

ANALYSIS OF EFFICIENT PARALLEL PIPELINED ARCHITECTURE FOR LOW POWER APPLICATIONS

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

CHEGONI ANUSHA (40130707)

CHAITANYA LAKSHMI (40130256)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY -1 UNIVERSITY BY UGC**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

APRIL – 2024

ANALYSIS OF EFFICIENT PARALLEL PIPELINED ARCHITECTURE FOR LOW POWER APPLICATIONS

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

CHAITANYA LAKSHMI (40130256)

CHEGONI ANUSHA (40130707)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CHAITANYA LAKSHMI (40130256)** and **CHEGONI ANUSHA (40130707)** who carried out the project entitled "**ANALYSIS OF EFFICIENT PARALLEL PIPELINED ARCHITECTURE FOR LOW POWER APPLICATIONS**" under our supervision from November 2023 to April 2024.

Internal Guide

Dr. R Narmadha, M.E, Ph.D.,

Head of the Department

Dr.T. RAVI M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, CHAITANYA LAKSHMI (40130256) and CHEGONI ANUSHA (40130707) hereby declare that the Project Report entitled "ANALYSIS FOR EFFICIENT PARALLEL PIPELINED ARCHITECTURE FOR LOW POWER APPLICATIONS" done by me under the guidance of Dr. R. Narmadha, M.E., Ph.D., Associate Professor is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE: CHENNAI

Anusha...

Y. C. Lakshmi

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Dean, School of Electronics and Communication Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R Narmadha, M.E, Ph.D., Associate Professor, Department of Electronics and Communication Engineering**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The ever-growing demand for portable and battery-powered devices necessitates the development of low-power Very-Large-Scale Integration (VLSI) architectures. This project investigates the potential of parallel pipelined architectures for achieving high performance while minimizing power consumption. The proposed approach leverages pipelining, a technique that divides a complex task into smaller, sequential stages. Each stage operates on independent data, enabling parallel processing and improved throughput. Additionally, the project explores the utilization of sleep techniques. By strategically placing sleep transistors within the pipeline stages, inactive portions of the circuit can be powered down, leading to significant power savings. This project analyzes the trade-offs between parallelism, throughput, and power consumption in a pipelined architecture. The effectiveness of various sleep techniques in reducing power while maintaining performance will be evaluated. The project aims to identify an optimal design that balances these crucial aspects, paving the way for efficient and low-power VLSI implementations for future applications.

INDEX

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	V
	LIST OF FIGURES	VIII
	LIST OF SYMBOLS AND ABBREVIATIONS	X
	LIST OF TABLES	XI
1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Introduction of Three Operand Adders	2
	1.3 CMOS Technology	4
	1.3.1 NMOS	4
	1.3.2 PMOS	5
	1.3.3 CMOS Gate	6
	1.3.4 CMOS Logic Gates	6
2	LITERATURE SURVEY	8
	2.1 Literature Survey	8
	2.2 Literature Summary	11
	2.3 Problem Definition	11
3	EXISTING SYSTEM	12
	3.1 Multiplexer	12
	3.2 Register	13

	3.3 Flip Flops	15
	3.4 Parallel Pipelined Architecture	16
4	PROPOSED SYSTEM	19
	4.1 D Flipflop	19
	4.2 Leakage Power Reduction Technique	21
	4.3 Standards	24
	4.4 Constraints	24
	4.5 Trade-Offs	25
5	RESULTS AND DISCUSSION	26
	5.1 Existing System Simulation Outputs	26
	5.2 Proposed System Simulation Outputs	28
	5.3 Tabulation	31
6	CONCLUSION	32
	6.1 Conclusion	32
	REFERENCES	33

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	NMOS Transistor	4
1.2	PMOS Transistor	5
1.3	CMOS Logic Gate	6
1.4	CMOS Inverter	7
1.5	CMOS Nand Gate	7
3.1	Multiplexer	12
3.2	Register	14
3.3	Types of Flip Flop	15
3.4	3 Parallel Pipelined Architecture Diagram	17
3.5	3 Parallel Pipelined Block Diagram	18
4.1	Schematic of D-FF Using Sleep Transistor Technique	20
4.2	Schematic of Sleep Technique	22
5.1	2x1 Multiplexer	26
5.2	D Flipflop Without Using Sleep Technique	26
5.3	Step A Without Using Sleep Technique	27
5.4	Step B Without Using Sleep Technique	27
5.5	Final Output Without Using Sleep Technique	28
5.6	2x1 Multiplexer	28
5.7	D Flipflop Using Sleep Technique	29
5.8	Step A Using Sleep Technique	29

5.9	Step B Using Sleep Technique	30
5.10	Final Output Using Sleep Technique	30

LIST OF SYMBOLS AND ABBREVIATIONS

PRBG	Pseudo-Random Bit Generator
LCG	Linear Congruential Generator
CLCG	Coupled LCG
MDCLCG	Modified Dual CLCG

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
5.1	Analysis of Power, Delay & Power Delay Product without using "Sleep Technique"	31
5.2	Analysis of Power, Delay & Power Delay Product with using "Sleep Technique"	31

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The relentless miniaturization of transistors in Very-Large-Scale Integration (VLSI) circuits has ushered in a new era of computational power and functionality. However, with this advancement comes a growing concern: power consumption. As devices become smaller and more complex, their energy demands rise, posing challenges for battery life and heat dissipation in both portable and high-performance applications. This necessitates a paradigm shift in VLSI design, prioritizing architectures that deliver exceptional performance while minimizing power consumption.

One promising approach to address this challenge lies in the realm of parallel pipelined architectures. Parallel processing harnesses the power of multiple processing units working concurrently, significantly accelerating computations. Pipelining, on the other hand, decomposes a complex task into smaller, independent stages that can be executed in a sequential pipeline fashion. By strategically combining these techniques, parallel pipelined architectures offer a compelling solution. Pipelining enables concurrent processing within each stage of the parallel architecture, further boosting throughput and efficiency.

However, traditional parallel pipelined architectures can still incur significant power consumption due to the constant activity within processing units. This necessitates the exploration of innovative strategies to mitigate power dissipation without compromising performance. The following sections of this work will delve deeper into the design principles of parallel pipelined architectures, analyze their inherent power efficiency, and explore potential optimization techniques for achieving optimal low-power operation in VLSI applications.

The fusion of parallelism and pipelining in an architecture can significantly improve throughput and, crucially, reduce the power requirements of the system. This project will focus on analyzing the architectural aspects that enable efficient parallel processing and pipelining, considering various design parameters and trade-offs. The core constituents of this investigation encompass a profound

comprehension of the theoretical foundations governing parallel pipelined architectures, a thorough exploration of extant designs with a meticulous assessment of their performance metrics, and the proposition of enhancements to attain an even higher echelon of efficiency. Additionally, we intend to assess the ramifications of parallelism and pipelining on power consumption, performance, and overall system throughput.

1.2 INTRODUCTION TO THREE OPERAND ADDERS

The three-operand binary addition can be carried out either by using two two-operand adders or one three-operand adder. Carry-save adder (CS3A) is the area-efficient and widely adopted technique to perform the three-operand binary addition in the modular arithmetic used in cryptography algorithms to enhance the security operations. However, the longer carry propagation delay in the ripple-carry stage of CS3A seriously influences the performance of the MDCLCG and other cryptography architectures on IoT based hardware devices.

Several random numbers theory have been introduced in the last several decades. Linear congruential generator (LCG) that introduced 1954 by Lehmer is the oldest and the most widely used pseudorandom number generator (PNG). Park & Miller suggest good parameters for LCG. The suggestion is used in Matlab for generating uniform random numbers.

Hardware for generating random number also available as well as its algorithm. The hardware has been used since 2008. LETech is the fastest among all hardware for computing random numbers, this hardware has been developed since 2008. Research for finding the suitable algorithm of generating random number is well establish field until now. Many researchers use field programmable gate arrays (FPGA) for testing their ideas. Several of these has been realized into hardware and sell into the market.

Initially, the algorithm of LCG combined with MonteCarlo method has been used for generating non uniform random numbers using Matlab. After that, we develop the circuit of random number generator and implemented in FPGA. The increment factor (c) has been ignored ($c = 0$). Besides, we present novel design and FPGA implementation of LCG algorithm without ignoring the increment. The rest of this work report are organized as follows. Section II deals with theory of LCG algorithm.

The design circuit of LCG for FPGA implementation and nets connections are covered in section III. Section IV provides some implementations data and analysis. Finally, the conclusions are viewed in section V.

The three-operand binary addition is also a primary arithmetic operation in the linear congruential generator (LCG) based pseudo-random bit generators (PRBG) such as coupled LCG (CLCG), modified dual-CLCG (MDCLCG) and coupled variable input LCG (CVLCG) [11]. Modified dual-CLCG (MDCLCG) is the most secure and highly random PRBG method among all the LCG-based and other existing PRBG methods. It is polynomial-time unpredictable and secure if $n \geq 32$ -bits. Therefore, the security of the MDCLCG enhances with the increase of operand size. However, the area and critical path delay increases linearly since its hardware architecture consists of four three-operand modulo- $2n$ adders, two comparators, four multiplexers area. Hence, the performance of the MDCLCG can be improved by the efficient implementation of the three-operand adder.

The three-operand binary addition can be carried out either by using two two-operand adders or one three-operand adder. Carry-save adder (CS3A) is the area-efficient and widely adopted technique to perform the three-operand binary addition in the modular arithmetic used in cryptography algorithms and PRBG methods. However, the longer carry propagation delay in the ripple-carry stage of CS3A seriously influences the performance of the MDCLCG and other cryptography architectures on IoT based hardware devices. To shorten the critical path delay, a parallel prefixed two-operand adder such as Han-Carlson (HCA) can also be used for three-operand binary addition.

It reduces the critical path delay in the order of $O(\log_2 n)$ but increases the area in the order of $O(n \log_2 n)$ [15]. Therefore, it is necessary to develop an efficient VLSI architecture to carry out the fast three-operand binary addition with minimum hardware resources. Hence, a new high-speed area-efficient adder technique is proposed using pre-compute bitwise addition followed by carry-prefix computation logic to perform the three-operand addition in this paper that consumes considerably less gate area while minimizing the propagation delay in comparison to the HCA-based three-operand adder (HC3A). Furthermore, the proposed adder architecture is implemented with the Verilog HDL, and then synthesized with commercially available 32nm CMOS technology library.

1.3 CMOS TECHNOLOGY

CMOS technology is a predominant technology for manufacturing integrated circuits. CMOS stands for “Complementary Metal Oxide Semiconductor”. Microprocessors, batteries, and digital sensors among other electronic components make use of this technology due to several key advantages. This technology uses both NMOS and PMOS to realize various logic functions. Both N and P MOSFET channels are designed to have matching characteristics.

Before CMOS, PMOS and NMOS logic were widely used for implementing logic gates. PMOS was then replaced by the NMOS Technology, which used to be the standard IC fabrication technology. Initially, CMOS was slower and more expensive than NMOS. The main advantages of NMOS technology are simple physical process, functional density, processing speed and manufacturing efficiency.

1.3.1 NMOS

N-channel MOSFET consists of an N-type source and drain diffused on a P-type substrate. The majority carriers are electrons. When the applied voltage to the gate is high enough, the NMOS will conduct; otherwise, it will not. Since the majority carriers (electrons) travel faster than holes, NMOS are faster than PMOS.

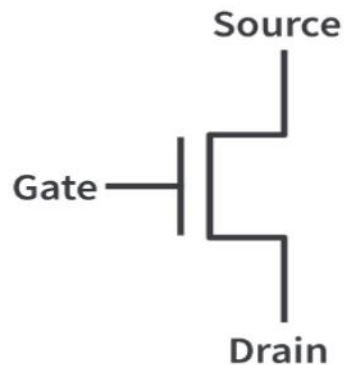


Fig 1.1: NMOS transistor

NMOS, or N-type Metal-Oxide-Semiconductor, is a fundamental building block in modern electronics. This type of transistor relies on electrons, negative charges, to conduct current. NMOS transistors act like switches controlled by

voltage. A positive voltage applied to the gate terminal turns the transistor on, allowing electrons to flow between the source and drain. Conversely, zero or negative voltage on the gate blocks electron flow, turning the transistor off.

NMOS transistors are frequently paired with PMOS (P-type Metal-Oxide-Semiconductor) transistors to form CMOS (Complementary Metal-Oxide-Semiconductor) logic circuits. CMOS circuits offer advantages over NMOS alone, including improved energy efficiency. This combination is the cornerstone of most modern integrated circuits.

1.3.2 PMOS

P-channel MOSFET also has a Source and Drain diffused on a substrate. The Source is P-type while the substrate is N-type. The majority carriers are holes. PMOS will conduct when a low voltage is applied. When a high voltage is applied to the gate, the PMOS will not conduct.

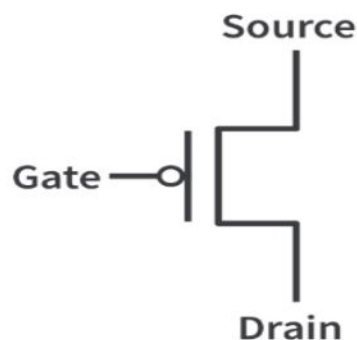


Fig 1.2: PMOS Transistor

PMOS transistors share a similar structure with NMOS, having three terminals: source, gate, and drain. However, their operation is based on an opposite voltage logic. When a negative voltage is applied to the gate of a PMOS transistor, it attracts electrons away from the channel, creating more holes. These holes act as the current carriers, allowing current to flow between the source and drain. This negative voltage effectively turns the PMOS transistor on. Conversely, applying a zero or positive voltage to the gate repels holes, hindering current flow and turning the transistor off.

1.3.3 CMOS

Since CMOS technology uses both N-type and P-type transistors to design logic functions, a signal which turns ON a transistor type is used to turn OFF the other transistor type. This eliminates the need for pull-up resistors in favour of simple switches.

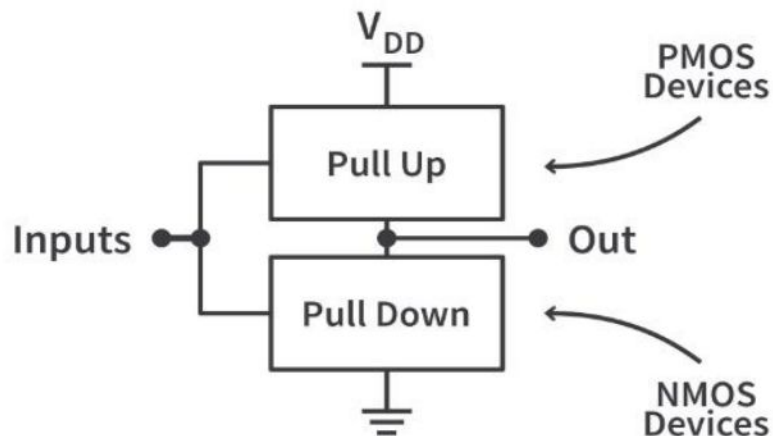


Fig 1.3 CMOS logic gate

In CMOS logic gates N-type MOSFETs are arranged in a pull-down network between the output and the low voltage supply rail (VSS or ground) while P-type MOSFETs are in a pull-up network between the output and the higher-voltage rail (often VDD). Thus, the N-type MOSFET will be ON when the P-type MOSFET is OFF, and vice-versa. For any input pattern, one of the networks is ON and the other is OFF. CMOS advantages are high speed, low power dissipation, high noise margins in both states, and a wide range of source and input voltages (fixed source voltage).

1.3.4 CMOS Logic Gates

1) CMOS Inverter

The CMOS inverter is the simplest CMOS logic gate. The circuit consists of PMOS and NMOS FET. Input A serves as the gate voltage for both transistors while Y is the output.

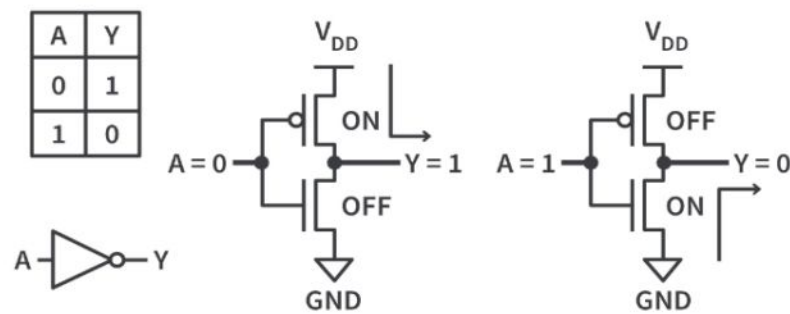


Fig 1.4: CMOS Inverter

The NMOS transistor has an input from V_{SS} or ground and the PMOS transistor has an input from V_{DD} . When the input (A) is low ($<V_{DD}$, ~ 0 V, Logic 0), the NMOS is OFF while the PMOS is ON. V_{DD} will appear at the output through the P-channel MOSFET path. Hence, there is output (Logic 1) with the circuit pulled up to V_{DD} . When the input is high ($\sim V_{DD}$, Logic 1), the PMOS is OFF while the NMOS is ON. The output is pulled down and is therefore low (Logic 0).

2) CMOS NAND Gate

A 2-input NAND gate has two N-channel MOSFETs connected in series between Y (output) and GND and two P-channel MOSFETs connected in parallel between V_{DD} and Y.

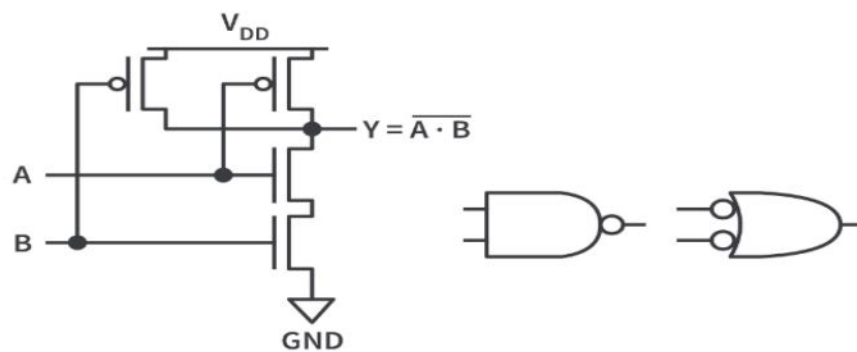


Fig 1.5: CMOS NAND Gate

If either A or B is low (Logic 0), at least one of the NMOS transistors will be OFF. This breaks the path from Y to GND since the NMOS transistors are connected in series. But in this case at least one of the PMOS transistors is ON, completing a path from Y to V_{DD} .

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

B. Zhang, Z. Ma and W. Luo, the authors are introducing the context and background for their research on matrix transposition algorithms and hardware architectures. They mention the importance of matrix transposition in various fields, such as image signal processing and artificial intelligence, particularly in deep learning applications. The authors discuss previous work in the field, highlighting existing algorithms and hardware architectures for matrix transposition. They note that some previous designs support square matrices, while others focus on non-square matrices, but each has limitations. They emphasize the need for an architecture that can efficiently transpose matrices with rows and columns that are integer multiples of a given parameter "P."

The text also briefly outlines the structure of the paper, mentioning that it will cover basic pipelined algorithms, hardware architectures, control strategies, latency, memory, and comparisons with existing architectures. In summary, the literature survey in this section provides context for the research, highlights the importance of matrix transposition, and briefly reviews previous work in the field, setting the stage for the presentation of the authors' new algorithm and architecture.

Y. Wang, Z. Ma and F. Yu, the author introducing the literature survey presented in this abstract provides an overview of the existing research and developments in the field of matrix transposition algorithms and architectures. Here's a breakdown of what the literature survey covers:

- **Introduction to Matrix Transposition:** The survey starts by introducing the concept of matrix transposition and its significance in various applications such as signal processing, image compression, and video coding.
- **Previous Approaches:** It reviews various approaches and algorithms proposed in previous research for matrix transposition. These include methods like fast methods for specific matrix sizes, cache-efficient approaches, systolic arrays, and register allocation schemes.

- **Comparison of Previous Work:** The survey compares and contrasts these previous approaches based on criteria such as memory usage, latency, and their suitability for different scenarios.
- **Challenges and Limitations:** It mentions the challenges and limitations of some existing methods, such as sub-optimality in memory or latency, and restrictions on matrix size.
- **Proposed Solution:** The survey then introduces the novel pipelined algorithm and modular architectures proposed in the research. It highlights the key features of these architectures, including their optimality in terms of memory and latency.
- **Parallel Implementations:** The survey discusses how the proposed algorithm and architecture can be extended to support parallel implementations, including both N-parallel and p-parallel designs.
- **Comparison with Existing Architectures:** It provides a comparative analysis of the proposed architectures with existing ones, emphasizing the advantages of the new designs in terms of flexibility and efficiency.
- **Experimental Results:** The survey briefly mentions experimental results, showcasing the practical implementation of the proposed architecture on FPGA hardware.

The informative overview of the state of the art in matrix transposition algorithms and architectures, leading up to the introduction of the novel approach presented in the research.

B.Zhang,Z.Ma and F.Yu, The authors introducing the literature survey in this abstract provides an overview of the existing research and developments related to non-square matrix transposition. Here are the key points covered in the literature survey:

- **Introduction to Matrix Transposition:** The survey starts by introducing the concept of matrix transposition and its importance in various applications, such as signal and image processing, fast Fourier transform (FFT), and convolutional neural networks (CNNs).
- **Previous Approaches for Square Matrices:** It mentions that previous research has focused on transposing square matrices and highlights the work of

Järvinen et al. and Wang et al., who proposed algorithms and architectures for square matrix transposition with minimal memory and latency.

- **Importance of Non-Square Matrices:** It emphasizes that in practical applications, non-square matrices are common and sometimes more convenient to work with than square matrices. This motivates the need for transposing non-square matrices efficiently.
- **Previous Work on Non-Square Matrices:** The survey mentions previous work on non-square matrix transposition, including some design methods mentioned and, as well as the work of Garrido and Pirsch. It notes that while Garrido's work addressed non-square matrix transposition, it was not optimal in terms of memory or latency.
- **Proposed Contribution:** The literature survey introduces the main contribution of the paper, which is a novel pipelined algorithm and architecture for non-square matrix transposition. It highlights that the proposed architecture achieves theoretical minimums for both memory and latency and can be controlled using a simple strategy.
- **Parallel Implementations:** It mentions that the proposed algorithm and architecture support both serial and N-parallel implementations for matrix transposition.
- **Comparison with Existing Architectures:** The survey provides a comparison with existing matrix transposition architectures, such as those proposed by Järvinen et al., Wang et al., and Garrido. It highlights the advantages of the proposed architecture in terms of resource utilization and performance.
- **Experimental Results:** It briefly mentions the experimental results obtained through FPGA implementation, showing that the proposed architecture outperforms Garrido's design in terms of resource utilization, latency, and throughput. The literature survey sets the context for the proposed work by discussing the importance of non-square matrix transposition, summarizing previous research, and highlighting the contributions and advantages of the proposed algorithm and architecture.

2.2 LITERATURE SUMMARY

To implement the reduced implementation complexity presented by B. Zhang, Z. Ma, W. Luo, and Y. Wang, F. Yu offer an expert examination of the landscape of matrix transposition algorithms and hardware architectures. B. Zhang, Z. Ma, and W. Luo meticulously outline the pivotal role of matrix transposition in diverse domains, particularly image signal processing and deep learning, while underscoring the need for optimized architectures accommodating matrices with specific dimensional requirements. The authors judiciously emphasize the necessity for an architecture adept at efficiently transposing matrices with dimensions as integer multiples of a designated parameter, 'P.' Meanwhile, Y. Wang, Z. Ma, and F. Yu delve into a comprehensive overview, presenting a meticulous breakdown of the existing research in matrix transposition. They elucidate the critical importance of matrix transposition in fundamental applications like signal processing and convolutional neural networks. The survey deftly categorizes previous approaches, providing a discerning analysis based on memory utilization, latency, and adaptability across different contexts. Notably, the surveys converge in introducing novel contributions, showcasing the authors' expertise in proposing innovative pipelined algorithms and modular architectures aimed at addressing the deficiencies in existing designs. These insights collectively lay the groundwork for advancing the field, highlighting the transformative potential of optimized matrix transposition algorithms and architectures.

2.3 PROBLEM DEFINITION

To analyze an efficient parallel pipelined architecture for low power applications. This project aims to analyze and optimize a parallel pipelined architecture for low power applications, leveraging sleep techniques. This involves identifying power consumption, designing efficient sleep mechanisms, and evaluating their impact on system performance. The objective is to develop a robust solution that minimizes power consumption without compromising application efficiency, addressing the pressing need for energy-efficient computing in various domains.

CHAPTER 3

EXISTING SYSTEM

3.1 MULTIPLEXER

A multiplexer, commonly abbreviated as "mux," is a fundamental digital electronic circuit that operates as a data selector or data routing device. It allows multiple input data lines to be selectively routed to a single output line based on the binary pattern presented on control or selection lines. The selection of a specific input line is determined by the digital values on the control lines, enabling efficient data handling and transfer in various applications. A multiplexer plays a crucial role in combining, selecting, and directing data from multiple sources to a single output, optimizing data transfer, and enhancing system efficiency.

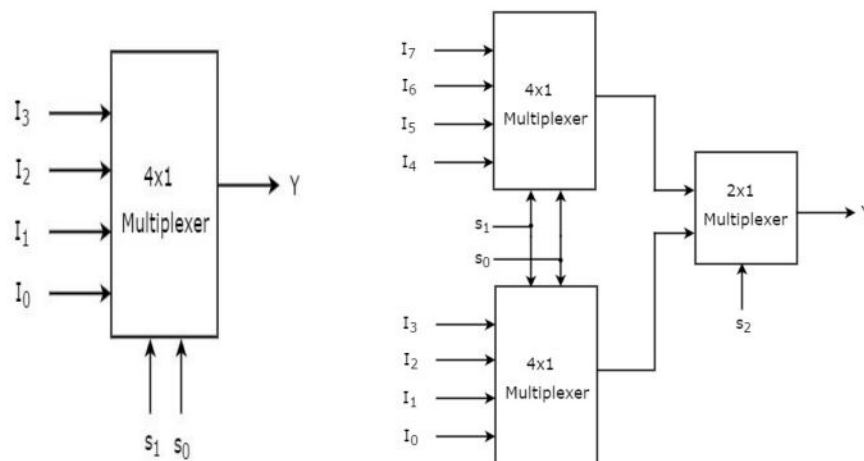


Fig 3.1: Multiplexer (4:1 MUX, 8:1 MUX)

3.1.1 Types of Multiplexers:

Based on Input/Output Lines:

- 4-to-1 Multiplexer: Has 4 input lines and 1 output line.
- 8-to-1 Multiplexer: Has 8 input lines and 1 output line.

Based on Control Lines:

- 2x1 Multiplexer: One control line ($m = 1$).
- 4x1 Multiplexer: Two control lines ($m = 2$).
- 8x1 Multiplexer: Three control lines ($m = 3$).

Based on Bit Width:

- n-bit Multiplexer: Has 2^n data input lines.

3.1.2 Applications of Multiplexers:

- Data Routing: Multiplexers are extensively used to route data from multiple sources to a single destination, optimizing data transfer and reducing hardware complexity.
- Memory Addressing: In memory systems, multiplexers assist in selecting memory cells by enabling dynamic addressing and efficient access to stored information.
- Arithmetic Logic Units (ALUs): Multiplexers play a critical role in ALUs, aiding in operations such as addition, subtraction, and bitwise logic by selecting appropriate inputs.
- Digital Signal Processing (DSP): In DSP applications, multiplexers are employed to switch between various signal processing operations, enhancing signal efficiency and processing speed.
- Bus Systems: Multiplexers are used in buses to direct data flow between various components in a computer system, optimizing data transfer within the system.

3.2 REGISTER

The Registers are crucial components that enable efficient data handling, movement, and manipulation within digital systems. Registers play a central role in CPU operations, providing high-speed data storage and processing capabilities. Shift registers are essential for data serialization, deserialization, and parallel-to-serial or serial-to-parallel conversions, making them key components in communication systems and data processing circuits.

Registers are foundational to digital electronics, facilitating the seamless operation and optimization of modern digital systems across a wide range of applications. Understanding their structure, types, operations, and applications is paramount for professionals in the field of electronics and computer engineering.

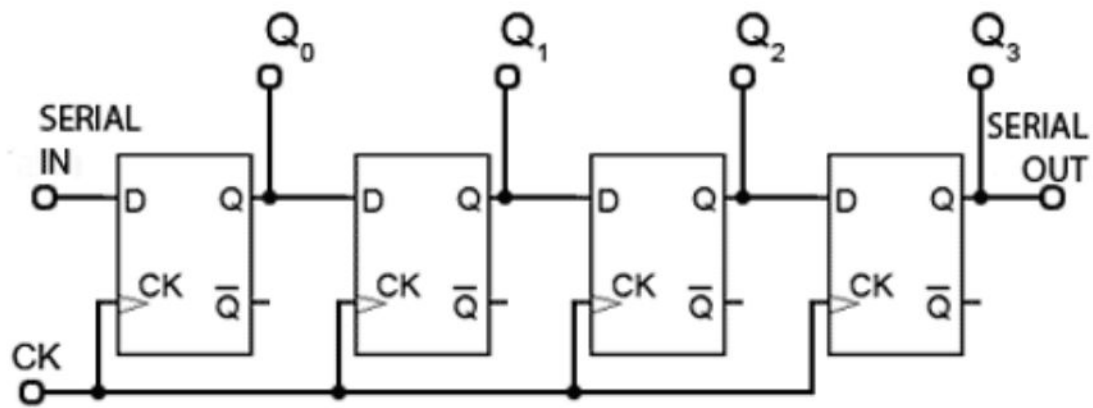


Fig 3.2: Register

We are using Shift register while implementing the architecture where a shift register is a digital electronic circuit or a collection of flip-flops arranged in a linear or circular manner, allowing the sequential shifting of binary data (bits) into and out of the register. This shifting of bits can occur either to the left (left shift) or to the right (right shift), based on the applied clock pulses or control signals. Each flip-flop in the shift register holds a single bit of data, and the movement of bits from one flip-flop to another is controlled and synchronized.

Shift registers are versatile components used for various applications, including data storage, serial-to-parallel or parallel-to-serial data conversion, data delay, arithmetic operations, and data manipulation. They are integral in serial communication, image processing, data compression, and numerous other digital systems where controlled and efficient movement of data is essential.

Shift registers come in different types, including Serial-In Serial-Out (SISO), Serial-In Parallel-Out (SIPO), Parallel-In Serial-Out (PISO), and Parallel-In Parallel-Out (PIPO), each with specific applications and functionalities. The choice of the type of shift register depends on the requirements of the application at hand. Overall, shift registers play a fundamental role in digital circuits, aiding in efficient data handling, processing, and transfer, making them a crucial component in modern digital electronics.

3.3 FLIP FLOPS

Flip-flops are fundamental building blocks in digital electronics, serving as bistable multivibrators capable of storing binary information. They are the foundation of memory elements and sequential logic circuits, pivotal for digital computation and data processing. This introduction provides an overview of flip-flops, highlighting their importance, fundamental characteristics, and key types.

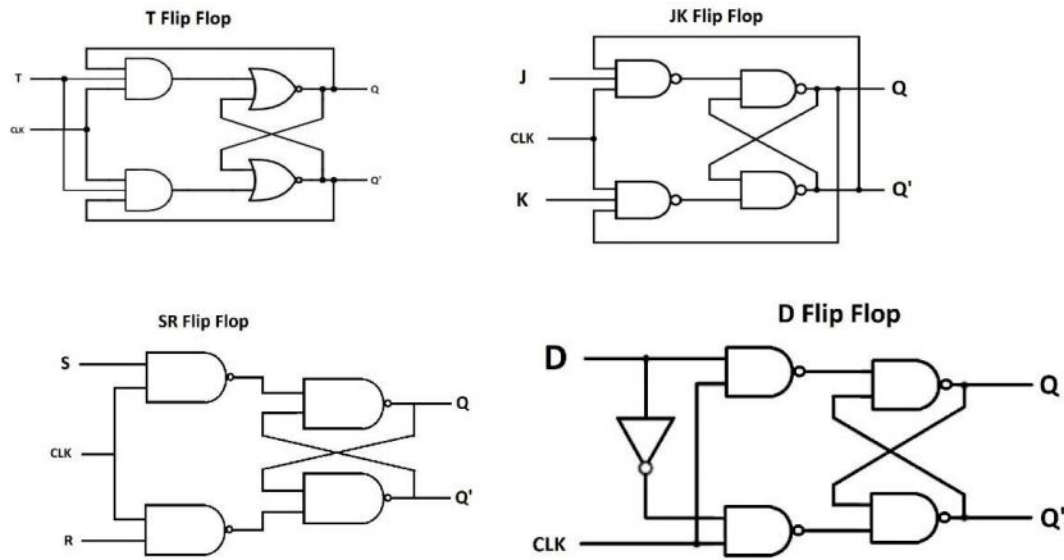


Fig 3.3: Types of Flip flops

3.3.1 The Key Characteristics and Functions:

- **Bi-stability:** A flip-flop has two stable states, and it remains in one of these states until an external input or clock signal prompts a transition.
- **Clock-Triggered Operation:** Many flip-flops are triggered by a clock signal. Changes in input data are typically captured and stored at the rising or falling edge of the clock signal.
- **Data Storage:** Flip-flops are used to store a binary digit (0 or 1) and are a fundamental building block for memory cells in digital circuits.
- **Synchronization:** Flip-flops help synchronize data within digital systems, ensuring that changes in data occur at specific clock cycles.
- **Sequential Logic:** Flip-flops play a critical role in sequential logic circuits, allowing the creation of memory elements and the building of more complex circuits such as registers and shift registers.

3.3.2 Types of Flip-Flops:

- SR Flip-Flop (Set-Reset Flip-Flop): Allows for setting, resetting, or holding the current state. It has two inputs, Set (S) and Reset (R), and can be used to store one bit of data.
- D Flip-Flop (Data or Delay Flip-Flop): Stores data on the clock signal's rising or falling edge, making it a popular choice for storage applications. It has a Data (D) input and a clock input.
- JK Flip-Flop: Combines the features of SR and D flip-flops, providing a versatile solution for storing binary information.
- T Flip-Flop (Toggle Flip-Flop): Toggles its output state (0 to 1 or 1 to 0) based on the clock signal. It has a toggle (T) input and a clock input.

3.4 Parallel Pipelined Architecture

The transposition of an $NR \times NC$ matrix is achieved through well-defined steps involving permutations of matrix values. Two circuit designs, referenced as discussed for swapping data points within a matrix. The proposed Algorithm1 is particularly suited for P-parallel architecture implementations. The architecture is composed of two types of processing elements (PEs), PEI and PEII, with each type fulfilling specific functions within the transposition process.

Control signals, determined by counters C0, C1, and C2, govern the stages and steps of the transposition process. The control strategy utilizes these counters to generate control signals for the various stages, aligning with the steps in Algorithm1. This enables an efficient transposition process, as illustrated in the timing diagram for a 3-parallel architecture transposing a 9×6 matrix. The latency and memory considerations are based on the matrix dimensions ($NR \times NC$). The number of stages in each step (A, B, C) is determined by the matrix dimensions, influencing the overall latency and memory requirements.

The ever-growing demand for high-performance and low-power computing in Very-Large-Scale Integration (VLSI) circuits has led to the exploration of innovative architectures. One such approach, garnering significant attention, is the parallel pipelined architecture. This powerful technique combines the strengths of both parallelism and pipelining to achieve remarkable efficiency gains.

Parallel processing, the cornerstone of this architecture, leverages multiple processing units working concurrently on a task. Imagine an assembly line – each unit tackles a specific step, significantly accelerating the overall process compared to a single unit handling everything sequentially. Similarly, parallel pipelined architectures divide a complex computational task into smaller, independent subtasks. These subtasks are then executed in a dedicated processing unit within the parallel architecture.

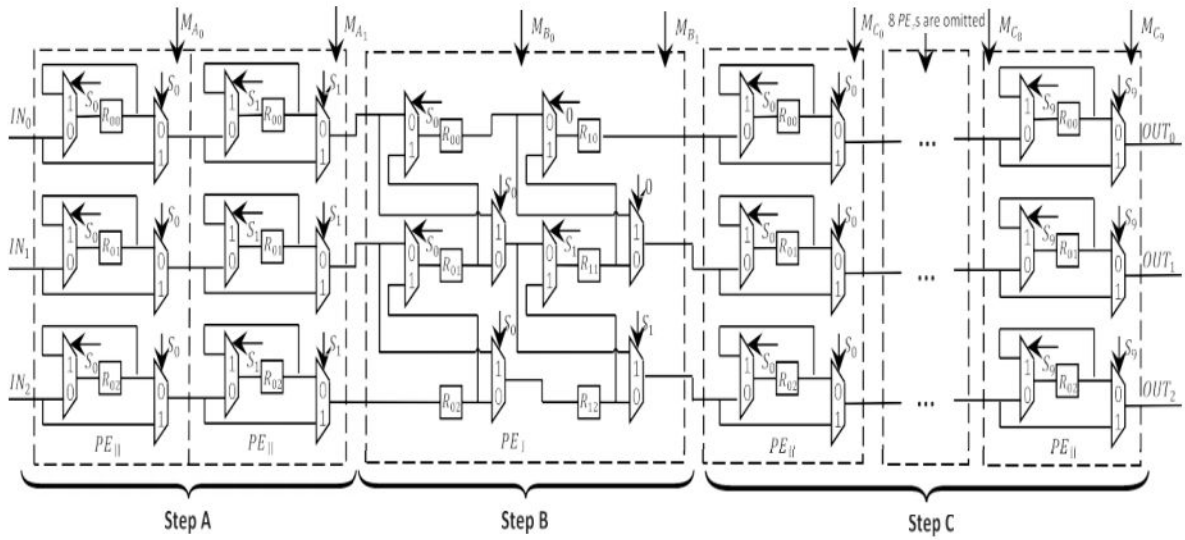


Fig 3.4: 3-parallel architecture diagram in Algorithm 1

Pipelining, the other pillar of this approach, further optimizes performance. It breaks down each subtask into even smaller, sequential stages. This creates a pipeline where data flows from one stage to the next, undergoing a specific transformation at each step. While one subtask completes a stage, another subtask can enter the same stage in the pipeline. This continuous flow maximizes resource utilization and significantly reduces the overall processing time.

The synergy between parallelism and pipelining creates a powerful combination. Parallel processing enables concurrent execution within each stage of the pipeline, maximizing throughput. Pipelining, in turn, ensures efficient utilization of processing units by keeping them constantly engaged. This collaborative approach leads to significant performance improvements over traditional sequential architectures.

However, it is crucial to acknowledge that parallel pipelined architectures are not without challenges. While they excel at increasing throughput, the constant activity within processing units can result in high power consumption.

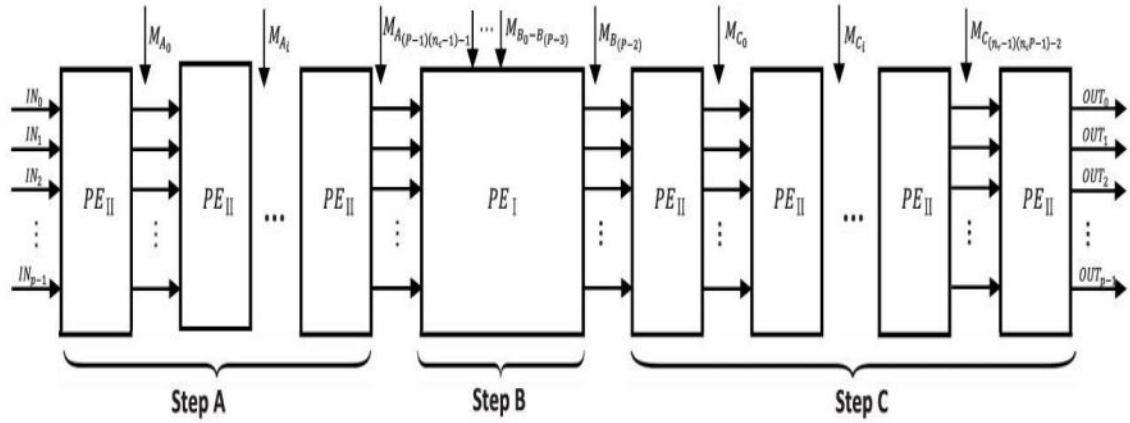


Fig 3.5: 3-parallel pipelined architecture block diagram

CHAPTER 4

PROPOSED SYSTEM

To improve efficiency, existing system as Parallel Pipeline architecture will be analyzed using 130nm, 32nm CMOS Technology. Further, Parallel Pipeline architecture will be modified and analyzed using 130nm, 32nm CMOS Technology. In terms of power, speed, and power delay product. Also, the proposed system will be analyzed and compared. This outperforms an efficient Parallel Pipeline architecture for high speed and low power applications.

4.1 D FLIP FLOP

In the realm of low-power digital design, D flip-flops (D-FFs) play a critical role. These fundamental building blocks are essential for storing data and synchronizing signals. However, with the ever-increasing demand for energy-efficient portable devices, minimizing D-FF power consumption has become a top priority.

Traditional D-FF designs can be significant contributors to a chip's overall power dissipation. This is due to two main factors: dynamic power, consumed during switching activity, and static leakage power, present even in idle states. To combat these issues, several design techniques have emerged for low-power D-FFs.

The ever-increasing demand for portable and battery-powered devices in the VLSI (Very-Large-Scale Integration) domain necessitates a continuous focus on reducing power consumption. D flip-flops (DFFs) are fundamental building blocks in digital circuits, but their constant operation contributes to overall power dissipation.

Traditional DFFs continuously consume power, even when the data they hold remains unchanged. In the context of DFFs, this translates to selectively turning off transistors when the data input (D) and clock (CLK) signals are stable, essentially putting the flip-flop into a low-power sleep mode.

Several sleep transistor configurations can be employed to achieve this functionality. A common approach involves introducing a sleep transistor in series with the power supply path. When activated by a sleep control signal, this transistor cuts off the power supply to the core DFF circuitry, significantly reducing leakage

current. Additionally, techniques like reverse body biasing can further enhance power savings by adjusting the threshold voltage of transistors, making them less susceptible to leakage in sleep mode.

The implementation of sleep techniques requires careful consideration to ensure proper functionality and minimal performance degradation. Activation and deactivation of sleep mode introduce additional delays, which need to be minimized for optimal performance. Techniques like gated clocks or sleep permission signals can be employed to ensure the DFF wakes up only when necessary, mitigating these delays.

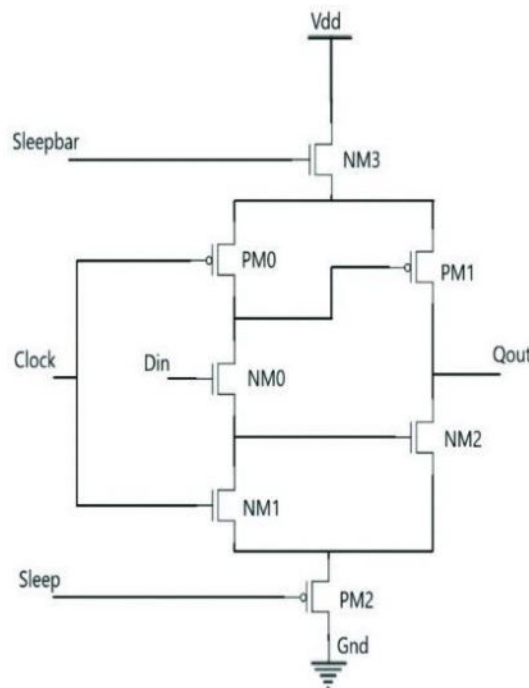


Fig: 4.1: Schematic of D-FF using Sleep Transistor Technique

One approach focuses on reducing dynamic power. This can be achieved through techniques like clock gating, which disables the clock signal when unnecessary, and data-dependent precharging, where precharging circuits only activate when data changes. Additionally, optimizing transistor sizing and logic styles can minimize internal switching activity, further lowering dynamic power consumption. Another strategy tackles static leakage power. By lowering the supply voltage or operating the D-FF in the sub-threshold region (where current flow is significantly reduced), leakage can be curtailed. However, these methods often come at the cost of increased delay or reduced noise immunity.

The ideal low-power D-FF design depends on the specific application's requirements. Factors like operating frequency, data activity, and leakage tolerance all play a role in selecting the most suitable approach. Cutting-edge research explores even more advanced techniques, such as adaptive voltage scaling and multi-threshold CMOS (MTCMOS) for further power optimization.

In conclusion, low-power D-FF design is a crucial area of research for battery-powered devices. By employing various techniques to minimize dynamic and static power consumption, engineers can create more energy-efficient circuits, paving the way for a future of longer battery life and sustainable electronics. The trade-off between power savings and performance is crucial in sleep transistor design. While aggressive sleep mode implementation can achieve significant power reduction, it may introduce larger wake-up delays. Careful optimization and design choices are required to strike a balance between these competing factors. This research explores various sleep transistor configurations and analyzes their impact on power consumption, wake-up delay, and overall DFF performance in VLSI applications.

4.2 LEAKAGE POWER REDUCTION TECHNIQUE

As Very-Large-Scale Integration (VLSI) circuits shrink and pack more transistors, leakage power consumption becomes a critical concern. Leakage current, the current that flows even when a transistor is supposedly off, wastes energy and generates unwanted heat.

The core principle of sleep techniques lies in power gating. A sleep transistor, typically a high threshold voltage (V_t) device, is inserted between the power rail (V_{dd} or ground) and the logic circuit. When the circuit enters sleep mode, the sleep transistor turns off, isolating the circuit from the power supply and slashing leakage current. Popular sleep transistor implementations include the sleepy keeper approach.

Power consumption has become a critical concern in the realm of Very-Large-Scale Integration (VLSI) design. As transistor scaling continues, leakage currents pose a significant challenge, especially for battery-powered devices and high-performance applications. To combat this issue, sleep techniques have emerged as a powerful tool for dynamic power management, offering substantial reductions in energy consumption without compromising functionality.

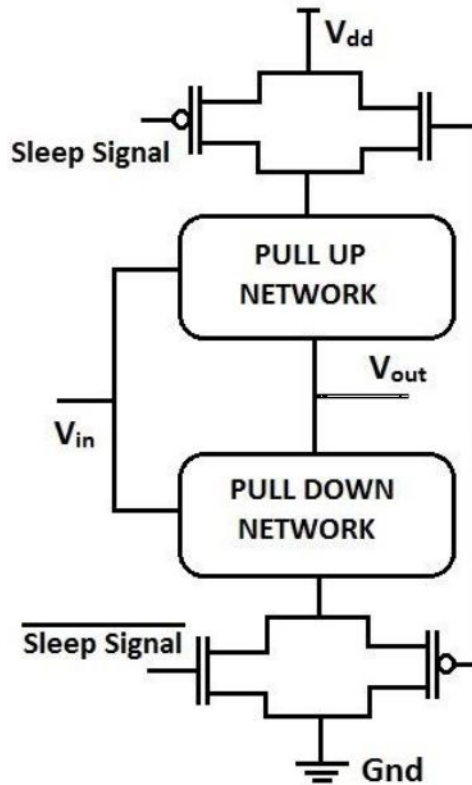


Fig: 4.2: Schematic of Sleep Technique

The fundamental principle behind sleep techniques lies in strategically powering down inactive portions of a circuit. In the context of digital VLSI circuits, this often translates to selectively turning off transistors in functional blocks during idle periods. This approach significantly reduces leakage currents, which are the primary culprits for static power dissipation.

There are various sleep transistor configurations that can be implemented to achieve this power gating functionality. A common strategy involves inserting a sleep transistor, typically a high threshold voltage (V_t) transistor, in series with the power supply path. When activated by a sleep control signal, this transistor effectively cuts off the power supply to the core logic, minimizing leakage current.

However, implementing sleep techniques requires careful consideration. Activation and deactivation of sleep mode introduce additional delays due to the charging and discharging of circuit nodes. These delays, known as wake-up delays, can impact performance if not managed effectively. Techniques like gated clocks or sleep permission signals can be employed to ensure that sleep mode is activated only when necessary and that the circuit wakes up only when required data arrives.

The choice of sleep transistor configuration and the control strategy for sleep mode activation significantly impact the trade-off between power savings and performance. Aggressive sleep mode implementation can achieve substantial power reduction, but it may come at the cost of larger wake-up delays. Therefore, VLSI designers must carefully optimize sleep transistor design and control mechanisms to strike a balance between minimizing power consumption and maintaining acceptable performance metrics.

In conclusion, sleep techniques offer a compelling solution for power management in VLSI design. By strategically powering down inactive circuits, these techniques enable significant reductions in leakage currents and overall power dissipation. As research continues to explore innovative sleep transistor configurations and control strategies, sleep techniques are poised to play an increasingly vital role in achieving energy-efficient VLSI circuits for the future.

Sleep techniques offer significant advantages. The most prominent benefit is the dramatic reduction in leakage power, leading to improved battery life in portable devices and lower power consumption in general. Additionally, reduced heat generation translates to better thermal performance, which can enhance circuit reliability and lifespan. Furthermore, the sleep/wake up cycles introduce a slight performance penalty due to the time it takes to turn the power supply on and off. Careful design considerations are necessary to balance the power savings with the area and performance impact.

In conclusion, sleep techniques are a powerful tool in the VLSI designer's arsenal for combating leakage power. While there are trade-offs in terms of area and performance, the benefits often outweigh the drawbacks, making sleep techniques a cornerstone of low-power VLSI design.

4.3 STANDARDS

- IEEE standards for Low-Power Design: There are standards focusing on design practices and methodologies to minimize power consumption. Examples include IEEE 1801 (Unified Power Format)
- Clarity and Accuracy: Emphasize the low-power benefit instead of just mentioning the "sleep technique."
- Accurate representation: Ensure the title accurately reflects the project's goals and methodology.
- Conciseness and Impact: Aim for a title that is clear and informative, but concise enough to be easily understood and remembered.
- Intrigue the audience: A well-crafted title should pique the reader's interest and encourage them to learn more about your project.

4.4 CONSTRAINTS

- Time Constraint: The project must be completed within the specified timeframe, adhering to the allocated schedule for research, analysis, design, implementation, testing, and documentation phases.
- Resource Limitations: Due consideration should be given to the availability of resources such as hardware, software tools, and computing resources. The project should be designed to operate within these constraints.
- Budget Constraints: The project should operate within the allocated budget for procurement of hardware, software licenses, research materials, and any other necessary resources.
- Technical Compatibility: The proposed architecture should be compatible with existing hardware and software systems commonly used in low power applications. Compatibility issues should be identified and addressed effectively.
- Power Consumption Limit: The implemented architecture should demonstrate significant reductions in power consumption compared to conventional approaches. The project should define specific power consumption targets and aim to achieve them.

- Performance Targets: While optimizing for low power consumption, the project should not compromise on performance metrics such as throughput, latency, and scalability. Performance targets should be defined and met accordingly.

4.5 TRADE-OFFS:

- Power vs. Performance: Balancing power reduction with maintaining performance levels.
- Complexity vs. Efficiency: Managing complexity for optimal power efficiency.
- Resource Use vs. Savings: Utilizing resources for power savings without increasing energy consumption.
- Latency vs. Efficiency: Minimizing latency while implementing power-saving techniques.
- Design Flexibility vs. Efficiency: Choosing the right balance between design flexibility and power efficiency.
- Verification Complexity vs. Savings: Managing verification complexity while ensuring power savings.
- Compatibility vs. Innovation: Balancing compatibility with the exploration of innovative approaches.
- Cost vs. Savings: Evaluating the cost of implementing power-saving features against potential savings.
- Scalability vs. Efficiency: Ensuring scalability without compromising power efficiency.
- Reliability vs. Savings: Maintaining reliability while achieving power savings.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 EXSISTING SYSTEM SIMULATED OUTPUTS

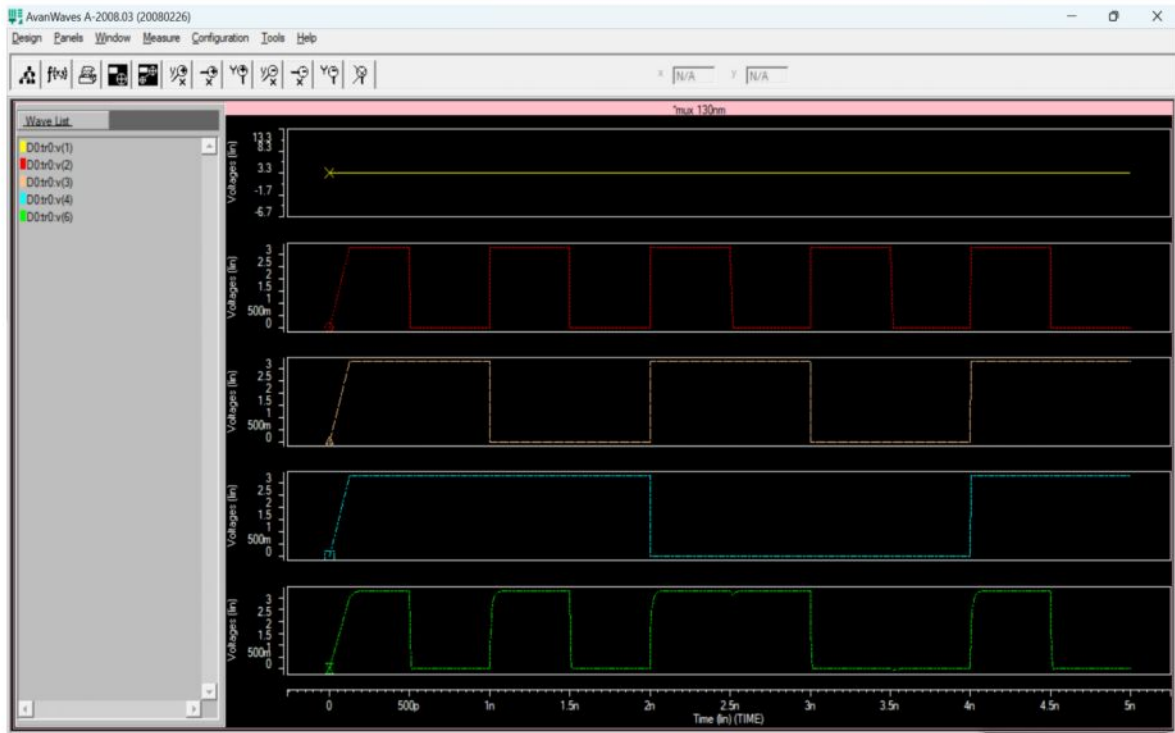


Fig: 5.1: 2x1 Multiplexer

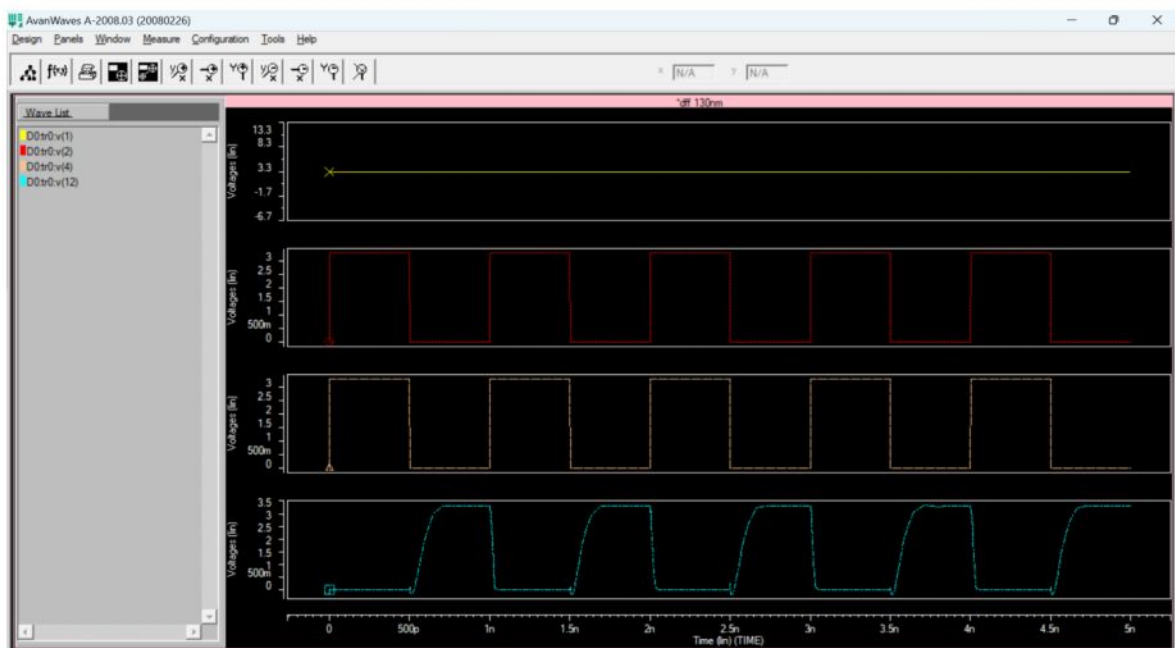


Fig: 5.2:D flip flop without using sleep technique

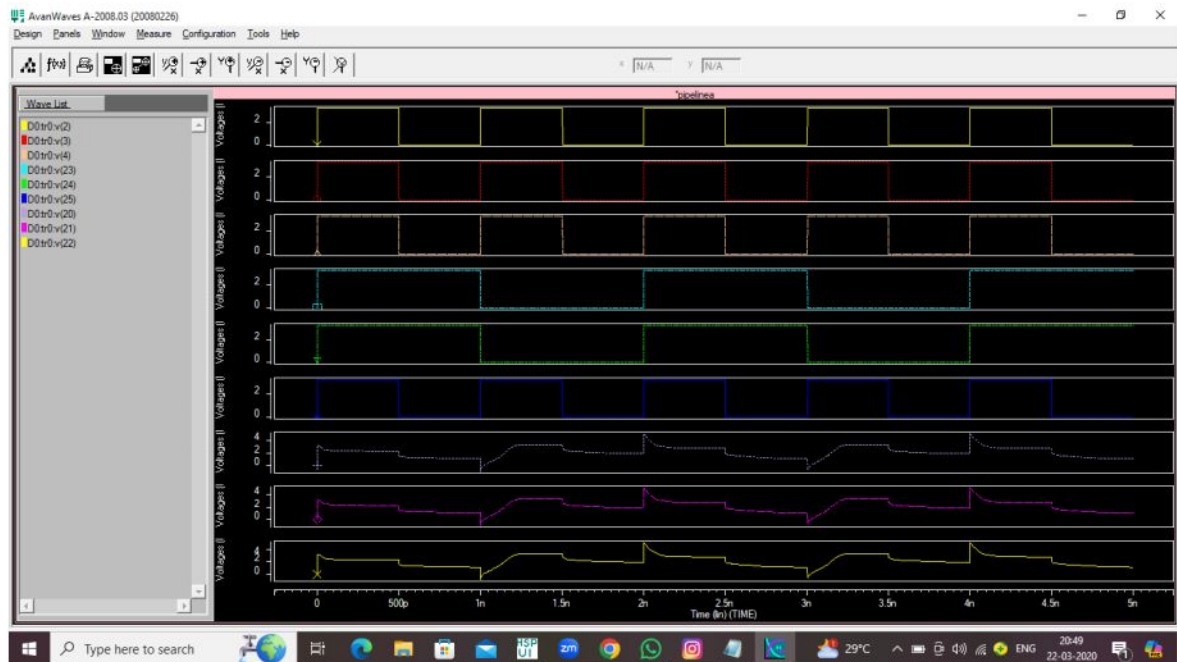


Fig: 5.3: Step A without using sleep Technique

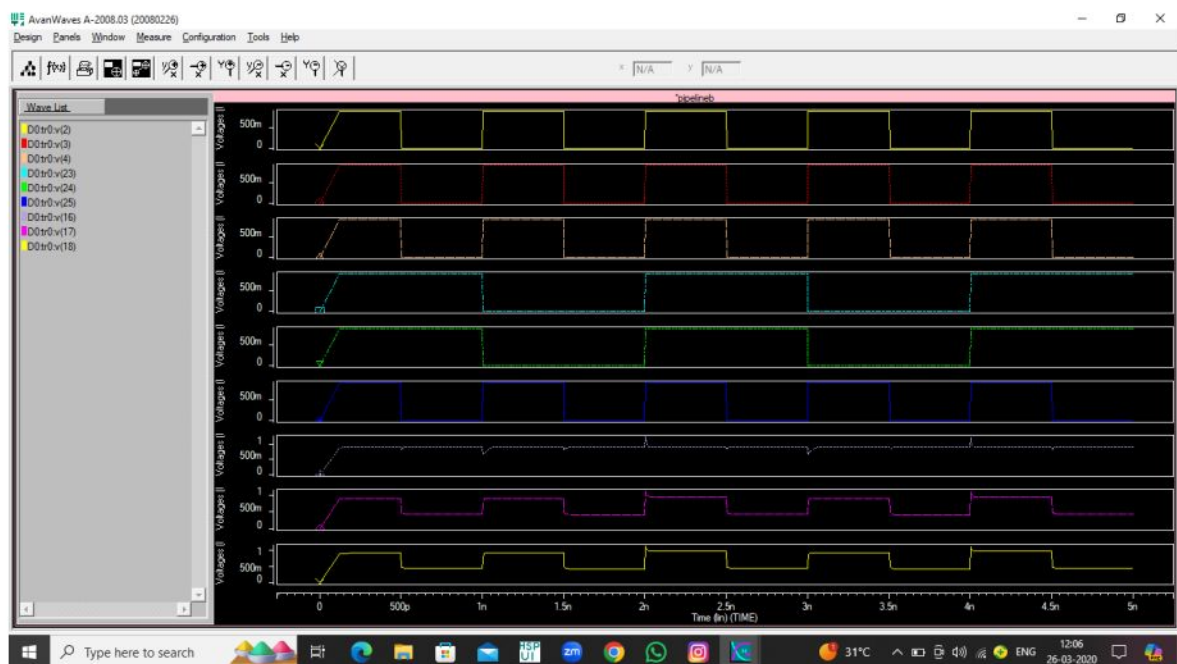


Fig: 5.4: Step B without using sleep Technique

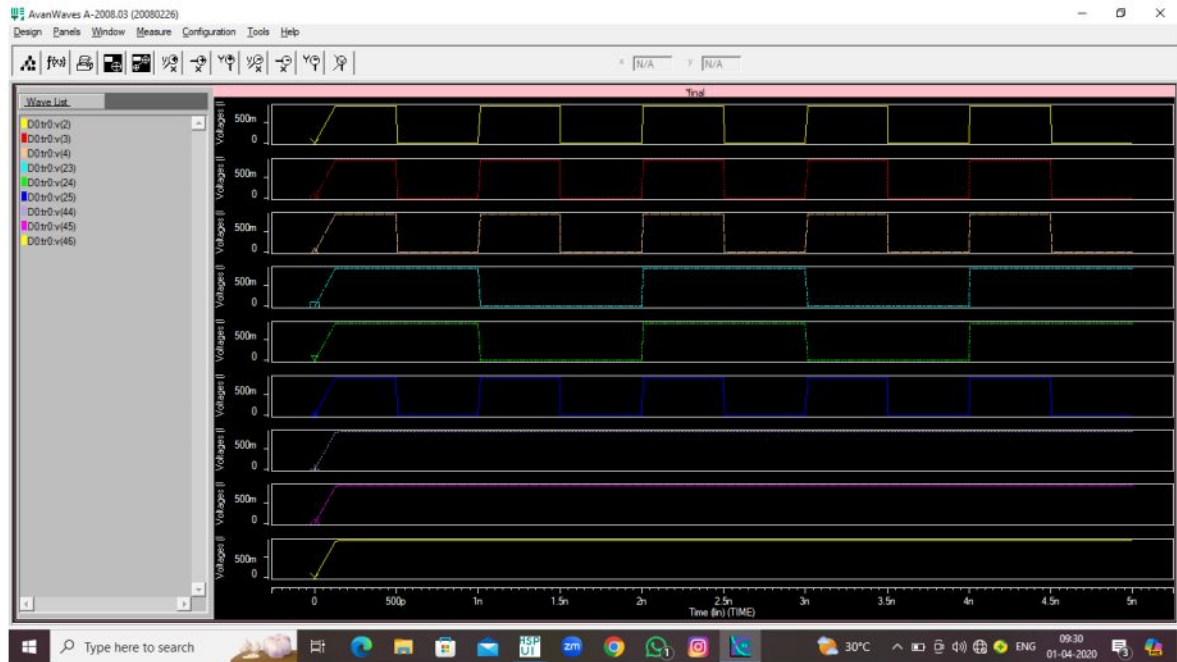


Fig: 5.5: Final output without using sleep Technique

Figure 5.5 depicts the final output waveform without using sleep technique for which the inputs are 2, 3, 4, and signal lines S0, S1 are 23, 24 and clock is 25 and the output obtained is 44, 45 and 46.

5.2 PROPOSED SYSTEM SIMULATED OUTPUTS

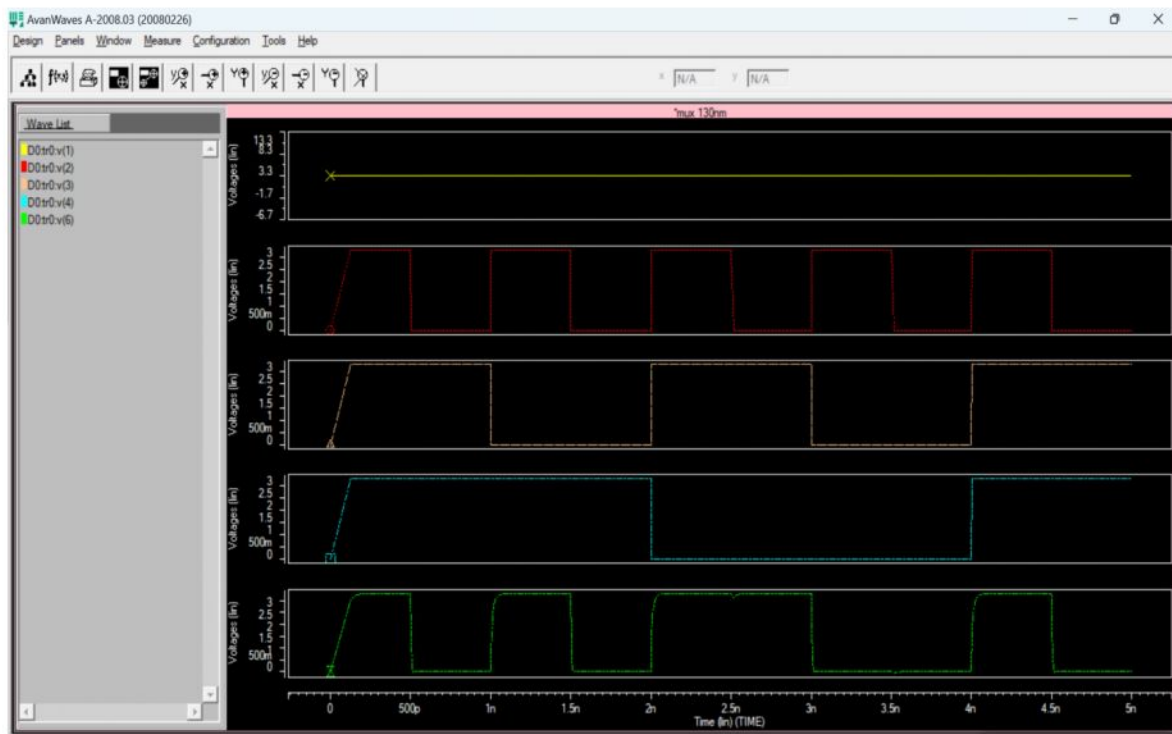


Fig: 5.6: 2x1 Multiplexer

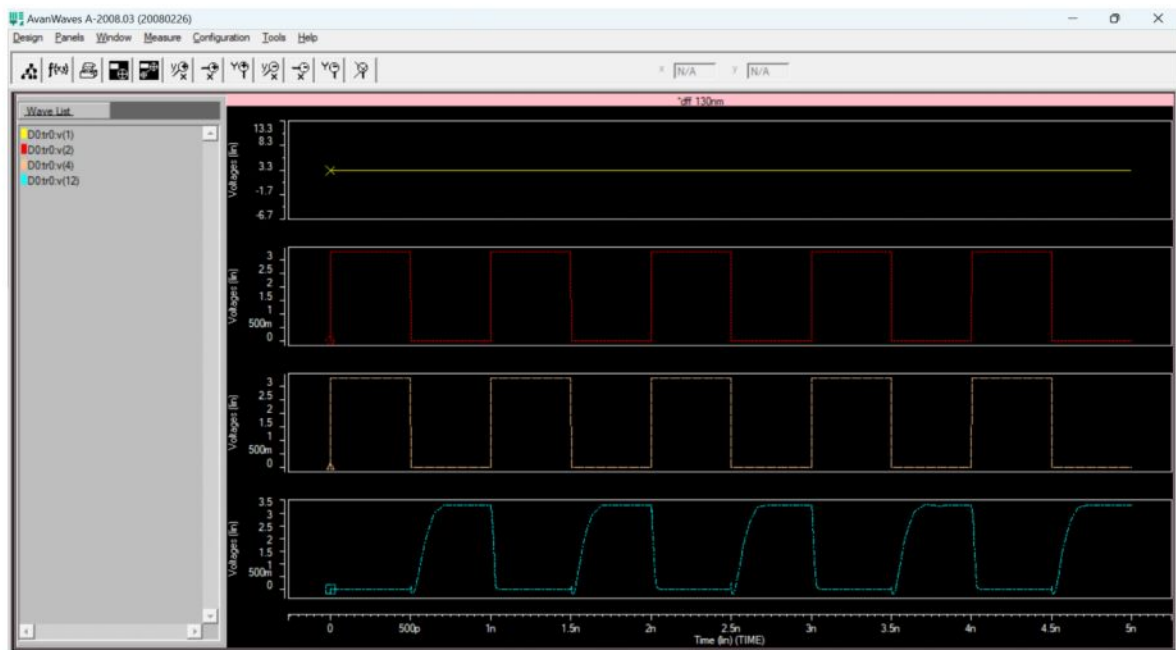


Fig: 5.7:D flip flop

Figure 5.7 depicts the output waveform of modified D Flip Flop for which the inputs are 4,5 and clock is 2, and clock bar is 3, and the output obtained is 6, 7, 13.

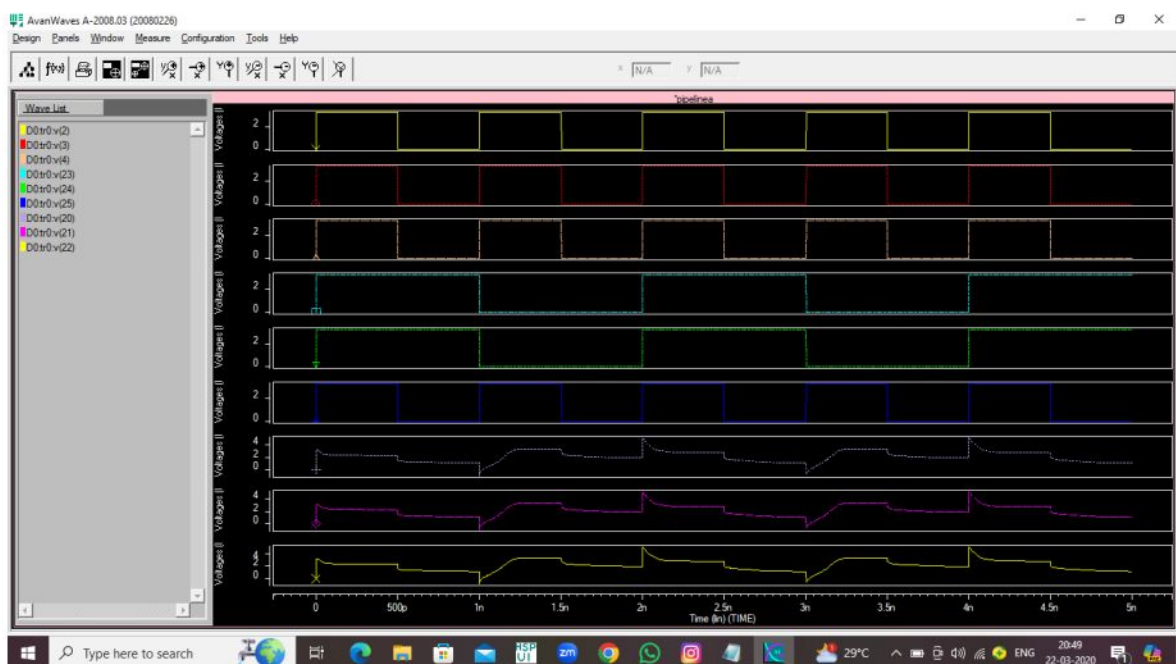


Fig: 5.8: Step A with using sleep Technique

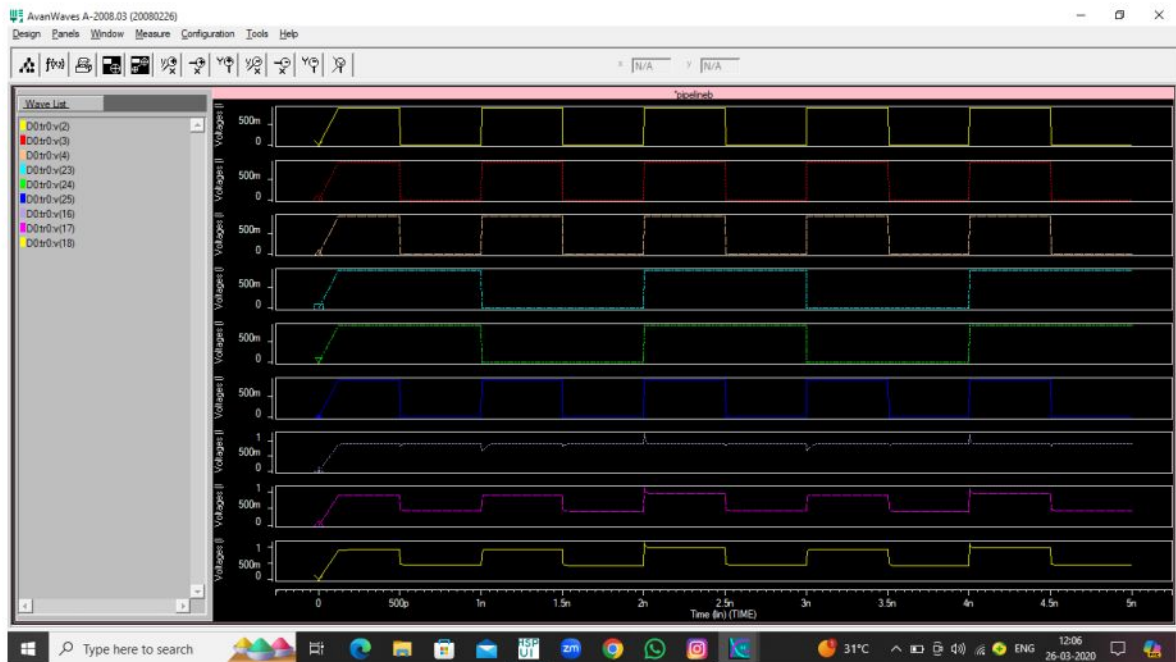


Fig 5.9: Step B with using sleep Technique



Fig 5.10: Final output with using sleep Technique

Figure 5.10 depicts the final output waveform using sleep technique for which the inputs are 2, 3, 4, and signal lines S0, S1 are 23, 24 and clock is 25 and the output obtained is 44, 45 and 46.

5.3 TABULATION

	130nm			32nm		
TITLE	POWER	DELAY	POWER DELAY PRODUCT	POWER	DELAY	POWER DELAY PRODUCT
MULTIPLEXER	1.49×10^{-5}	5.5213p	8.22×10^{-17}	1.13×10^{-6}	7.013p	7.92×10^{-18}
D FLIP FLOP	2.58×10^{-5}	498.5p	1286.13×10^{-17}	1.03×10^{-8}	735.83p	757.90×10^{-20}
STEP A	5.56×10^{-4}	496.7p	2761.65×10^{-16}	6.28×10^{-6}	48.28p	303.19×10^{-18}
STEP B	4.44×10^{-4}	499.8p	2219.11×10^{-16}	7.82×10^{-6}	26.52p	207.38×10^{-18}
FINAL	1.88×10^{-3}	437.76p	$.822.98 \times 10^{-15}$	2.37×10^{-6}	223.76p	530.31×10^{-18}

Table5.1: Analysis of Power, delay & Power delay product without using “Sleep Technique”

	130nm SLEEP			32nm SLEEP		
TITLE	POWER	DELAY	POWER DELAY PRODUCT	POWER	DELAY	POWER DELAY PRODUCT
MULTIPLEXER	1.05×10^{-5}	2.12p	2.22×10^{-17}	0.76×10^{-16}	5.12p	3.89×10^{-28}
D FLIP FLOP	1.48×10^{-5}	3.797p	5.61×10^{-17}	0.93×10^{-8}	73.07p	67.95×10^{-20}
STEP A	1.23×10^{-4}	151.55p	186.40×10^{-16}	5.35×10^{-6}	18.93p	101.27×10^{-18}
STEP B	2.80×10^{-5}	9.0325p	25.29×10^{-17}	3.83×10^{-6}	1.71p	6.54×10^{-18}
FINAL	1.12×10^{-4}	1.6002p	1.79×10^{-16}	2.13×10^{-6}	1.63p	3.47×10^{-18}

Table5.2: Analysis of Power, delay & Power delay product with using “Sleep Technique

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

In conclusion, this work presented a comprehensive analysis of parallel pipelined architectures for achieving low-power operation in VLSI applications. The investigation revealed the inherent efficiency gains from parallel processing and pipelining, demonstrating their capability to deliver high throughput while minimizing power consumption. Furthermore, the research explored the integration of sleep techniques as a strategic method for further power optimization. By dynamically powering down inactive sections of the pipeline during idle periods, sleep techniques offer a compelling solution for significantly reducing energy usage without sacrificing performance. This combined approach represents a significant advancement in low-power VLSI design, particularly for battery-powered devices and applications demanding high performance with minimal energy consumption. The successful implementation of parallel pipelined architectures with sleep techniques paves the way for a new generation of energy-efficient VLSI circuits, enabling more sustainable and portable computing solutions.

REFERENCES

- [1] Y. Wang, Z. Ma, and F. Yu, "Pipelined algorithm and modular architecture for matrix transposition," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 4, pp. 652–656, Apr. 2019.
- [2] M. Garrido, J. Grajal and O. Gustafsson, "Optimum circuits for bit-dimension permutations", *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 5, pp. 1148-1160, May 2019.
- [3] Bo Zhang, Zhenguo Ma, and Wei Luo, "Parallel Pipelined Architecture and Algorithm for Matrix Transposition Using Registers", *IEEE Transactions on Circuits and Systems II: Express Briefs* Vol. 69, Issue: 3, March 2022.
- [4] Ch. Meghana Reddy, Souvik Dhara, B. Srikanth, P. Ramakrishna, "Designing a D-Flipflop Using Novel Sleep Transistor Technique", 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 08-10 July 2022.
- [5] Asif Jahangir Chowdhury, Md. Shahriar Rizwan, Shahriar Jalal Nibir, Md. Rifat Alam Siddique, "A new leakage reduction method for ultralow power VLSI design for portable devices", 2nd International Conference on Power, Control and Embedded Systems, 2012.
- [6] J. Park, "Sleepy Stack: A New Approach to Low Power VLSI and Memory," Ph.D. Dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2005.
- [7] Se Hun Kim, V. J. Mooney, "Sleepy Keeper: A New Approach to Low-leakage Power VLSI Design," in *Proc. 2006 IFIP International Conference on Very Large-Scale Integration*, pp 367-372, October, 2006.
- [8] M. Aguirre-Hernandez; M. Linares-Aranda, "A Clock-Gated Pulse-Triggered D Flip-Flop for Low-Power High-Performance VLSI Synchronous Systems", *International Caribbean Conference on Devices, Circuits and Systems*, 2006.
- [9] K. G. Sharma, Tripti Sharma, B. P. Singh, "Modified SET D-Flip Flop Design for Low-Power VLSI Applications", *International Conference on Devices and Communications (ICDeCom)*, 2011.

- [10] Z. Chen, M. Johnson, L. Wei and K. Roy, "Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks", International Symposium on Low Power Electronics and Design, pp. 239-244, August 1998.
- [11] Gagandeep Singh; Gurmohan Singh; Vemu Sulochna, "High performance low power dual edge triggered static D flip-flop ", Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 04-06 July 2013.
- [12] H. Yu, and J. Cho, "Low-Power Design and Architecture", IEEE Potentials, 18-22, pp. 114-119, September 2001.
- [13] Pritam Bhattacharjee, Alak Majumder, Tushar Dhabal Das, "A 90 nm leakage control transistor-based clock gating for low power flip flop applications", IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), 16-19 October 2016.
- [14] N. Hanchate; N. Ranganathan, "LECTOR: a technique for leakage reduction in CMOS circuits", IEEE Transactions on Very Large-Scale Integration (VLSI) Systems (Volume: 12, Issue: 2, February 2004), pp. 196-205.
- [15] Osama Bondonq, Khaldoun Abugharbieh, Abdullah Hasan. "A D-Type Flip-Flop with Enhanced Timing Using Low Supply Voltage ". IEEE Canadian Conference on Electrical and Computer Engineering (CCECE).30 August 2020 - 02 September 2020.
- [16] Dhandapani Vaithyanathan; Britto Pari James; Karuthapandian Mariammal, "A High-Speed Computational Pipeline Single MAC-Based VLSI Architecture for Real-Time Signal and Image Processing", Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 11-12 January 2024.
- [17] R Anjana, Ajay Kumar Somkuwar, "Analysis of sub threshold leakage reduction techniques in deep submicron regime for CMOS VLSI circuits". International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT). 07-09 January 2013.

- [18] K. N. Bhargav, A. Suresh, Gaurav Saini, "Stacked keeper with body bias, A new approach to reduce leakage power for low power VLSI design", IEEE International Conference on Advanced Communications, Control and Computing Technologies, 08-10 May 2014.
- [19] Madhusudan Maiti, Anupama Paul, Suraj Kumar Saw, Alak Majumder, "A Dynamic Current Mode D-Flipflop for High-Speed Application", 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), 29-31 August 2019.
- [20] A. Kapoor, Y. Hu and R. Bashirullah, "A current-density centric logical effort delay and power model for high-speed CML gates", IEEE Trans. Circuits Syst. I Reg. Papers, vol. 60, no. 10, pp. 2618-2630, Oct. 2013.