology is its cost-effectiveness. Because it allows for the integration of many components onto a single chip, it can reduce the number of discrete components that are needed in a circuit, which can help to reduce the overall cost of the circuit. Additionally, it allows for the creation of ICs with high levels of performance and functionality, which can help to improve the overall performance and efficiency of a system.

1. **History of IC’s**

Integrated circuits (ICs) have a fascinating history that dates to the mid-20th century. Here’s a brief overview:

**Table1.1: History of IC’s**

|  |  |  |
| --- | --- | --- |
| **Stage** | **Description** | **Year** |
| **Early Development** | Researchers like William Shockley, John Bardeen, and Walter Brattain invented the transistor at Bell Labs in 1947, revolutionizing electronics. | 1947 |
| **jack Kilby and the First IC** | In 1958, Jack Kilby of Texas Instruments created the first integrated circuit, integrating multiple components onto a single semiconductor substrate. | 1958 |
| **Robert Noyce and the Monolithic IC** | Around the same time, Robert Noyce developed the monolithic IC at Fairchild Semiconductor, where all components were fabricated on a single silicon chip. | Late 1950s |
| **IC Revolution** | ICs marked a revolution, allowing miniaturization, reduced power consumption, increased reliability, and lower production costs compared to discrete components. | 1958 onwards |
| **Evolution of ICs** | IC technology advanced rapidly, leading to small-scale integration (SSI), medium-scale integration (MSI), large-scale integration (LSI), and very-large-scale integration (VLSI). Today, we have ultra-large-scale integration (ULSI) with billions of transistors on a single chip. | 1960s-present |
| **Types of ICs** | ICs include analog ICs (for amplification, filtering, etc.), digital ICs (for logic operations), mixed-signal ICs (integrating analog and digital functions), and microprocessors (complete CPUs on a chip). | 1960s-present |
| **Applications** | ICs are used in computers, smartphones, IoT devices, automotive electronics, medical equipment, aerospace systems, and more. | 1960s-present |
| **Moore’s Law** | Gordon Moore's observation in 1965 that the number of transistors on a chip double approximately every two years, driving rapid IC technology advancement. | 1965 |

Overall, the history of ICs is a testament to human ingenuity and innovation, showcasing how a small invention can have a monumental impact on technology and society.

1. **Moore’s law:**

Moore's law is the observation that over the history of computing hardware, the number of transistors on integrated circuits doubles approximately every two years. A black background with a thin line

Description automatically generated

**Figure 1.1** Microprocessor transistor count

The period often quoted as "18 months" is due to Intel executive David House, who predicted that period for a doubling in chip performance (being a combination of the effect of more transistors and their being faster).[1] The law is named after Intel co-founder Gordon E. Moore, who described the trend in his 1965 paper. [2][3][4] The paper noted that the number of components in integrated circuits had doubled every year from the invention of the integrated circuit in 1958 until 1965 and predicted that the trend would continue "for at least ten years".[5] His prediction has proven to be uncannily accurate, in part because the law is now used in the semiconductor industry to guide long-term planning and to set targets for research and development. [6]

The capabilities of many digital electronic devices are strongly linked to Moore's law: processing speed, memory capacity, sensors and even the number and size of pixels in digital cameras. [7] All of these are improving at (roughly) exponential rates as well (see other formulations and similar laws).

This exponential improvement has dramatically enhanced the impact of digital electronics in nearly every segment of the world economy.[8] Moore's law describes a driving force of technological and social change in the late 20th and early 21st centuries.[9][10] This trend has continued for more than half a century. Sources in 2005 expected it to continue until at least 2015 or 2020.[11][12] However, the 2010 update to the International Technology Roadmap for Semiconductors has growth slowing at the end of 2013,[13] after which time transistor counts and densities are to double only every three years.



# **Figure 1.2** Osborne Executive

1. **Advantages of IC’s**

VLSI technology provides high integration density, enabling complex functions in compact spaces, reducing size and weight in portable devices. It optimizes power consumption for energy-efficient operation, enhances performance with high-speed processing capabilities, and offers cost-effectiveness through mass production. Customization options cater to specific design requirements, ensuring reliability and scalability for evolving technological demands.

1. **Background:**

In digital circuit design, decoders are indispensable for transforming coded inputs into corresponding outputs, serving as foundational components in microprocessors, memory units, and communication devices. Traditional decoder designs utilizing CMOS logic gates often exhibit substantial power consumption, particularly in high-speed applications, prompting exploration into alternative methodologies for reduced power usage without compromising performance. The Gate Diffusion Input (GDI) technique has emerged as a notable solution, optimizing logic gates to achieve lower power consumption and reduced area overhead compared to conventional approaches. GDI gates employ a single NMOS transistor for input and multiple PMOS transistors for diffusion inputs, facilitating efficient implementation of complex logic functions while minimizing transistor count and optimizing gate structures. This research delved into existing decoder designs, analysed their power consumption profiles, and explored GDI methodology's principles, revealing its potential to significantly lower power consumption while meeting performance criteria.

1. **LITERATURE SURVEY**
2. **Literature Survey**

**V. G. Oklobdzija and B. Duchene (1995)** presented their work on "Pass-transistor dual value logic for low-power CMOS" at the International Symposium on VLSI Technology in 1995. Their research focuses on utilizing pass-transistor logic to achieve low-power consumption in CMOS circuits. The paper likely discusses the advantages of this approach in terms of energy efficiency and circuit performance. Pass-transistor logic is known for its simplicity and reduced transistor count, making it attractive for low-power designs. This work contributes to the ongoing efforts in developing energy-efficient circuitry, a critical aspect in modern semiconductor technology. Accessing the paper would provide more detailed insights into their methodologies and experimental results.

**BNM Reddy, HN Sheshagiri, et al. (2014)** the implementation of a low power 8-bit multiplier using Gate Diffusion Input (GDI) logic, as discussed in the paper by BNM Reddy, HN Sheshagiri, et al. (2014), focuses on leveraging GDI techniques to achieve energy-efficient and compact logic design. GDI methodology is known for its reduced transistor count and improved energy efficiency compared to traditional CMOS logic. The key highlight of this implementation is the utilization of Booth encoding, a technique that optimizes multiplication operations by reducing the number of partial product terms generated during the computation process. By efficiently encoding input operands, the multiplier can perform computations with fewer logic gates, leading to lower power consumption and enhanced performance.

Furthermore, the design incorporates a Wallace tree architecture, which is renowned for its ability to reduce the number of partial products and the overall critical path delay in multipliers. This architectural choice not only contributes to power reduction but also improves the speed of the multiplier circuit.

The synergistic combination of GDI logic, Booth encoding, and Wallace tree architecture allows the 8-bit multiplier to achieve significant reductions in power consumption and area utilization while maintaining low complexity in logic design. These techniques demonstrate effective strategies for implementing energy-efficient digital circuits, particularly in arithmetic units such as multipliers, within low-power design contexts

**N. Sowjith, K. Sandeep, S., Sumanth, M., and S. Agrawal in (2016)** the paper authors introduce a novel low-power VLSI architecture specifically designed for a combined

FMO/Manchester encoder. This architecture emphasizes reusability and efficiency, addressing key challenges in video coding and communication systems. Presented at the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) 2016, their work likely integrates modular components for easy adaptation into various applications. The focus on reducing power consumption while handling FMO and Manchester encoding suggests innovative optimization strategies. Further exploration of the paper would unveil detailed methodologies, experimental results, and contributions to the field of VLSI design for multimedia systems.

**Bissa P.R. and Pande K.S. (2018)** presented their research on "All Digital Phase Locked Loop for Low Frequency Applications" at the International Conference on Advances in Computing Communications and Informatics (ICACCI) in 2018. Their focus was on designing a digital phase-locked loop (PLL) suitable for low-frequency applications. The paper likely discusses the challenges and solutions specific to low-frequency PLL design, which often requires different approaches compared to higher-frequency counterparts. Their work contributes to advancing PLL technology for various applications, including wireless communication systems, sensors, and low-power devices. Accessing the paper would provide a deeper understanding of their PLL design methodology, performance evaluation, and contributions to the field of digital signal processing.

**D. Balobas and N. Konofaos (2017)** published their paper on "Design of Low Power, High Performance 2-4 and 4-16 Mixed Logic Line Decoders" in the IEEE Transactions on Circuits and Systems - II: Express Briefs in February 2017. Their focus was on achieving a balance between low power consumption and high performance in line decoders, specifically targeting 2-4 and 4-16 mixed logic configurations. Line decoders are crucial components in digital systems, and optimizing power efficiency is vital for modern electronics.

Their work likely involves innovative design techniques aimed at reducing power consumption while maintaining or improving performance, possibly leveraging mixed logic configurations. These configurations often combine different logic families to capitalize on their strengths, enhancing overall circuit characteristics.

**Mohan Shoba and Rangaswamy Nakkeeran's (2016)** paper on "GDI-based full adders for energy-efficient arithmetic applications" was published in the International Journal on Engineering Science and Technology in 2016. Their research focuses on designing full adders using **Gate Diffusion Input (GDI)** logic to enhance energy efficiency in arithmetic operations.

Full adders are fundamental components in digital circuits, particularly in arithmetic units like processors and calculators. The use of GDI logic in their design indicates a novel approach to reducing power consumption while maintaining performance in arithmetic applications.

The paper likely discusses the design principles, circuit implementations, and performance evaluations of GDI-based full adders compared to traditional designs. GDI logic is known for its simplicity and potential for energy savings, making it suitable for energy-efficient digital systems. Their work contributes to advancing energy-efficient design methodologies in digital circuits, which is crucial for modern electronic devices aiming for reduced power consumption and longer battery life.

**Anjaneyulu Guddety, S. Sreenivasulu, and S. Mohan Das(2019)** from the Department of Electronics and Communication Engineering at SVR Engineering College in Nandyal focuses on the design of low-power mixed-logic line decoders using DSCH. Their work likely includes a comprehensive survey of various mixed-logic line decoder architectures and techniques aimed at reducing power consumption in digital circuits. This survey would delve into methodologies such as gate diffusion input (GDI), clock gating, power gating, and voltage scaling, among others. It would explore how these techniques can be applied to mixed-logic line decoders to achieve lower power dissipation while maintaining or enhancing performance. Additionally, the survey may cover challenges specific to mixed-logic line decoders, such as signal integrity issues, area constraints, and trade-offs between power efficiency and speed. It may also discuss the impact of these design choices on circuit reliability and manufacturability.

**O.P.P.Vardhanam, T.Sahithi, B. Sai Ram, M. Srikanth Reddy, G.V.Vinod (2020)** In designing low-power, high-performance 2-4 and 4-16 mixed logic line decoders, several key considerations come into play. First and foremost is the selection of appropriate logic gates and architectures that can efficiently handle the decoding process while minimizing power consumption. Implementing techniques like static CMOS logic or dynamic logic can be beneficial in achieving this balance. It looks like you've provided a detailed abstract and introduction to a research paper on the design of low-power, high-performance 2-4 and 4-16 mixed logic line decoders. The paper discusses various design techniques, including transmission gate logic, pass transistor dual-value logic, and static complementary metal oxide semiconductors (CMOS). It also introduces novel topologies for 2-4 decoders with a focus on minimizing transistor count and energy dissipation, as well as high-power optimization.

Mixed logic designs are prioritised in logic design approaches that will offer a streamlined framework for the analysis of digital circuits. A mixed logic implementation also offers a clear picture of the actions within a circuit. presented mixed logic designs here, among others. Examples include pass transistor dual value logic (DVL), static CMOS, and transmission gate logic (TGL). When using mixed logic, a 2:4-line decoder can be built with merely 14 transistors (14T), compared to the 20 transistors required when using CMOS technology. To introduce the mixed logic technique, a 2:1 MUX was made using the 2:4-line decoder of the mixed logic architecture. This innovative approach enhances operating speed and uses less power than conventional logic architecture since it employs fewer transistors.

1. **Overview of Literature**

When comparing various low power decoder designs, each approach brings distinct advantages and considerations. CMOS-based decoders, while standard and widely used, offer moderate power consumption and area utilization, suitable for general-purpose applications. DVL-based designs stand out with their compactness and high energy efficiency, making them ideal for low power-focused applications where space and power constraints are critical. On the other hand, TGL-based decoders excel in high-speed performance and energy efficiency, catering to applications demanding rapid data processing with minimized power consumption.

Robustness and reliability are generally high across these designs, with DVL and TGL designs showcasing superior performance. However, the complexity and design effort may vary, with CMOS designs being relatively straightforward compared to DVL and TGL implementations, which require more intricate circuitry. Cost considerations Favor DVL and TGL designs, known for their cost-effectiveness due to reduced power requirements.

Flexibility and adaptability also vary, with CMOS designs offering standard flexibility

while DVL and TGL designs provide moderate to high flexibility, allowing customization for specific power and performance trade-offs. Overall, the suitability of each design depends on the application requirements, with CMOS designs serving as versatile options, DVL designs excelling in low power scenarios, TGL designs prioritizing high-speed performance, and other designs offering specialized solutions tailored to specific needs.

**Table 2.1: Overview of Existing Literature**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **criteria** | **CMOS-Based Decoder** | **DVL-Based Decoder** | **TGL-Based Decoder** | **Other Designs** |
| Power Consumption | Moderate | Low | Moderate | Varies |
| Area Utilization | Standard | Compact | Compact | Varies |
| Speed and Delay | Moderate | High | High | Varies |
| Energy Efficiency | Standard | High | High | Varies |
| Robustness and Reliability | Standard | High | High | Varies |
| Technology Compatibility | Standard | Compatible | Compatible | Varies |
| Complexity and Design Effort | Moderate | Moderate | Moderate | Varies |
| Cost Considerations | Standard | Cost-Effective | Cost-Effective | Varies |
| Flexibility and Adaptability | Limited | Moderate | Moderate | Varies |
| Overall Suitability | General Purpose | Low Power Focus | High Performance | Varies |

1. **Decoders**
2. **Overview of Decoders**

The combinational circuit that changes the binary information into 2N output lines is known as **Decoders.** The binary information is passed in the form of N input lines. The output lines define the 2N-bit code for the binary information. In simple words, the **Decoder** performs the reverse operation of the **Encoder**. At a time, only one input line is activated for simplicity. The produced 2N-bit output code is equivalent to the binary information.



**Figure 3.1.** N to 2N Decoder

## **Types of Decoders**

## Decoders come in various types, each suited to different purposes:

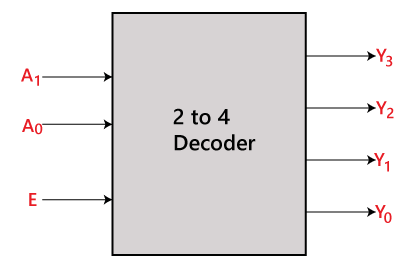
1. **Binary Decoders**: As the name suggests, these decoders take a binary input and decode it into a corresponding output.
2. **BCD to Decimal Decoders**: These are used to decode **binary-coded decimal (BCD)** numbers into their corresponding decimal numbers.
3. **7-segment Display Decoder**: Commonly found in digital clocks and calculators, this type of decoder converts binary numbers into specific patterns that light up the correct segments of the display.
4. **How Decoders Work**

At their core, decoders are complex logic circuits designed using logic gates. The process involves:

1. **Input**: The decoder receives a binary code as input.
2. **Processing**: The decoder processes the input using a combination of logic gates.
3. **Output**: Based on the input, one of the output lines is selected and activated.
4. **Binary Decoders**

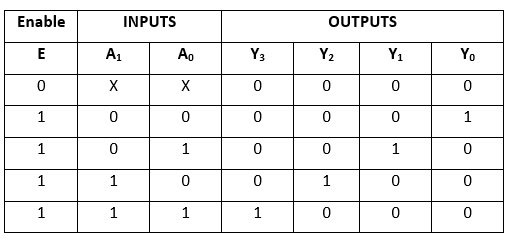
There are various types of decoders which are as follows:

1. **2–4-line Decoder**

 In the 2-to-4-line decoder, there is a total of three inputs, i.e., A0, and A1 and E and four outputs, i.e., Y0, Y1, Y2, and Y3. For each combination of inputs, when the enable 'E' is set to 1, one of these four outputs will be 1. The block diagram and the truth table of the 2-to-4-line decoder are given below.

**Figure 3.2** 2-4-line Decoder

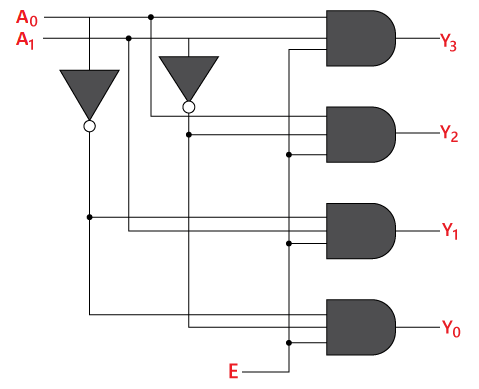
**Table 3.1 Truth Table of 2–4-line decoder:**



The logical expression of the term Y0, Y0, Y2, and Y3 is as follows:

Y3=E.A1.A0  
Y2=E.A1.A0'  
Y1=E.A1'.A0  
Y0=E.A1'.A0'

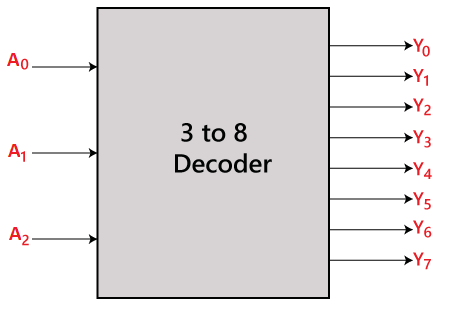
Logical circuit of the above expressions is given below:



**Figure 3.3** 2-4-Line Decoder circuit

## **3-to-8-line decoder:**

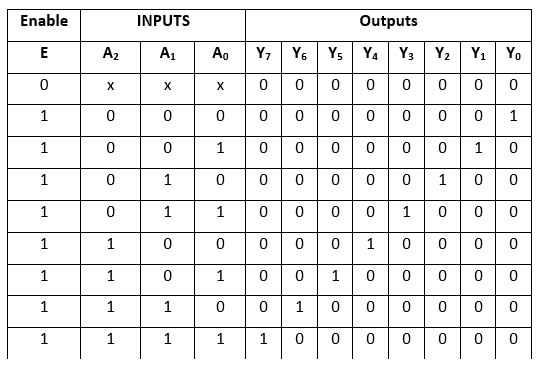
The 3-to-8-line decoder is also known as **Binary to Octal Decoder**. In a 3-to-8-line decoder, there is a total of eight outputs, i.e., Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 and three outputs, i.e., A0, A1, and A2. This circuit has an enable input 'E'.



**Figure 3.4** 3–8-line decoder

Just like 2-to-4-line decoder, when enable 'E' is set to 1, one of these four outputs will be 1. The block diagram and the truth table of the 3-to-8-line encoder are given below.

### **Table 3.2: Truth Table of 3–8-line decoder**

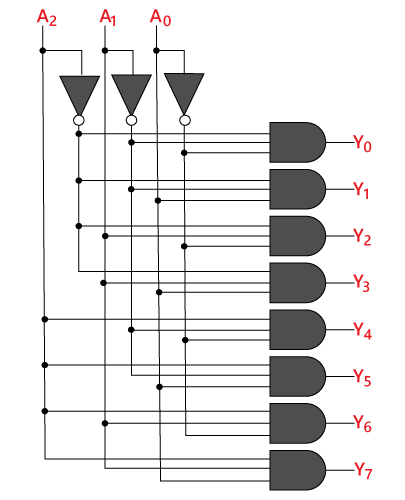


The logical expression of the term Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 is as follows:

Y0=A0'.A1'.A2'

Y1=A0.A1'.A2'  
Y2=A0'.A1.A2'  
Y3=A0.A1.A2'  
Y4=A0'.A1'.A2  
Y5=A0.A1'.A2  
Y6=A0'.A1.A2  
Y7=A0.A1.A2

Logical circuit of the above expressions is given below:



**Figure 3.5** 3-8-line decoder

## **4-to-16-line Decoder**

In this section, let us implement **4 to 16 decoders using 3 to 8 decoders**. We know that 3 to 8 Decoder has three inputs A2, A1 & A0 and eight outputs, Y7 to Y0. Whereas, 4 to 16 Decoder has four inputs A3, A2, A1 & A0 and sixteen outputs, Y15 to Y0.

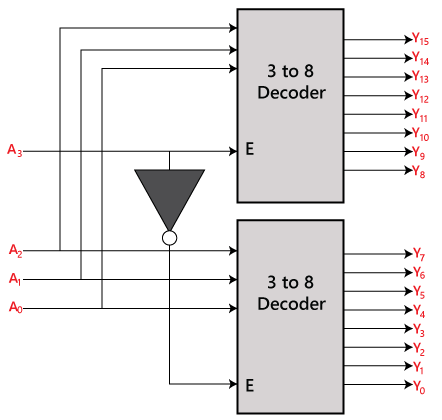
We know the following formula for finding the number of lower order decoders required.

Requirednumberoflowerorderdecoders=m2m1𝑅𝑒𝑞𝑢𝑖𝑟𝑒𝑑𝑛𝑢𝑚𝑏𝑒𝑟𝑜𝑓𝑙𝑜𝑤𝑒𝑟𝑜𝑟𝑑𝑒𝑟𝑑𝑒𝑐𝑜𝑑𝑒𝑟𝑠=𝑚2𝑚1

Substitute, m1𝑚1 = 8 and m2𝑚2 = 16 in the above formula.

Requirednumberof3to8decoders=168=2

Therefore, we require two 3 to 8 decoders for implementing one 4 to 16 decoders. The **block diagram** of 4 to 16 decoder using 3 to 8 decoders is shown in the following figure.



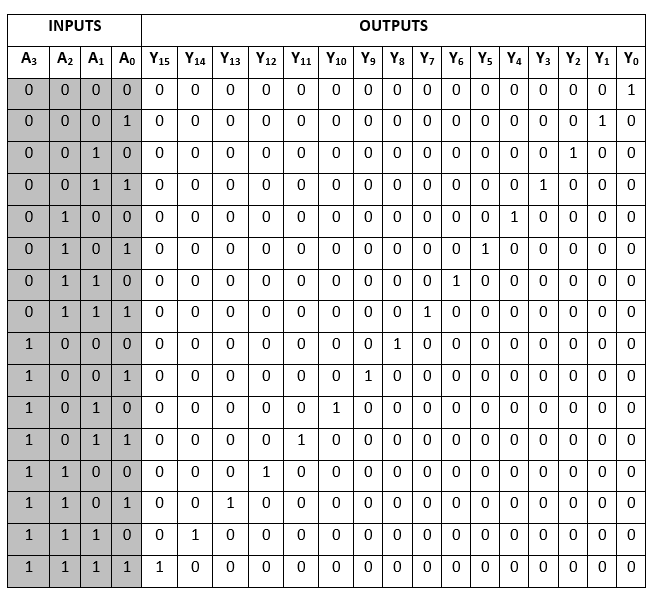
**Figure 3.6** 4–16-line decoder

The 3-to-16-line decoder can be constructed using either 2 to 4 decoder or 3 to 8 decoders. There is the following formula used to find the required number of lower-order decoders. Required number of lower order decoders=m2/m1

m1 = 8  
m2 = 16

Required number of 3 to 8 decoders=Decoder=2

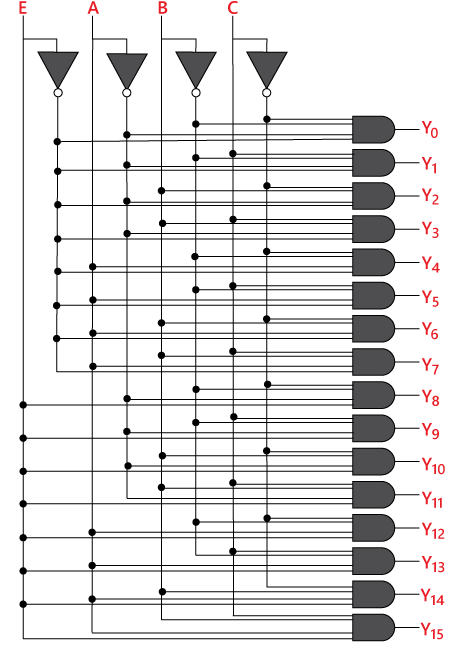
### **Table 3.3 Truth Table of 4–16-line decoder**



The logical expression of the term A0, A1, A2,…, A15 are as follows:

Y0=A0'.A1'.A2'.A3'  
Y1=A0'.A1'.A2'.A3  
Y2=A0'.A1'.A2.A3'  
Y3=A0'.A1'.A2.A3  
Y4=A0'.A1.A2'.A3'  
Y5=A0'.A1.A2'.A3  
Y6=A0'.A1.A2.A3'  
Y7=A0'.A1.A2.A3  
Y8=A0.A1'.A2'.A3'  
Y9=A0.A1'.A2'.A3  
Y10=A0.A1'.A2.A3'  
Y11=A0.A1'.A2.A3  
Y12=A0.A1.A2'.A3'  
Y13=A0.A1.A2'.A3  
Y14=A0.A1.A2.A3'  
Y15=A0.A1.A2'.A3

Logical circuit of the above expressions is given below:



**Figure 3.7** 4-16-line decoder

## **Practical Applications of Decoders**

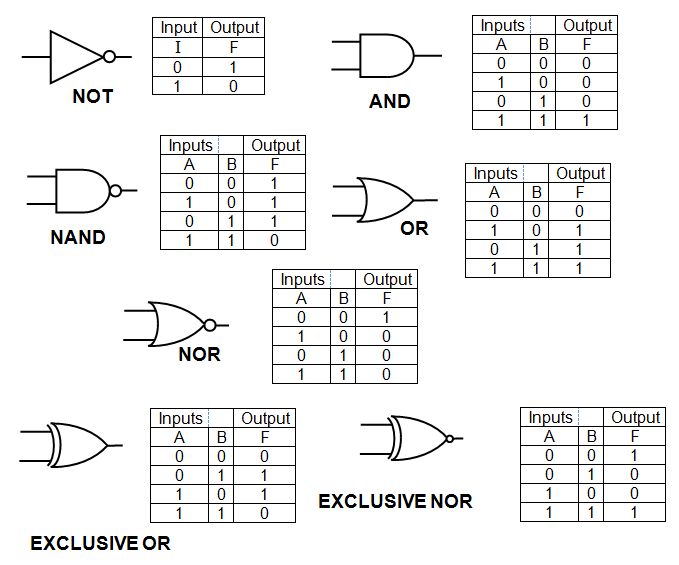
Decoders find applications in real-world scenarios across various domains:

1. **Data Demultiplexing**: Decoders are used in data demultiplexing, where a single input signal is divided into several separate output signals.
2. **Memory Address Decoding**: In computer systems, decoders identify specific memory locations for data retrieval or storage.
3. **Memory Devices**: Memory devices like RAM and ROM utilize decoders to select specific memory cells.
4. **Instruction Decoding in CPUs**: In CPUs, decoders translate encoded instructions into control signals that perform desired operations on the hardware.

**4. Existing System**

# **Logic Gates**

**Logic gates**are the fundamental components of all digital circuits and systems. In digital electronics, there are seven main types of logic gates used to perform various logical operations. A logic gate is basically an electronic circuit designed by using components like diodes, transistors, resistors, capacitors, etc., and capable of performing logical operations. In this article, we will study the definition, truth table, and other related concepts of logic gates. So, let’s start with the basic introduction of logic gates.



**Figure 4.1** logic gates

## **What is a Logic Gate?**

A **logic gate**is an electronic circuit designed by using electronic components like diodes, transistors, resistors, and more. As the name implies, a logic gate is designed to perform logical operations in digital systems like computers, communication systems, etc.

Therefore, we can say that the building blocks of a digital circuit are logic gates, which execute numerous logical operations that are required by any digital circuit. A logic gate can take two or more inputs but only produce one output.

The output of a logic gate depends on the combination of inputs and the logical operation that the logic gate performs.

Logic gates use Boolean algebra to execute logical processes. Logic gates are found in nearly every digital gadget we use on a regular basis. Logic gates are used in the architecture of our telephones, laptops, tablets, and memory devices.

## **Types of Logic Gates**

A logic gate is a digital gate that allows data to be manipulated. Logic gates, use logic to determine whether to pass a signal. Logic gates, on the other hand, govern the flow of information based on a set of rules.

The logic gates can be classified into the following major types:

1. **Basic Logic Gates**– There are three basic logic gates:

AND Gate

OR Gate

NOT Gate

1. **Universal Logic Gates**– In digital electronics, the following two logic gates are considered as universal logic gates:

NOR Gate

NAND Gate

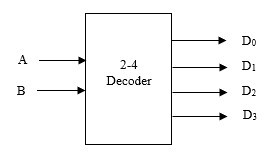
1. **Derived Logic Gates**– The following two are the derived logic gates used in digital systems:

XOR Gate

XNOR Gate

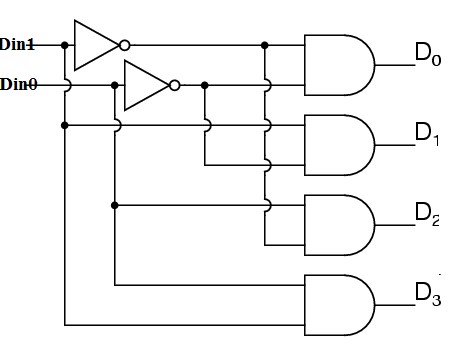
* 1. **Existing system**

The fundamental digital module is the decoder which decodes the coded input which is generally used in all types of memory devices.



**Figure 4.2.** Block diagram of 2-4 Decoder.

Most common decoder circuit is an *n* input to 2*n* output binary decoder. In the conventional design the CMOS technology is used to design the logic of any application.



**Figure 4.3.** 2-4 decoder Gate level diagram.

**Table 4.1: truth table of the 2-4 decoder**

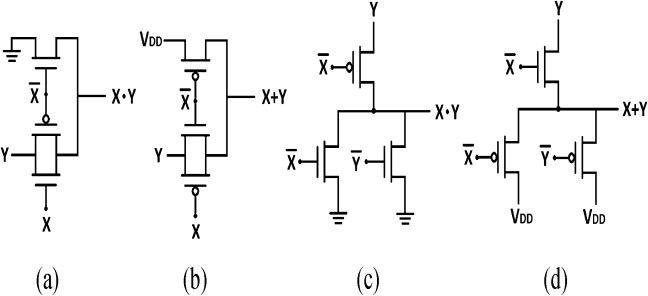
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **D0** | **D1** | **D2** | **D3** |
| **L** | **L** | **H** | **L** | **L** | **L** |
| **L** | **H** | **L** | **H** | **L** | **L** |
| **H** | **L** | **L** | **L** | **H** | **L** |
| **H** | **H** | **L** | **L** | **L** | **H** |

**TABLE 2: truth table of the inverting 2-4 decoder**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **I0** | **I1** | **I2** | **I3** |
| **L** | **L** | **L** | **H** | **H** | **H** |
| **L** | **H** | **H** | **L** | **H** | **H** |
| **H** | **L** | **H** | **H** | **L** | **H** |
| **H** | **H** | **H** | **H** | **H** | **L** |

## **Pass Transistor Logic**

The Pass transistor logic uses more NMOS and PMOS transistors which are used in parallel or series to make the desired logic. In this PTL logic the transistors act as switches to propagate the VDD to the output port so that the desired logic is achieved. This method reduces the total number of the active transistors, but it has a drawback of the voltage variation is present at the output of each stage it propagating to the final output stage. So, this PTL design methods are exclusively for some specific applications rather than the regular designs.



**Figure 4.4.** Logic AND as well as OR Gates Circuits(3transistor). (a) Transmission Gate Logic AND gate. (b) Transmission Gate Logic OR gate. (c) Dual Value Logic AND gate. (d) Dual Value Logic OR gate.

The above fig.3 (a) shows a transmission gate logic AND gate which operates when both X and Y are logic High [H,H] then only the output gives the logic high value otherwise the output is logic zero value. Similarly, fig 3. (b) shows the OR gate which operate when either

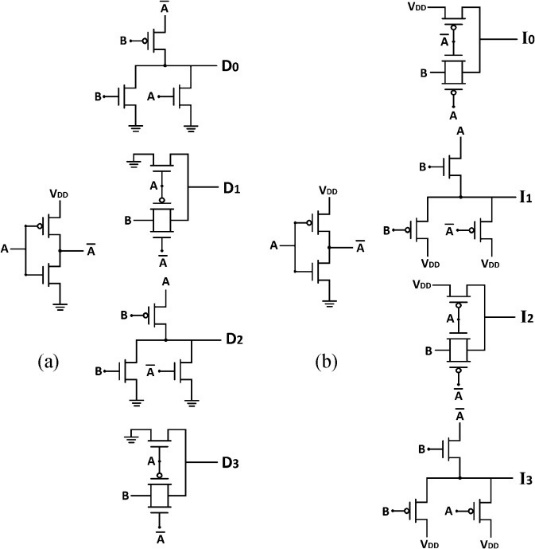
X or Y are at logic high then the output logic high otherwise the output is logic zero. Both the cases the circuits require only one transmission gate and one NMOS transistor. Fig 3.(c),(d) shows dual value logics of the AND gate and OR gate which are required only 3-transistors when compared to conventional CMOS 6-transistors design.

1. **Transmission Gate Logic**

The transmission gate logic has the advantage of maintaining the good output voltage value over the pass transistor logic that is why transmission gates are widely used. In the proposed design uses the transmission gate logic to realize the 14-transistor low power and low power inverted design, 15-transistor high power and high-power inverted decoder designs to achieve the low power, high performance, and less die area.

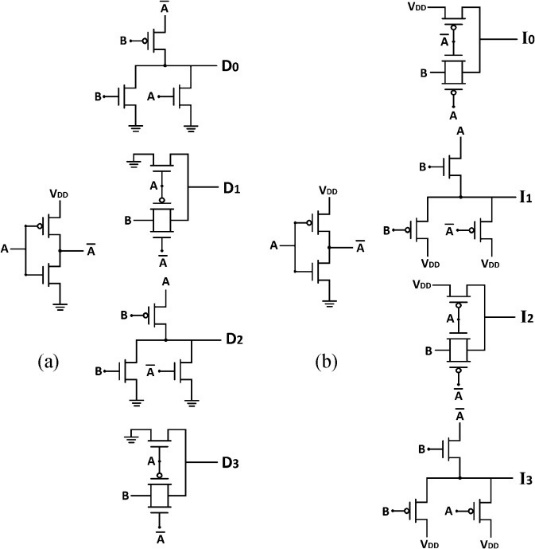
1. **Mixed Logic Design**

In this mixed logic design technique combines the CMOS, PTL, and DVL logics to achieve the low operating power, low power dissipation, minimal die area, and high performance. In the proposed design uses mostly transmission gate logic and CMOS technologies to achieve the fundamental digital logic gates and 14-transistors low power and low power inverted designs,15-transistors high power and high-power inverted designs to get the 2 to 4 decoders.



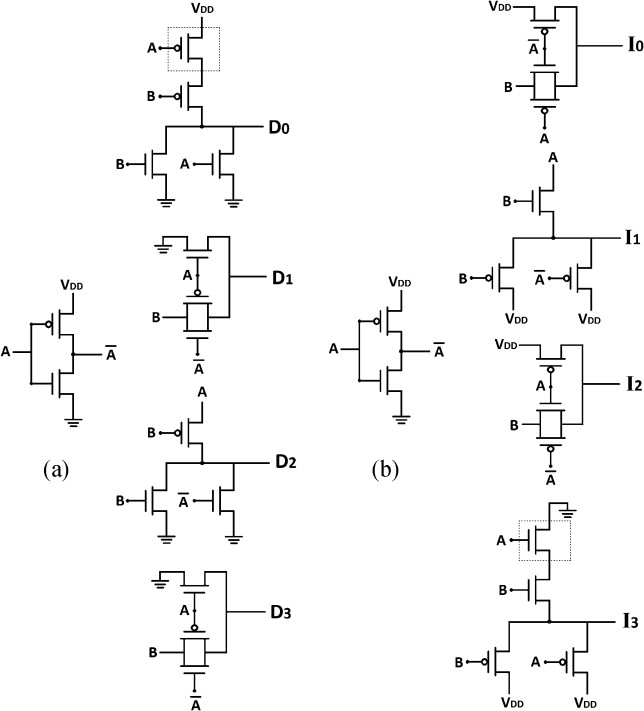
**Figure 4.5.** 2-to-4-line decoders (14-transistors). (a) Low Power 2 to 4

The above fig.4 (a) shows the fourteen-transistor mixed logic design to realize the 2-to-4 decoder which needs transmission gate logic (TGL gate for D1 and D3), Dual value logic (DVL for D0 and D2) and one CMOS inverter. Fig 4.(b) shows fourteen transistor mixed logic design to realize the 2-to-4 decoder which needs transmission gate logic(TGL gate for D0 and D2), Dual value logic (DVL for D1 and D3) and CMOS inverter. Similarly, below fig.5 (a) shows the fourteen-transistor mixed logic design to realize the 2-to-4 decoder which needs transmission gate logic (TGL gate for D1 and D3), Dual value logic (DVL for D2) and one CMOS NOR gate and inverter.



**Figure 4.6.** (b) Low power Inverted 2 to 4.

Fig 4.(b) shows fourteen transistor mixed logic design to realize the 2-to-4 decoder which needs transmission gate logic(TGL gate for D0 and D2), Dual value logic (DVL for D1 ) and CMOS NOR gate and inverter.



**Figure 4.7.** 2-to-4-line decoders (15-transistors). (a) High Power 2 to 4(b) High power Inverted 2 to 4.

**5. Proposed System**

1. **Proposed Methodology**
2. **Motivation for GDI:**

As VLSI technology advances, designers seek ways to improve performance while minimizing power consumption. Traditional CMOS (Complementary Metal-Oxide-Semiconductor) and pass-transistor logic (PTL) have their limitations. GDI aims to address these limitations by reducing power consumption, propagation delay, and circuit area while maintaining low complexity. GDI stands for Gate Diffusion Input. It's a technique used in digital VLSI (Very Large-Scale Integration) circuit design to implement logic gates using a minimum number of transistors, thereby reducing the area and power consumption of the circuit.

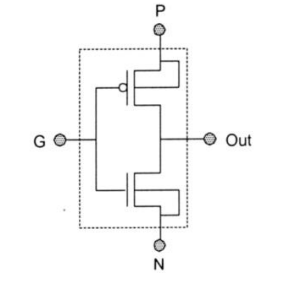
1. **Analysis of Gdi Technique**

The GDI method which is first proposed by A. Morgenshtein, A. Fish, and I. A. Wagner in 2001 , is based on the use of a simple cell as shown in figure.4. At first glance, the basic cell

reminds the standard CMOS inverter, but there are some important differences:

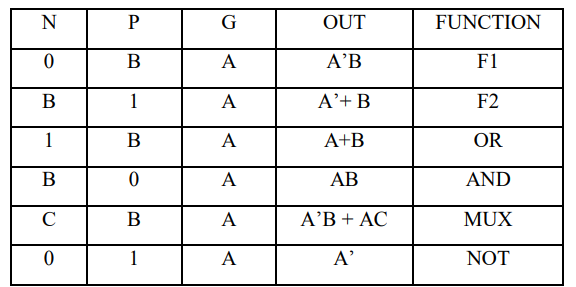
1. The GDI cell contains three inputs: G (common gate input of NMOS and PMOS), P (input to the source/drain of PMOS), and N (input to the source/drain of NMOS).
2. Bulks of both NMOS and PMOS are connected to N or P (respectively), so it can be arbitrarily biased at contrast with a CMOS inverter. It must be remarked that not all of the functions are possible in standard p-well CMOS process but can be successfully implemented in twin-well CMOS or silicon on insulator (SOI) technologies.

The Basic GDI cell is shown in fig:



**Figure 5.1.**GDI basic cell

**Table 5.1: Various Logic Functions of GDI Cell**

****

**N**: Represents an input value (usually binary) for the NMOS (n-channel Metal-Oxide-Semiconductor) transistor.

**P**: Represents an input value (usually binary) for the PMOS (p-channel Metal-Oxide-Semiconductor) transistor.

**B**: Represents a constant value (usually 0 or 1) used in the logic expressions.

**G**: Represents a common gate input for both NMOS and PMOS transistors in the GDI (Gate Diffusion Input) cell.

**A**: Represents another input value (usually binary) used in the logic expressions.

1. **F1 (A’B)**:
   * In the first row, we have:
     + **N=0**, **P=B**, and **G=A**.
     + The output is **A’B**.
   * The logic operation performed here is an **AND operation** between **A** and its complement **B**.
   * So, when **N=0**, the NMOS transistor is off (since it’s connected to ground), and the PMOS transistor is on (since it’s connected to VDD). This results in an output of **A’B**.
2. **F2 (A’+B)**:
   * In the second row, we have:
     + **N=B**, **P=1**, and **G=A**.
     + The output is **A’+B**.
   * The logic operation performed here is an **OR operation** between **A’** (complement of **A**) and **B**.
   * When **N=B**, the NMOS transistor is on (since it’s connected to VDD), and the PMOS transistor is off (since it’s connected to ground). This results in an output of **A’+B**.
3. **OR Gate (A+B)**:
   * When **N=1**, **P=B**, and **G=A**, the output is **A+B**.
   * This corresponds to the third row in the truth table.
   * The logic operation performed here is a straightforward OR operation between **A** and **B**.
4. **AND Gate (AB)**:
   * When **N=B**, **P=0**, and **G=A**, the output is **AB**.
   * This corresponds to the fourth row in the truth table.
   * The logic operation performed here is an AND operation between **A** and **B**.
5. **MUX Gate:**

Function: This gate is a multiplexer (MUX) with two data inputs A and B, and a control input C. When C is low (0), the output (OUT) is the logical AND between A' and B. When C is high (1), the output is the logical AND between A and C.

**Truth Table:**

When C=0 and B=1, the output is A' AND B, representing one of the possible inputs of the multiplexer.

When C=1 and A=1, the output is A AND C, representing the other possible input of the multiplexer.

1. **NOT Gate:**

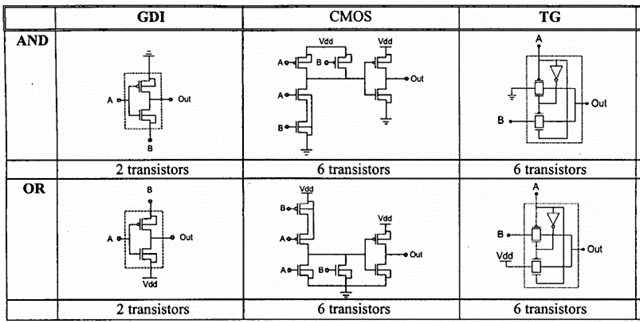
Function: This gate performs the logical NOT operation on input A. The output (OUT) is the complement of input A.

**Truth Table:**

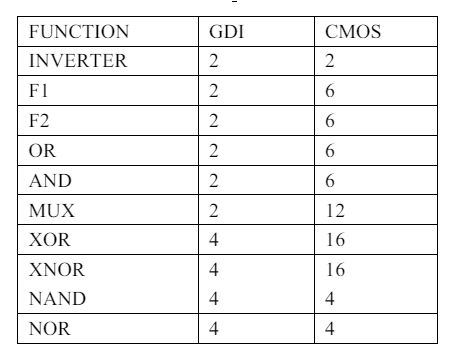
When A=0, the output is 1 (A').

When A=1, the output is 0 (A').

**Table 5.2 Comparisons of AND Gate, OR Gates transistors in different logics**

****

**Table 5.3 :** **comparison of transistor counts of gdi and static cmos**

****

It can be seen that large number of functions can be implemented using the basic GDI cell. MUX design is the most complex design that can be implemented with GDI, which requires only 2 transistors, which requires 8-12 transistors with the traditional CMOS or PTL design. Many functions can be implemented efficiently by GDI by means of transistor count. Table 5.3 shows the comparison between GDI and the static CMOS design in terms of transistors count. It can be seen from table 5.3 that using GDI technique AND, OR, Function1, Function2, XOR, XNOR can be implemented more efficiently. However, to implement NAND, NOR it requires 4 transistors as that in Static CMOS design. NAND and NOR the universal logic gates, any Boolean Function can be implemented using these gates, are most very efficient and popular with static design style.

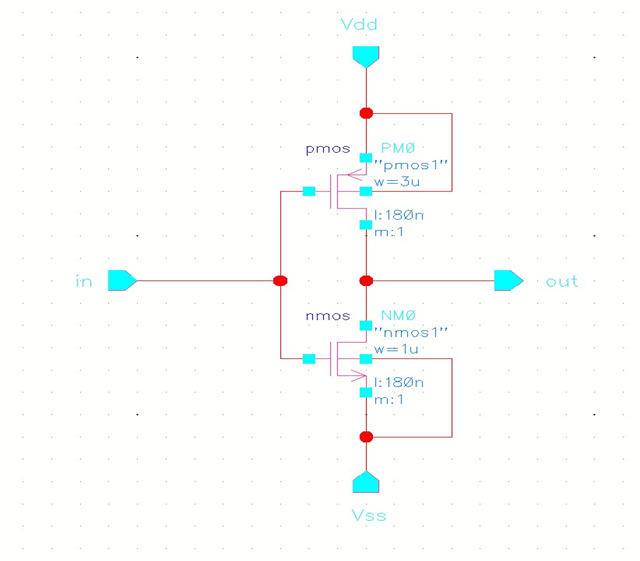
1. **Advantages of GDI:**
2. **Reduced transistor count:** GDI gates use fewer transistors, simplifying circuit design.
3. **Lower power dissipation:** GDI circuits consume less power due to fewer transistors.
4. **Faster operation:** Smaller node capacitances lead to higher speed.
5. **Reduced wiring complexity:** GDI gates occupy less area, resulting in fewer interconnection effects.

**6. software implementation**

# **Introduction**

This tutorial is an introduction to schematic capture and circuit simulation for ENGN1600 using Cadence Virtuoso. These courses use the NCSU FreePDK45 library for a 45nm technology. The NCSU library provides the models for a 45nm Bulk‐Si technology from Fujitsu (details about the PDK can be found at <http://www.eda.ncsu.edu/wiki/FreePDK45:Contents>). This tutorial will guide you in the process of designing CMOS circuits using both users defined, transistor level ]schematics. It will also walk you through simulating the circuits in Spectre. To launch Cadence Virtuoso (either on the instructional machines or on your laptop), you will need to connect to the Computation and Visualization cluster network at Brown (CCV). For additional details on how to connect to CCV and launch Cadence, please refer to the CCV tutorial accessible from the ENGN1600 course webpage. Note that only the first part of this tutorial about logging on to the CCV machines is relevant to you.

1. **Schematic Capture**



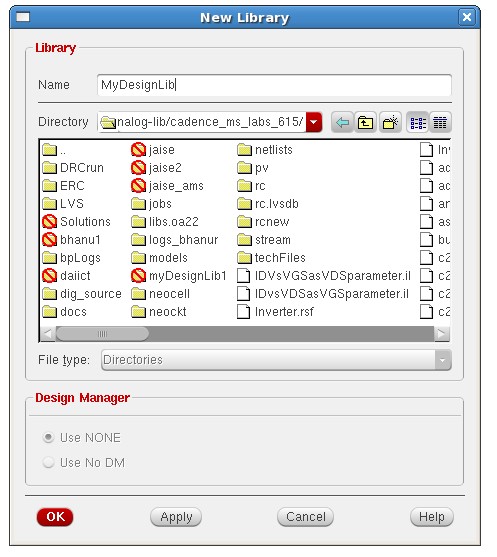
**Figure 6.1** Schematic of Inverter

1. **Schematic Entry**

To create a library and build a schematic of an Inverter

Below steps explain the creation of new library “**MyDesignLib**” and we will use the same throughout this course for building various cells that we going to create in the next labs.

1. **Creating a new library**
2. Execute **Tools – Library Manager** in the CIW (Command Interpreter Window) to open Library Manager.
3. In the Library Manager, execute **File – New – Library**. The new library form appears.
4. In the “New Library” form, type “**MyDesignLib**” in the Name section.

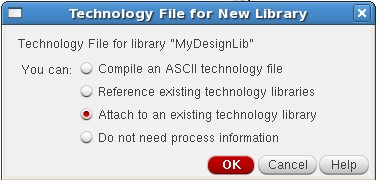


**Figure 6.2** Creating new library

1. In the field of Directory section, verify that the path to the library is set to **~/Database/cadence\_analog\_labs\_615** and click **OK**.

**Note:** New library an also be created directly from the CIW by executing **File - New – Library**

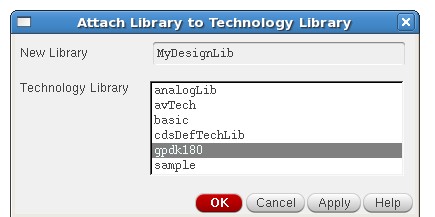
1. A “**Technology File for New library**” form appears, select option “**Attach to an existing technology library‖**and click **OK**.



**Figure 6.3** Technology File for new library

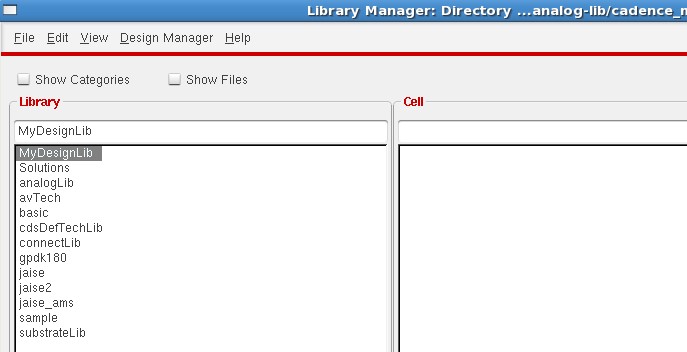
**Note:** A technology file is not required if you are not interested to do the layouts for the design.

1. A **―Attach library to Technology Library‖** form appears, select option **―gpdk180‖** from the cyclic field and click **OK**.

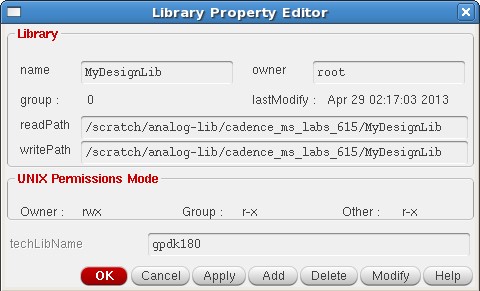


**Figure 6.4** Attach library

1. After creating a new library, we can verify it from the “**Library Manager‖**



**Figure 6.5.** Library Manager

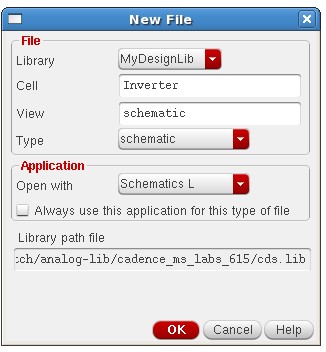
1. If we right clickon the “**MyDesignLib**” and select properties, we can find that “**gpdk180‖** library is attached as techLib to “**MyDesignLib**”.

**Figure 6.6.** Property editor

1. **Creating a Schematic Cell view**

In this section we will learn how to open new schematic window in the new “**MyDesignLib**” library and build the inverter schematic as shown in the figure at the start of this lab.

1. In the CIW or Library manager, execute **File – New – Cell view**.
2. Set up the **―New File‖** window as follows:



**Figure 6.7.** New file

**Note:** Do not edit the library path file and the one above might be different from the path shown in your window.

1. Click **OK** when done the above settings. A blank schematic window for the **Schematic**design appears.

**Note:** Make sure the library name is same as what we created earlier, **―MyDesignLib‖.**

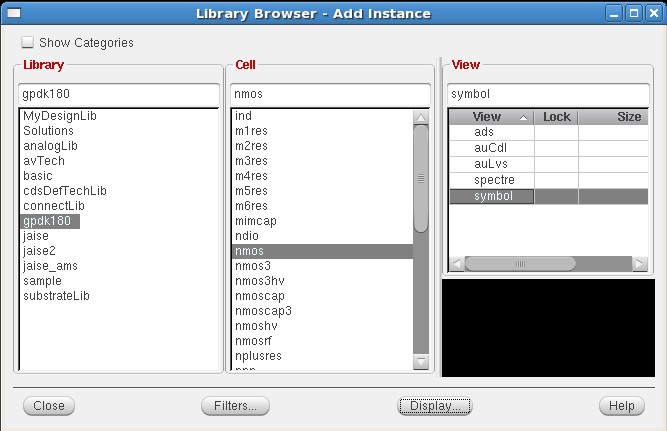
1. **Adding Components to schematic**

**Tip:** You can also execute **Create — Instance** or press **i**.

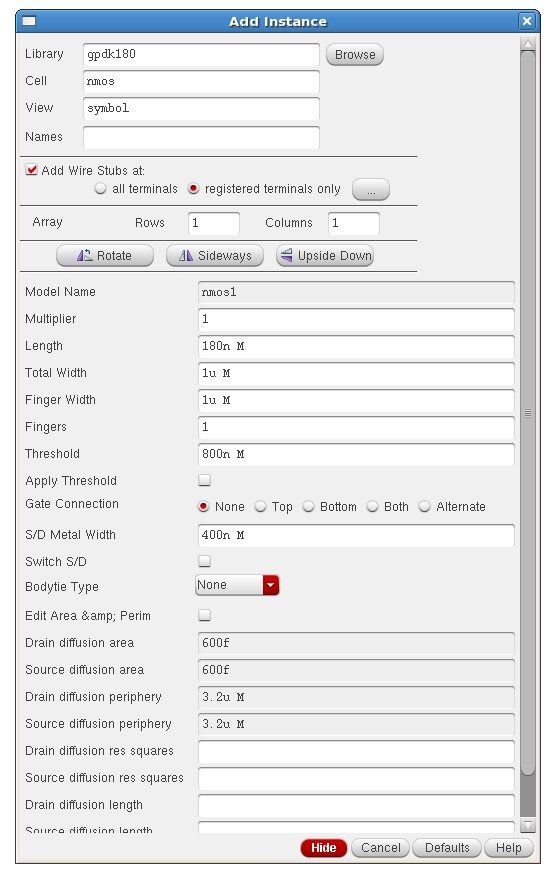


**Figure 6.8** Add Instance

1. Click on the **Browse** button. This opens a Library browser from which you can select components and the **symbol** view.



**Figure 6.9.** Library Browser



**Figure 6.10.** Instantiating the NMOS symbol

1. You will update the Library Name, Cell Name, and the property values given in the table on the next page as you place each component.
2. Instantiating the NMOS symbol is shown below as an example
3. After you complete the Add Instance form, move your cursor to the Schematic window and **left click** to place a component.
4. After entering components, click **Cancel** in the Add Instance form or press ***Esc.***  This is a table of components for building the Inverter schematic.

|  |  |  |
| --- | --- | --- |
| **Library name** | **Cell Name** | **Properties** |
| gpdk180 | PMOS | Total Width**= wp**, Length=180n |
| gpdk180 | NMOS | Total Width = 1u, Length=180n |

1. If you place a component with the wrong parameter values, use the **Edit— Properties— Objects** command to change the parameters.

Use the **Edit— Move** command if you place components in the wrong location.

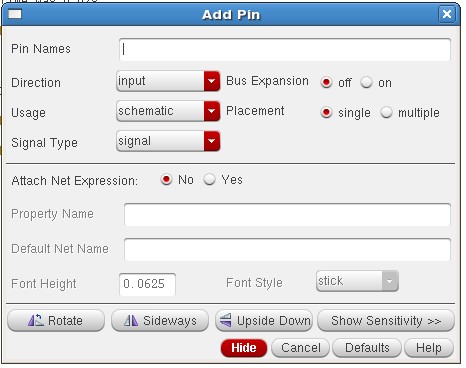


You can rotate components at the time you place them or use the **Edit— Rotate** command after they are placed.

1. **Adding pins to Schematic**
2. Click the **Pin** fixed menu icon in the schematic window. 

**Tip:** You can also execute **Create — Pin**or press **p**.

The Add pin form appears.



**Figure 6.11** Creating Pin

1. Type the following in the Add pin form in the exact order leaving space between the pin names.

|  |  |
| --- | --- |
| **Pin Names** | **Direction** |
| in, Vdd, Vss | input |
| out | output |

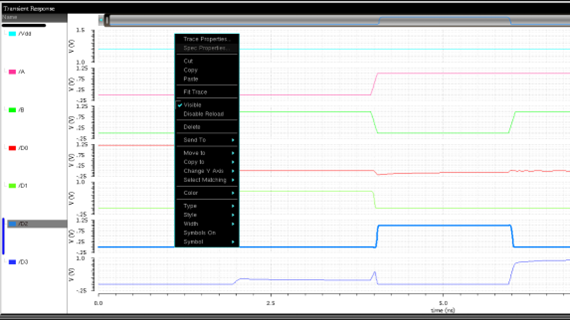
Make sure that the direction field is set to **input/output/input Output**when placing the **input/output/in out**pins respectively and the Usage field is set to **schematic**.

1. Select **Cancel** from the Add – pin form after placing the pins.

In the schematic window, execute **Window— Fit** or press fto fit all the components to the schematic editor window.

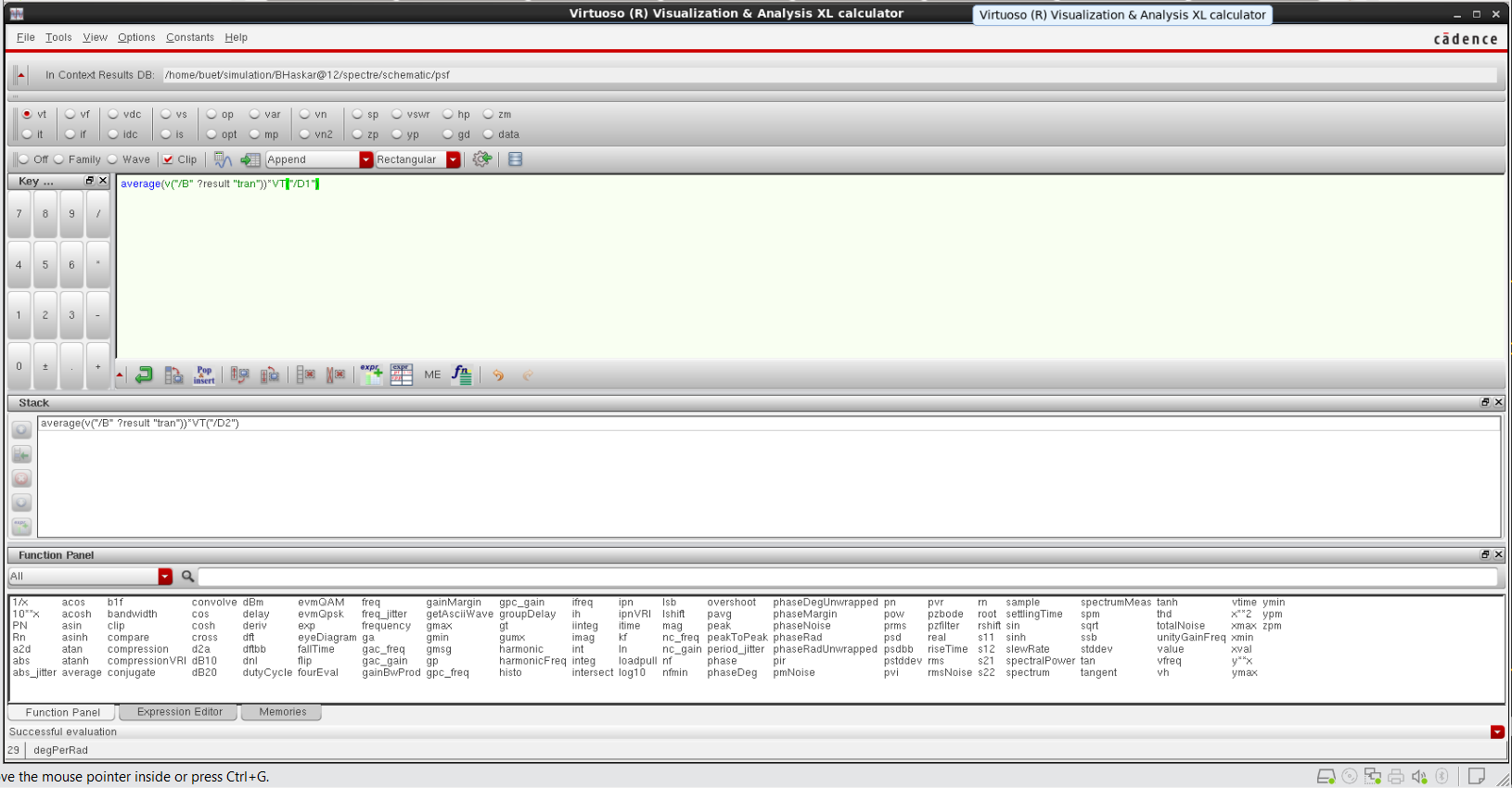


1. **Measuring Power Dissipation**
2. In the wave form window select any one of the following waveforms, click on right button and select “send to” to calculate the power Dissipation.

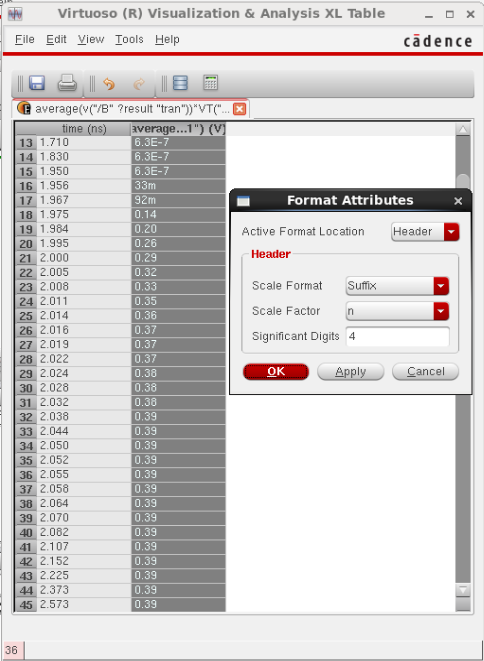


**Figure 6.12.** Navigating to Power Dissipation

1. After selecting any waveform, a Browser will appear to calculate the power dissipation of the circuit diagram.
2. Now, select “**Average**” to calculate the average of the power.
3. A command will appear on the browser to evaluate the average of the power.
4. The Visualization and analysis XL calculator page will appear in that select “average” and click on the calculator.

****

**Figure 6.13** Visualization and Analysis XL calculator

****

**Figure 6.14** Analysis XL Table

1. From the table we can calculate the Low Power from the Waveform, and we can change the type of power we need by right-clicking.

**7. SIMULATION RESULT****S**

1. **Analog Simulation with Spectre**

To set up and run simulations on the low Power Decoder design.

In this section, we will run the simulation for Inverter and plot the transient; DC characteristics and we will do Parametric Analysis after the initial simulation.

1. **Starting the Simulation Environment**
2. In the **Inverter\_Test** schematic window, execute **Launch – ADE L**

The **Virtuoso Analog Design Environment (ADE)** simulation window appears.

##### **Choosing a Simulator**

Set the environment to use the **Spectre® tool**, a high speed, highly accurate analog simulator.

Use this simulator with the **Inverter*\_*Test**design, which is made-up of analog components.

1. In the simulation window (ADE), execute **Setup— Simulator/Directory/Host**.
2. In the Choosing Simulator form, set the Simulator field to **Spectre** (Not spectres) and click **OK**.
3. **Setting the Model Libraries**

The Model Library file contains the model files that describe the NMOS and PMOS devices during simulation. In the simulation window (ADE L), execute **Setup - Model Libraries.**

A screenshot of a computer

Description automatically generated

**Figure 7.1.** Model library setup

To view the model file, highlight the expression in the **Model Files** field and Click **Edit File**.



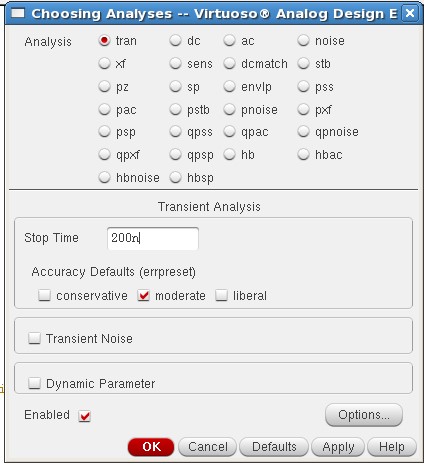
1. To complete the Model Library Setup, move the cursor and click **OK.**

##### **Choosing Analysis**

This section demonstrates how to view and select the different types of analysis to complete the circuit when running the simulation.

1. In the Simulation window (ADE), execute **Analysis - Choose Analysis** or click the A close up of a button

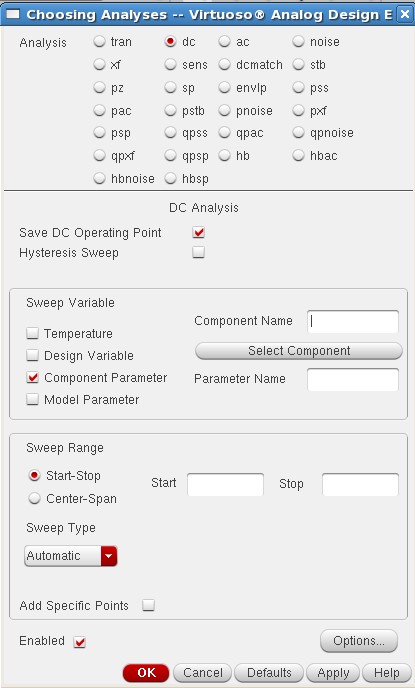
   Description automatically generated **Choose - Analysis**icon.
2. The Choosing Analysis form appears. This is a dynamic form, the bottom of the form changes based on the selection above.
3. To setup for transient analysis
   1. In the Analysis section select **tran**.
   2. Set the stop time as **200n**.
   3. Click at the **moderate** and **Enabled**buttons at the bottom, and then click **Apply**.



**Figure 7.2.** Choosing analyses

To set up for DC Analysis:

* 1. In the Analysis section, select **dc**.
  2. In the DC Analysis section, turn on **Save DC Operating Point.**
  3. Turn on the **Component Parameter**.
  4. Click the **Select Component**, which takes you to the schematic window.



**Figure 7.3** DC analysis setup

Select input signal **vpulse source** in the test schematic window.

* 1. Select **DC Voltage** in the “**Select Component Parameter‖** window and click **OK**.

A screenshot of a computer

Description automatically generated

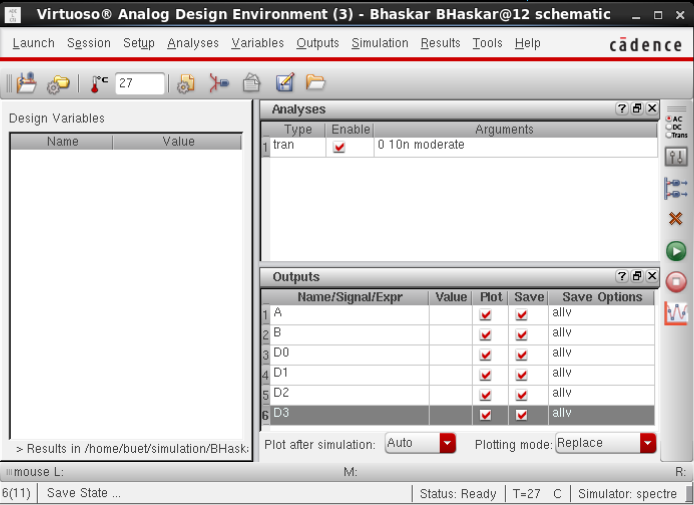
**Figure 7.4**. Select Component Parameter

* 1. In the analysis form, enter **start**and**stop** voltages as **0**to **1.8** respectively.
  2. Check the enable button and then click **Apply*.***

##### **Selecting Signals for Plotting**

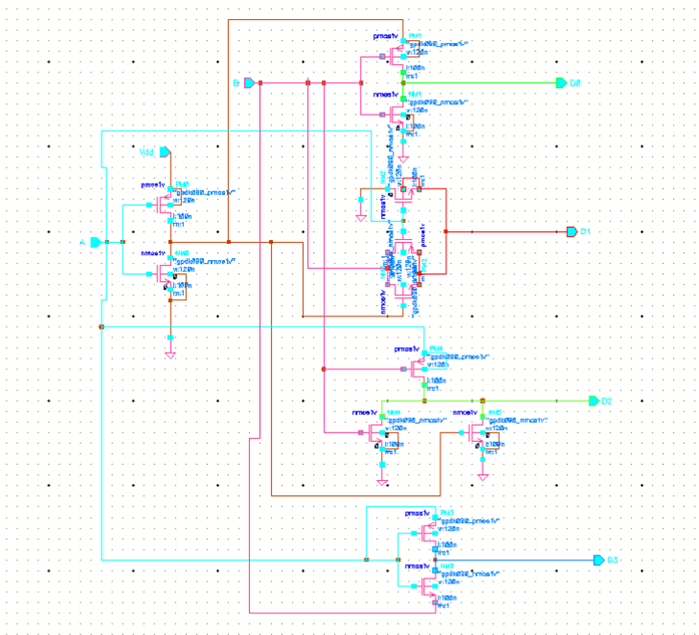
1. Execute **Outputs – To be plotted – Select on Schematic**in the simulation window.
2. Follow the prompt at the bottom of the schematic window, click on output net **Vout**, input net **Vin** of the Inverter\_Test. Press ***Esc*** with the cursor in the schematic after selecting it. After setting the transient and DC analysis and the signals for wave

plotting, the ADE L window will look like the figure below.

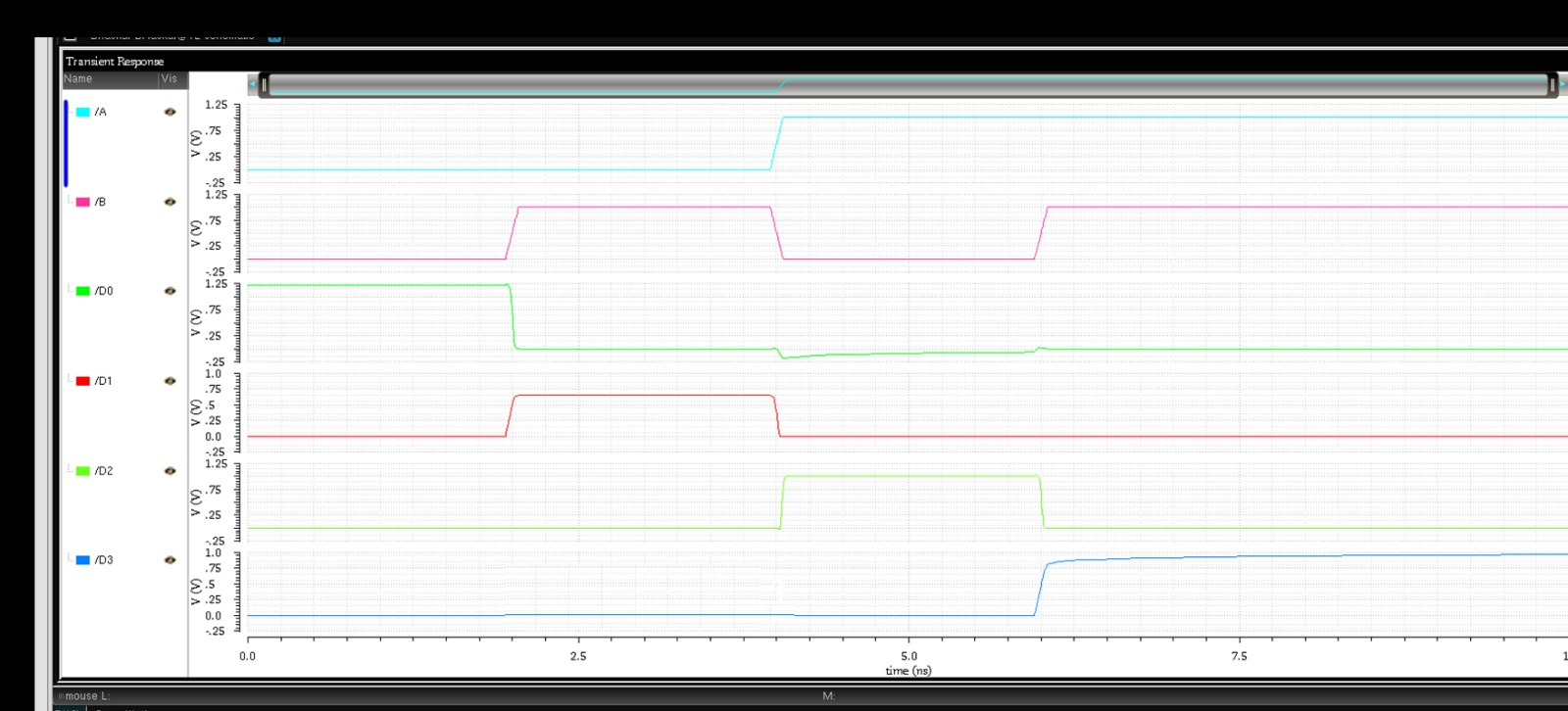


##### **Figure 7.5** Selecting Signals for Plotting

1. **Simulation Results**



**Figure 7.6** Schematic Diagram of 2-4 Decoder

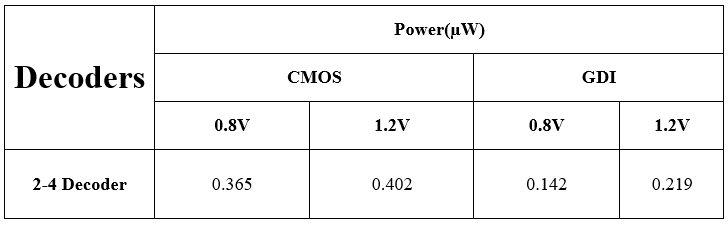


**Figure 7.7. Simulation Results**

1. **Comparison**

The CADENCE VIRTUOSO is used to calculate the power, delay, transistor count. All designs are compiled with 32nm technology, and a typical library of tanner EDA is used to get the results.

**Table 7.1:** POWER DISSIPATION FOR 2-4 DECODERS

****

**8. CONCLUSION AND FUTURE SCOPE**

**CONCLUSION**

The Gate Diffusion Logic (GDL) project aimed at reducing the transistor count by two and improving both power consumption and delay in comparison to existing systems. This conclusion will summarize the key findings and outcomes of the project, highlighting its significance and potential future directions.

**Reduction in Transistor Count**

The reduction in transistor count achieved through Gate Diffusion Logic represents a significant breakthrough in integrated circuit design. By streamlining the logic pathways and optimizing the gate structures, GDL enables complex functionality with fewer transistors. This reduction not only saves on silicon area and manufacturing costs but also contributes to improved yield and reliability.

Future research could explore advanced layout optimization techniques specific to GDL, leveraging machine learning algorithms or automated design tools to further minimize transistor count while maintaining performance and functionality. Additionally, investigating novel materials or structures that enable denser integration of GDL-based circuits could lead to even greater transistor count reductions and increased circuit complexity.

**Improved Power Consumption**

The enhanced power efficiency of GDL-based circuits is of paramount importance in today's energy-conscious world. By reducing active transistor count and optimizing signal paths, GDL significantly lowers power consumption without compromising performance. This is particularly advantageous for battery-operated devices, IoT applications, and high-performance computing systems.

Future efforts may focus on dynamic power management techniques tailored to GDL designs, such as adaptive voltage scaling or clock gating strategies. Exploring low-power variants of GDL, such as near-threshold logic (NTL) implementations, could further enhance power efficiency for ultra-low-power applications. Additionally, research into power-aware design methodologies specific to GDL could provide designers with comprehensive tools to optimize energy consumption at various design stages.

**Reduction Delay**

Minimizing signal propagation delays is essential for ensuring rapid response times and high-speed operation in electronic systems. The Gate Diffusion Logic project's success in reducing delays through logical simplification and optimized gate structures is a significant achievement. This contributes to improved system throughput, reduced latency, and enhanced overall performance.

Future directions in this area could include exploring advanced timing optimization techniques for GDL-based designs, such as delay balancing algorithms or clock skew minimization strategies. Investigating alternative circuit topologies or architectures that leverage GDL principles could also lead to further reductions in signal propagation delays while maintaining design scalability and flexibility.

**Significance and impact**

The significance of the Gate Diffusion Logic project extends beyond its immediate technical achievements. It underscores the continuous evolution and innovation within the semiconductor industry, driving towards more efficient, reliable, and sustainable electronic systems. The impact of GDL can be felt across a wide range of applications, from mobile devices and consumer electronics to automotive electronics and data centers.

Looking ahead, collaborations between academia, industry, and research institutions can foster cross-disciplinary advancements in GDL and related technologies. Standardization efforts and knowledge sharing initiatives can accelerate the adoption of GDL methodologies, benefiting designers, manufacturers, and end-users alike. Moreover, addressing the environmental impact of semiconductor manufacturing and design processes aligns with global sustainability goals, making GDL an integral part of future technology ecosystems.

In conclusion, the Gate Diffusion Logic project represents a paradigm shift in integrated circuit design, offering a synergistic blend of reduced transistor count, improved power efficiency, and reduced delays. Its impact spans technical, economic, and environmental dimensions, driving forward the trajectory of semiconductor technology. Embracing a collaborative and forward-thinking approach will continue to unlock the full potential of Gate Diffusion Logic and pave the way for the next generation of electronic system.

**FUTURE SCOPE**

In Future MOSFET can be replaced by CNTFET & GNRFET for Low Power Applications

**bibliography**

[1] N. Sowjith, K. Sandeep, S., Sumanth, M., and S. Agrawal, “Low power VLSI architecture for combined FMO/Manchester encoder for reusability and FMO/Manchester codecs”, IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016.

[2] Bissa P.R., Pande K.S., “All Digital Phase Locked Loop for Low Frequency Applications”, International Conference on Advances in Computing Communications and Informatics,ICACCI 2018.

[3] Mathew , Krishnamurthy ,Anders, Hsu and Borkar, “Advanced circuits techniques for high-performance microprocessor and low power DSPs”, Proceedings of the IEEE Dallas/CAS Workshop on Implementation of high-performance circuits, pp. 1-45, September 2004.

[4] N. H. E. Waste and D. M. Harris, CMOS VLSI Design, a Circuits and Systems perspective, 4th ed. Boston, MA, USA: Addison-Wesley, 2011.

[5] R. RogeNMOSer, H. Kaeslin, and N. Felber, ―The impact of transistor sizing on power Efficiency in submicron CMOS circuits,‖in Proc.22nd

[6] V. G. Oklobdzija and B. Duchene, ―Pass-transistor dual value logic for low-power CMOS,‖ InProc. Int. Symp. VLSI Technol.,pp.341–344,1995.

[7] M. Suzuki et al., ―A 1.5 ns 32b CMOS ALU in double pass transistorlogic,‖Proc.IEEE Int.Solid-State Circuits Conf.,pp.90–91,1993.

[8] X. Wu, “Theory of transmission switches and its application to design of CMOS digital circuits,” International Journal on Circuit Theory and Appllications, vol. 20, no. 4, pp. 349–356, 1992

[9] D. Balobas and N. Konofaos “Design of Low Power, High Performance 2-4 and 4-16 Mixed Logic Line Decoders” IEEE transaction on circuits and system –II :express briefs, VOL. 64, NO.2, February 2017.

[10]Arkadiy morgenshtein, Alexander fish & Israel Wagner, “Gate Diffusion input (GDI): A power efficient method for digital combinatorial circuits”, IEEE Transaction on very large-scale integration (VLSI) systems vol.10, no. 5 October 2002.

[11] Arkadiy Morgenshtein, Idan Shwartz and Alexander Fish, “Gate Diffusion Input (GDI) Logic in Standard CMOS Nanoscale Process ” 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel

[12] Arvind Kumar Mishra, Debiprasad Priyabrata Acharya, and Pradip Kumar Patra “Novel Design Technique of Address Decoder for SRAM”, IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), ISBN No. 978-1-4799-3914- 5/14/$31.00 ©20141EEE.

[13] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen,―LowpowerCMOS digital design,‖IEEE J. Solid-State Circuits,vol. 27, pp. 473–484, Apr. 1992.

[14] A. P. Chandrakasan and R.W. Brodersen,―Minimizing power consumption in digital CMOS circuits,‖Proc. IEEE, vol. 83, pp.498–523, Apr.1995.

[15] Pranshu Sharma, Anjali Sharma, Richa Singh,―Design and Analysis of Area and Power Efficient1-Bit FullSubtractor using 120nmTechnology‖, in International Journal of Computer Applications (IJCA), Vol. 88, No.12,February 2014,pp.33-42.

[16] Prabhu E. and Mangalam, H., “Power Optimized Vedic Parallel MAC Unit: GDI Technique”, Asian Journal of Information Technology, vol. 15, no. 16, pp. 2954-2957, 2016. [17] Mohan Shoba, Rangaswamy Nakkeeran, “GDI based full adders for energy efficient arithmetic applications”, International Journal on Engineering Science and Technology, pp. 485-496, 2016.

[18] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, “Precomputation-based sequential logic optimization for low power,” IEEE Trans. VLSI Syst., vol. 2, pp. 426–435, Dec. 1994.