

Performance Evaluation of an Efficient Fine-Grained Sleep Technique for Parallel Architectures

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

by

MOLUGURI AKASH (39130293)

MALLAMPALLI PHANICHANDER (39130271)



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

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

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 Moluguri Akash (39130293) and Mallampalli Phanichander (39130271) who carried out the project entitled **“Performance Evaluation of an Efficient Fine-Grained Sleep Technique for Parallel Architectures”** under our supervision from November 2022 to April 2023.

Internal Guide

Dr. MATHAN N. M.Tech, Ph.D.,

Associate Professor

Head of the Department

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

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, Moluguri Akash (39130293), Mallampalli PhaniChander here by declare that the Project Report entitled “**Performance Evaluation of an Efficient Fine-Grained Sleep Technique for Parallel Architectures**” done by us under the guidance of Dr. MATHAN N., M.Tech., Ph.D., Assistant Professor is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 15.04.2023

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

1. 

2. 

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our 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.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.MATHAN N., M.Tech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our 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

This project presents a design approach for improving energy-efficiency and throughput of parallel architectures in near- and subthreshold voltage circuits. The focus is to suppress leakage energy dissipation of the idle portions of circuits during active modes, which can allow us to wholly transform the throughput improvement from parallel architectures into energy savings via deep voltage scaling. We begin by investigating the efficacy of parallel and pipeline architectures in the near- and subthreshold circuits. The investigation reveals that active energy dissipation largely undermines the ability of deep voltage scaling to transform excessive throughput into energy savings. Techniques, such as power-gating switches (PGSs), can mitigate active-leakage power dissipation; however, the overhead for entering and exiting sleep modes can offset the energy savings provided by sleep mode, particularly if sleep time is fine-grained for suppressing active leakage. Therefore, in this paper, we propose a PGS design technique, inspired by the so-called zigzag super cutoff CMOS, in order to optimize the overheads of mode transitions of PGS in near- and subthreshold circuits. The proposed technique enables to have circuits in sleep mode for as short as a single clock cycle with a negligible amount of energy and delay overheads. Comparison in performance will be analyzed with existing and proposed systems in terms of power, delay, and power delay product which will be used for low power applications.

CONTENTS

Chapter no	TITLE	Page no
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF TABLES	x
1	INTRODUCTION	1
	1.1 VLSI Design flow	2
	1.2 Power gating system (PGS)	4
	1.3 ZSCMOS	8
	1.4 Multipliers	8
2	LITERATURE SURVEY	11
	2.1 Literature survey	11
	2.2 Literature summary	22
3	EXISTING SYSTEM	23
4	AIM AND SCOPE OF THE PRESENT INVESTIGATION	27
	4.1 Problem definition	
	4.2 Objective of the Project	27
	4.3 Parallelism and Pipelining in Near and sub-threshold circuits	28 28
	4.4 Mode transition overhead of PGS	29
5	IMPLEMENTATION DETAILS	30
	5.1 LCG Logic circuit	30
	5.2 Parallel Architecture with Temporarily fine-grained sleep Technique	31
	5.3 Software Used	32
	5.4 Net lists of gate	34
	5.5 Implementation of Logic Gates	34
6	RESULTS AND DISCUSSION	40
	6.1 Performance analysis	47
7	CONCLUSION	48
	REFERENCES	49

LIST OF FIGURES

FIGURE.NO	TITLE	PAGE NO:
1.1	Flow chart of VLSI Design	4
1.2	Power dissipation	6
1.3	Vlsi Soc design power gating	7
1.4	Footer Power gating systems	9
1.5	Zigzag super cut-off CMOS scheme	10
1.6	Software multiplier	11
3.1	Four PGS designs	24
3.2	A 16 bit two-way parallel multiplier	26
4.1	Proposed parallel multiplier	27
4.3.1	Three test architectures based on a 16-bit multiplier	29
5.21	Block Diagram of LCG	30
5.22	Circuit Diagram of LCG	31
5.31	50-stage inverter chain circuits with the ZSCCMOS.	34
6.1	The Simulation output of 4 Bit pipeline carry look ahead adder	40
6.2	The Simulation output of 4 Bit pipeline carry look ahead adder using sleep technique	40
6.3	The Simulation output of a Pipelined speculator	41
6.4	The Simulation output of a Pipelined speculator Using sleep technique	42
6.5	The Simulation output of a pipelined compensator	43
6.6	The Simulation output of a pipelined compensator Using sleep technique	44
6.7	The Simulation output of a pipelined in extract speculative Adder of existing system	44
6.8	The Simulation output of a pipelined in extract speculative Adder of Proposed system	45
6.9	The Simulation output of 4 Bit pipeline carry look ahead Adder output using stack technique	46

6.10	The Simulation output of a pipelined compensator	
	Using stack technique	46
6.11	The Simulation output of a pipelined speculator	
	Using stack technique	47

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
6.1.1	Delay analysis of Existing system	46
6.1.2	Delay analysis of Proposed system using sleep technique	46
6.1.3	Delay analysis of Proposed system using stack technique	47
6.1.4	Delay analysis of Proposed system using sleepy stack technique	47

CHAPTER - 1

INTRODUCTION

Technology scaling has enabled an incredible progress in integrated circuits for the last several decades. Guided by Moore's law and the Dennard scaling, transistors have become smaller, faster, and dissipated less power, enabling high performance and extremely integrated computing systems for a spectrum of digital computing applications. Nevertheless, in recent years, power dissipation has been changing the general mindset with respect to the design of digital circuits. As power and energy have come to be of greater significance, designers have had to conscientiously devise techniques that aim greater energy-efficiency while avoiding or mitigating performance degradation. The focus is to suppress leakage energy dissipation of the idle portions of circuits during active modes, which can allow us to wholly transform the throughput improvement from parallel architectures into energy savings via deep voltage scaling. We begin by investigating the efficacy of parallel and pipeline architectures in the near- and subthreshold circuits. In order to improve computing energy-efficiency beyond the conventional limit, i.e., VOPT, it is of great importance to reduce leakage energy dissipation during active modes. One of the potential solutions for this is to place idle parts of circuits into a low-leakage (sleep) mode. For example, although not targeting ultralow-voltage (ULV) circuits, Hu et al. and Tschanz et al. have proposed power-gating switch (PGS) for each function block of an execution stage of a pipelined microprocessor. By opportunistically having the blocks that perform no useful work in a sleep mode, we can reduce active leakage waste. While low-leakage sleep mode is a valid approach, it can cause nonnegligible energy and delay overheads to frequently enter and exit modes (i.e., mode transitions) for suppressing active-leakage dissipation. The use of PGS, as an example, can consume a significant amount of dynamic energy to charge and discharge parasitic capacitances during mode transitions. Furthermore, the delay required for transitioning from a sleep to an active mode can degrade throughput and complicate sleep control.

1.1 VLSI Design Flow

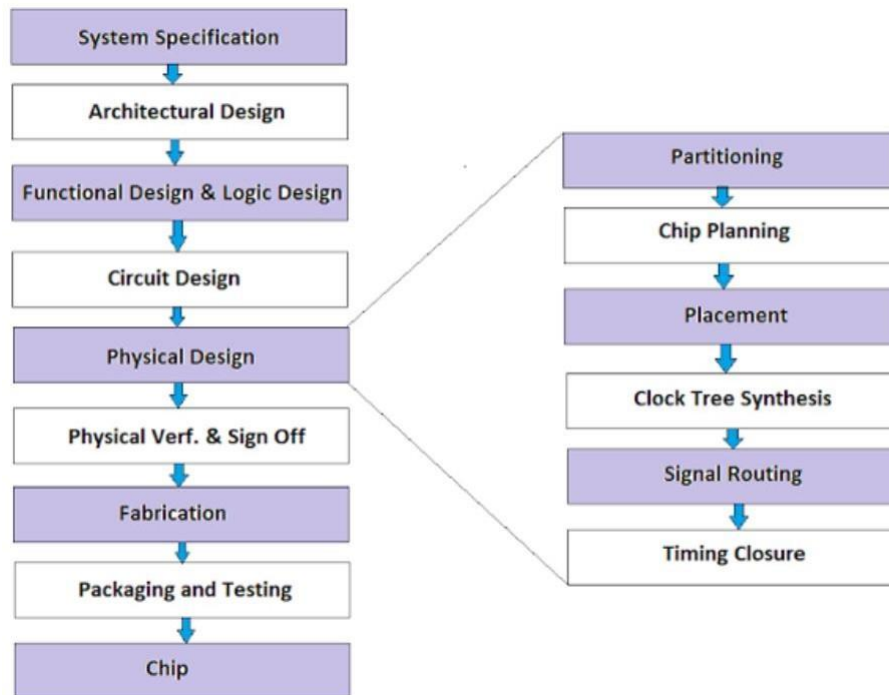


Fig 1.1: Flow chart of VLSI Design

System Specification is the first step of VLSI Design Flow is system specifications. System specification is a highlevel representation of the system. The factors to be considered in this process include performance, functionality and interface.

Architectural Design is the step where main work start with the help of system specification.Design engineer design the architecture according to system specification. Functional and Logic Design In this step functionality of design are identified.This specify the hardware implementation of system functionality. The outcome of functional design is usually a timing diagram.In logic design step, register allocation, logic and arithmetic operations of the design that represent the functional design are derived and tested this description is called RTL description. RTL stands for register transfer level. In this step, system specification is expressed in a hardware description language (HDL) such as Verilog and VHDL.RTL description is used for simulation to test the functionality with the help of EDA tools. Functional verification is performed to ensure the RTL design is done according to the specifications RTL code is converted to gate level netlist using synthesis tools. Netlist is a description of the circuit in terms of gates and connections between them.To verify whether the synthesis tool has correctly generated the gate-level netlist a verification should be done.

In Circuit Design this step circuit is designed based on the logic design. the Boolean expressions are converted into circuit representation by taking into consideration the power and speed requirement of original design. Circuit simulation is used to verify the correctness and timing of each component. Diagram consists circuit elements such as gates and transistors. Physical Design In this step the netlist is converted into physical geometric representation. Layout is representation of an IC in terms of planar geometric shapes which correspond to the patterns of metal oxide or semiconductor layers that make up the components of the Integrated circuit. Layout is designed by eda tool such as cadence virtuoso. Physical design is a very complex step therefore it is divided into sub steps such as floor planning, placement, clock tree synthesis, routing etc and timing analysis checks are formed in each and every step during physical design. Floor planning which is a process of placing the various blocks and the I/O pads across the chip area based on the design constraints. Placement of physical elements within each block and integration of analog blocks or external IP cores is performed. When all the elements are placed, a global and detailed routing is running to connect all the elements together. Output of layout is GDSII file which is given to the foundry to fabricate the chip. The layout should be done according to foundry design rules.

Physical Verification and Signoff In this step we perform physical verification checks such as Layout Vs schematic (LVS) and Design Rule check (DRC).DRC verifies whether the given layout satisfies the design rules provided by the fabrication team. DRC checks are nothing but physical checks of spacing rules between metals, minimum width rules, via rules etc.LVS is a major check in the physical verification stage. Layout is compared with the schematic for verifying whether their functionally match or not. If match, then the LVS reports clean.

Fabrication after physical verification step the design is ready for fabrication. Tape out is the final result of the design process for integrated circuits before they are sent for manufacturing. The tape-out is specifically the point at which the graphic for photo mask of the circuit is sent to the foundry. Fabrication process consists of several steps involving wafer growth, epitaxial growth masking, etching, doping, deposition, and diffusion of various materials on the wafer. During each step one mask is used.

Packaging and Testing Each of the wafers contains hundreds of chips. these chips are separated and packaged by a method called scribing and cleaving. The chips that are

failed in electrical test are discarded. Each chip is packaged and tested to ensure that it meets all the design specifications and functions properly.

1.2 POWER GATING SYSTEMS(PGS)

Power gating is yet another effective implementation employed in Low Power Designs. Unlike clock gating which saves the dynamic power, Power Gating saves the leakage power. As we move from micron (i.e greater than 90nm) technology nodes to sub-micron (i.e less than 90nm) technology nodes, leakage power dissipation dominates the dynamic power dissipation. It is therefore employed very frequently in modern SoCs. We shall talk about the structure of a power gate in this post.

Consider any CMOS digital logic circuit consisting of Pull-Up Network (made from PMOS transistors) and Pull-Down Network (made from NMOS transistors), as shown in the figure.

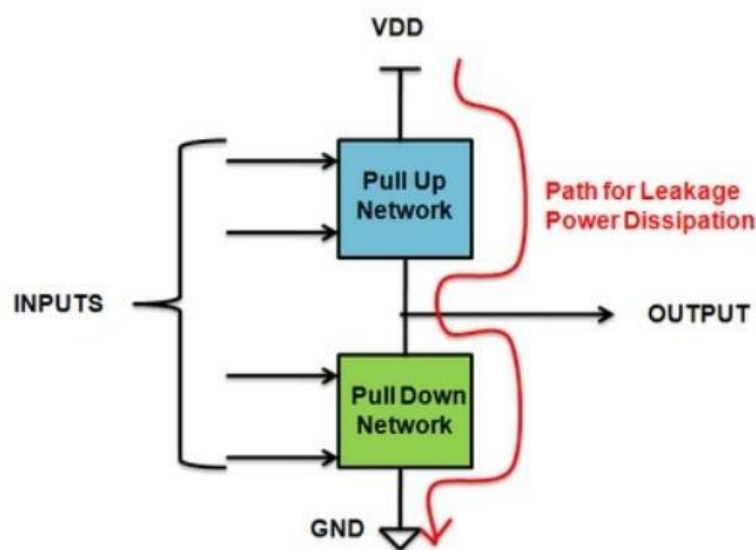


Fig 1.2: Power dissipation

At any point of time, if a direct path would exist from the power supply (VDD) and the ground (GND), the circuit would continue to dissipate leakage power. One can gate the power and ground terminals from the circuit when it is not intended to be used. That's what is accomplished by a power gate! Let's take a look at the circuit.

During normal operation, SLEEP = 0. Both the PMOS and NMOS Sleep Transistors (in blue and green respectively) are ON. And we have Virtual Power Rails and Virtual Ground which ensure normal circuit operation. However, during periods of low

activity, SLEEP 1. Hence the Sleep Transistors turn OFF. And a direct path from power rails to ground is broken and hence no leakage power is dissipated due to the Pull-up and Pull-down networks.

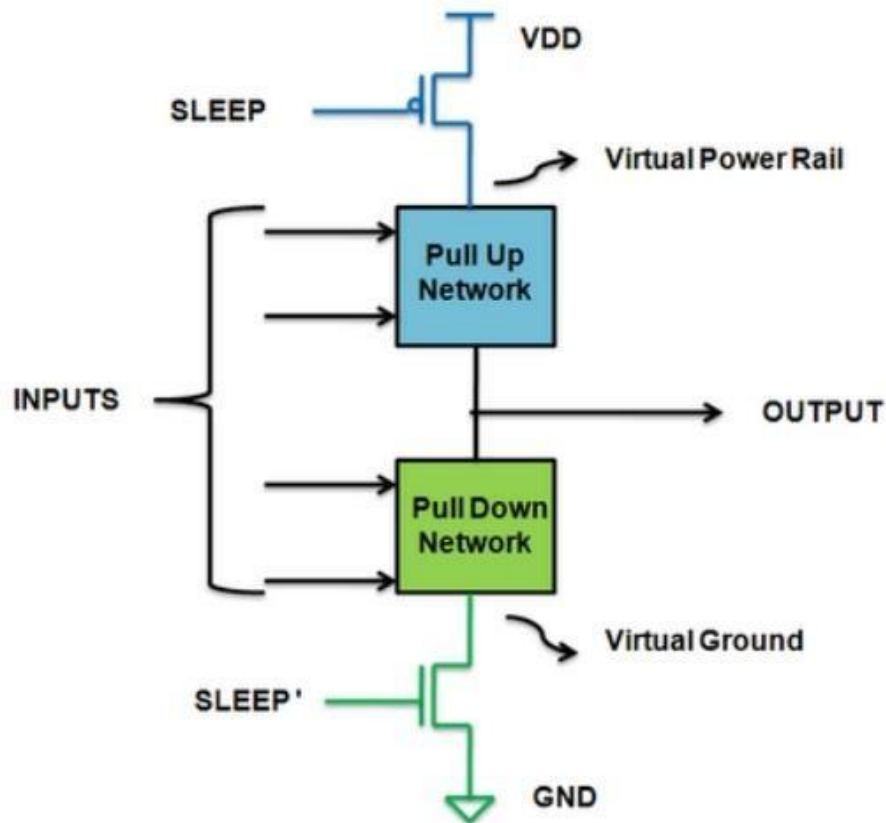


Fig 1.3: Vlsi Soc design power gating

Automatic gates commonly use electric motors or hydraulics to operate them. Swing gates rotate around hinges on gate posts at the sides of driveway entrances. Sliding gates are moved by a static motor. They typically have backup batteries, remote controls, other settings and can be solar powered too. Automatic gates replace the manual gate system that requires someone to physically open it and close it. Automatic gates offer a range of benefits, along with a range of installation options. They can be programmed to provide pedestrian access or voice control. They can use GPS or Wi-Fi integration, come equipped with obstacle sensors, timed movements, and a wealth of other convenient options. One of the great things about automatic gates is probably one of their most basic functions: they save effort. Gate automation provides not only added security but convenience too, either by sensors, or through the touch of a button.

Power-gating parameters Power gating implementation has additional considerations than the normal timing closure implementation. The following parameters

need to be considered and their values carefully chosen for a successful implementation of this methodology.

Power gate size must be selected to handle the amount of switching current at any given time. The gate must be bigger such that there is no measurable voltage (IR) drop due to the gate. Generally we use 3X the switching capacitance for the gate size as a rule of thumb. Designers can also choose between header (P-MOS) or footer (N-MOS) gate. Usually footer gates tend to be smaller in area for the same switching current. Dynamic power analysis tools can accurately measure the switching current and also predict the size for the power gate.

Gate control slew rate in power gating is an important parameter that determines the power gating efficiency. When the slew rate is large, it takes more time to switch off and switch-on the circuit and hence can affect the power gating efficiency. Slew rate is controlled through buffering the gate control signal.

Simultaneous switching capacitance has important constraint refers to the amount of circuit that can be switched simultaneously without affecting the power network integrity. If a large amount of the circuit is switched simultaneously, the resulting "rush current" can compromise the power network integrity. The circuit needs to be switched in stages in order to prevent this.

Power gate leakage are made of active transistors, leakage is an important consideration to maximize power savings.

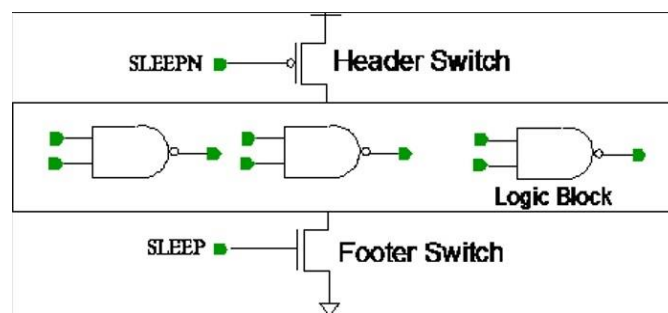


Fig 1.4: Footer Power gating systems

Fine-grain power gating is a sleep transistor to every cell that is to be turned off, imposes a large area penalty, and individually gating the power of every cluster of cells creates timing issues introduced by inter-cluster voltage variation that are difficult to resolve. Fine-grain power gating encapsulates the switching transistor as a part of the standard cell logic. Switching transistors are designed by either library IP vendor or

standard cell designer. Usually these cell designs conform to the normal standard cell rules and can easily be handled by EDA tools for implementation. The size of the gate control is designed with the worst case consideration that this circuit will switch during every clock cycle resulting in a huge area impact. Some of the recent designs implement the fine-grain power gating selectively, but only for the low V_t cells. If the technology allows multiple V_t libraries, the use of low V_t devices is minimum in the design (20%), so that the area impact can be reduced. When using power gates on the low V_t cells the output must be isolated if the next stage is a high V_t cell. Otherwise it can cause the neighboring high V_t cell to have leakage when output goes to an unknown state due to power gating. Gate control slew rate constraint is achieved by having a buffer distribution tree for the control signals. The buffers must be chosen from a set of always on buffers (buffers without the gate control signal) designed with high V_t cells. The inherent difference between when a cell switches off with respect to another, minimizes the rush current during switch-on and switch-off. Usually the gating transistor is designed as a high v_t device. Coarse-grain power gating offers further flexibility by optimizing the power gating cells where there is low switching activity. Leakage optimization has to be done at the coarse grain level, swapping the low leakage cell for the high leakage one. Fine-grain power gating is an elegant methodology resulting in up to 10X leakage reduction. This type of power reduction makes it an appealing technique if the power reduction requirement is not satisfied by multiple V_t optimization alone.

1.3 ZSCCMOS (Zigzag Super Cut-off CMOS)

It is used to reduce not only the switching power but also the leakage power. Due to its fast wakeup nature, the ZSCCMOS can be best suited to the clock-gating scheme. The wakeup time of the ZSCCMOS is estimated to be 12 times faster than the conventional Super Cut-off CMOS (SCCMOS) in 70-nm process technology. From the measurement of wakeup time in 0.6-/spl mu/m technology, it is observed to be eight times faster than the conventional scheme.

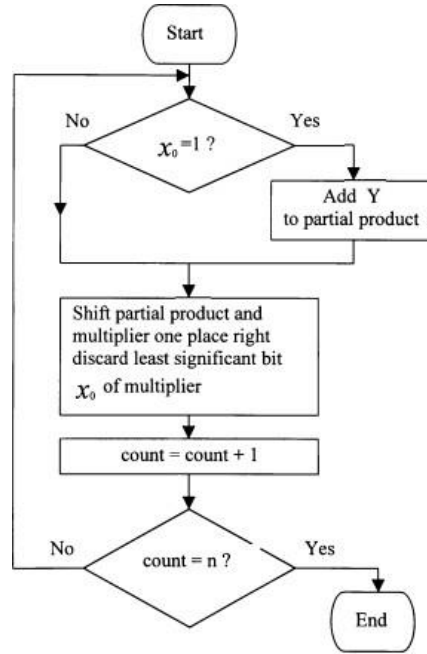


Fig 1.6 Software multiplier

This algorithm is used to reduce the number of partial product rows by about half, so, the speed of multiplication increases significantly and the chip area is reduced. The 3-bit recoding algorithm is also called the Modified Booth's Algorithm and was developed from Booth's algorithm [7]. A number of other multiple-bit recoding algorithms for multiplication have been developed [8]. Recent, a parallel hardware multiplier based on a 5-bit recoding algorithm has been proposed [9]. From the view of optimization, the 5-bit recoding algorithm is preferred to a 4-bit recoding algorithm. While more partial product rows can be reduced with the 5-bit recoding algorithm than with a 3-bit recoding algorithm, more complicated circuits are required to determine the odd multiples of the multiplicand. With the potential of improving both performance and the hardware requirements, the 5-bit recoding algorithm maybe good for a high bit multiplier, but, not for a low bit multiplier, such as, an 8 x 8 bit multiplier. Using the 5-bit recoding algorithm reduces the number of partial product rows to two. The partial products are selected from 17 different multiples of the multiplicand Y ($0, \pm Y, \pm 2Y, \pm 3Y, \pm 4Y, \pm 5Y, \pm 6Y, \pm 7Y, \pm 8Y$). Using the 3-bit recoding algorithm reduces the number of partial product rows to four. The partial products are selected from 5 different multiples of the multiplicand Y ($0, \pm Y, \pm 2Y$). The addition of four partial products can be changed to the addition of two binary numbers by using two rows of carry save adder arrays (CSA) with only a two gate delay introduced. The even multiples of the multiplicand Y can be implemented by using a hardwire shift.

For the 3-bit recoding algorithm, only the two's complement of Y needs to be determined. For the 5-bit recoding algorithm, additional high speed adders are required to determine odd multiples of Y . These high speed adders require more circuit into implement and suffer more time delay. For higher bit multipliers, the advantage of the 5-bit recoding algorithm can be seen. For example, for a 32×32 bit multiplier, the 5-bit recoding algorithm reduces the number of partial product rows to 8 and the 3-bit recoding algorithm reduces the number of partial product rows to 16. The reduction of the number of partial product rows is apparent. The 3-bit recording algorithm is used here because our objective is to build an 8×8 bit multiplier block. With this block, higher bit multipliers can be built.

CHAPTER - 2

LITERATURE SURVEY AND OBJECTIVES

This chapter discusses about the survey done for the Performance evaluation of an efficient fine grained sleep technique for parallel architectures.

2.1 LITERATURE SURVEY

According to wang and chandrakasan (2005) the wireless sensor nodes should be self-powered and must scavenge energy from the environment.[1] Due to the requirements for energy scavenging, the key metric is minimizing energy dissipation rather than processor speed. The Fast Fourier Transformation (FFT) processor was designed using a modified standard logic cell library, custom multiplier generators, and custom memory generators. The FFT processor, energy models estimate the optimal supply voltage to be in subthreshold where the supply voltages are below the threshold voltages. The FFT processor contains eight 128 W 16 b RAM blocks and four 256 16 b ROM blocks. First, we analyze the functionality of conventional memory designs in subthreshold. In standard memory schemes, a six-transistor (6T) SRAM is used. Current memory design challenges, such as bitline capacitance, bitline leakage, speed, and sense amplifier design, worsen as voltage supplies scale down and at worst case process variations. To create a subthreshold logic cell library, first we analyzed traditional logic circuits at low voltages. Parallel leakage, stacked transistors, and sneak leakage effects affect functionality during low-voltage operation when the gate drives are lowered to the point that the on currents are competing with the leakage currents. In this section, we show how we developed a methodology to design subthreshold logic cells that are used in the FFT processor design and ensure that these effects are minimized. A printed circuit board (PCB) is used to test the functionality of the subthreshold FFT chip. The Tectronix pattern generator has 2–5-V outputs which are level converted down to 500 mV using resistive dividers to interface with the subthreshold FFT chip. The outputs of the subthreshold FFT chip are level converted up to 5 V using off-the-shelf comparators. The lowest voltage supply for correct operation is 180 mV with a clock speed of 164 Hz. The power dissipated at 180 mV is 90 nW for 16-b 1024-pt operation. Fig. 15 shows an oscilloscope plot of various outputs, (the clock, two output bits and the data ready signal) from the FFT chip operating at 180 mV. However, the minimum supply voltage does not correspond to the optimal operating point which minimizes energy dissipation. The

minimum energy point of the 16-b 1024-pt FFT occurs at 350 mV. The power dissipated at the optimum is 600 nW at a clock frequency of 10 kHz and the energy dissipated is 155 nJ/FFT. Fig. 16 shows the measured and estimated energy dissipated for the 16-b 1024-pt FFT over – mV. The subthreshold FFT is fabricated in a standard 0.18- m logic process without any additional process steps or biasing techniques. The FFT operated as low as 180 mV. It showed the optimum supply voltage to be at 350 mV with a clock frequency of 10 kHz, where it dissipated 155 nJ for a 16-b 1024-pt FFT.

Fojtik and Mathew (2013) Explained about the smallest known, 8.75 mm, energy-autonomous microsystem, for temperature sensing containing a commercial microprocessor.[2] The microsystem is powered by a 2 mm, 5.48% efficiency Silicon solar cell and a 12 Ah thin-film Li battery. To achieve zero-net-energy consumption using these small, low-power, and low-capacity power supplies, all circuit components must be optimized with ultra-low power techniques. The microsystem spends the majority of its lifetime in a 550 pW standby mode, where the processor is power gated and data is retained in an ultra-low-leakage 3.3 fW/bitcell SRAM. A 50 Hz switched capacitor DC-DC converter delivers energy to these load components in standby mode. The same DC-DC converter uses a 1.2 MHz clock to harvest solar energy and recharge the battery. It supplies active mode power with 38% power efficiency, 384 ppm/nW load regulation, and 24 ppm/mV line regulation. In active mode, temperature is measured, processed with a 28 pJ/cycle, 72 kHz, 0.4 V ARM Cortex-M3 processor, and logged into SRAM. Myriad sensor applications have use models similar to the proposed device, collectively pointing to the need for mm-scale systems to replace today's bulky, expensive, and power-hungry wireless sensors. These mm-scale sensors and ubiquitous computers represent the continuation of Bell's Law and the next frontier in integrated circuit design. Adding a wireless transceiver would facilitate communication to and from the sensor system, and poses both average and peak power budget challenges. An ultra-low power radio is demonstrated with a transmission energy of 4.7 nJ/bit. To meet peak power constraints, the radio is supplied from an on-chip capacitor which is recharged between the transmission of each bit. Assuming the presence of this radio in the system, that each sensor measurement takes 10,000 clock cycles to complete, and that many measurements are transmitted together in a burst to reduce synchronization overhead, 280 nJ would be consumed per measurement for sensing and computation while 52 nJ would be consumed per measurement for transmission. This transceiver allows short

range communication with a base station and future work should consider peer-to-peer symmetric sensor networks. The proposed system harvests solar energy to keep the battery continually charged and the SCVR continually running. However, if a lack of sunlight causes the battery to completely drain system operation ceases. To enable autonomous startup from a zero-power state, future work includes developing energy-harvesters with low startup voltages to begin battery recharging. It should include data logging in non-volatile memory to log temperature or other sensor data even when unpowered. To enable low cost implementation of wireless sensor systems, a temperature sensor with one point calibration would be beneficial by reducing test complexity. A wide range of temperature sensor topologies are available with varying accuracies and energy consumptions. The system is designed to operate autonomously without any external power source and is intended for use in wireless sensor networks and other applications where power is limited. The system is composed of a microcontroller, a radio transceiver, a solar cell array, and a stacked battery module. The stacked battery module includes three thin-film lithium-ion batteries stacked together, which provides a high energy density while maintaining a small form factor. The solar cell array is designed to capture sunlight and charge the battery module. The system also features power management and energy harvesting circuits to ensure efficient use of the energy harvested. The authors present the design and implementation of the system and evaluate its performance through several experiments. The experimental results demonstrate that the system can operate autonomously for several months without any external power source. The system also demonstrates the ability to transmit sensor data wirelessly to a remote base station. Overall, the paper presents a low-power and energy-autonomous sensor system that utilizes stacked battery and solar cells. The proposed system can be useful in designing wireless sensor networks and other applications where power is limited, and energy harvesting is necessary. The design can be further optimized and scaled to meet specific application requirements.

A variety of approaches to reduce power consumption in CMOS circuits have been presented which involve optimization at all levels of the system design. At the technology level, the key low-power consideration is threshold reduction which allows supply voltage scaling to below 1 V. An optimum threshold voltage selection based on trading leakage and switching currents was presented which indicates that the threshold voltage should be scaled to the 0.3-0.4 V range as opposed to current day technology which uses a

threshold voltage on the order of 0.7 V-1 V. At next level, the physical, circuit, and logic levels are optimized for low power. This includes activity driven place and route, the use of minimum sized devices in cell design, use of reduced swing logic, and logic level optimization and power down. Optimization at the architectural level can have a major impact on power. An architecture driven voltage scaling strategy was presented in which parallel and pipelined architectures were used to scale the supply down into the 1-1.5 V range without loss in functionality. This strategy combined with threshold reduction allows voltage reduction to about 0.5 V, which would result in two orders of magnitude of power reduction compared to conventional designs running at 5 V. Various strategies were also presented for minimizing switched capacitance at the architectural level which includes choice of number representation, exploitation of signal correlations, and minimizing glitching transitions. Finally, the choice of algorithm is the most highly leveraged decision in meeting the power constraints. The ability for an algorithm to be parallelized is critical and the basic complexity of the computation must be highly optimized. A number of the low-power techniques that were presented were applied to the design of a chipset for a portable multimedia terminal that supports pen input, speech WO and full-motion video. The entire chipset that performs protocol conversion, synchronization, error correction, packetization, buffering, video decompression and D/A conversion operates from a 1.1 V supply and consumes less than 5 mW-which is more than three orders of magnitude lower power compared to equivalent commercial solutions. The 32b, 2-phase domino, Han-Carlson adder design contains NMOS and PMOS sleep transistors inserted between the virtual and real supply grids on chip (Fig. 6.1.2). The sleep transistors are ON during active mode and are turned OFF along with the local clock. To further improve on-resistance and leakage reduction capabilities of the sleep transistors, combinations of (1) gate overdrive during “active” and underdrive during “idle”, and (2) forward body bias (FBB) during “active” and zero body bias (ZBB) during “idle” are used. Low-VT devices are used for the sleep transistors to minimize performance and area impacts. The sleep transistors are distributed uniformly across the adder layout to prevent undesirable current crowding in the power grids. Metal levels M4 and M5 in the traditional power grid are re-connected through the sleep transistors instead of M4/M5 vias. The virtual supply transition waveform, following sleep transistor turn-off, can be measured directly using an on-chip 3b A/D converter using inverters of varying pre-characterized trip points. Dynamic body bias for the adder core, to apply FBB during “active” and ZBB during “idle”, is also implemented.

M.S Hrishikesh (2002) determined about the effects of varying clock frequency on the performance of a superscalar pipeline. We determined the amount of useful logic per stage (Φ_{logic}) that will provide the best performance is approximately 6 FO4 inverter delays for integer benchmarks. If Φ_{logic} is reduced below 6 FO4 the improvement in clock frequency cannot compensate for the decrease in IPC. Conversely, if Φ_{logic} is increased to more than 6 FO4 the improvement in IPC is not enough to counteract the loss in performance resulting from a lower clock frequency. For vector floating-point benchmarks the optimal Φ_{logic} was at 4 FO4. The clock period ($\Phi_{\text{logic}} + \Phi_{\text{overhead}}$) at the optimal point is 7.8 FO4 for integer benchmarks, corresponding to a frequency of 3.6GHz at 100nm technology. For vector floating-point benchmarks the optimal clock period is 5.8 FO4 which corresponds to 4.8GHz at 100nm technology. These optimal clock frequencies can be achieved only if on-chip microarchitectural structures can be pipelined to operate at high frequencies. We identified the instruction issue window as a critical structure, which will be difficult to scale to those frequencies. We propose a segmented instruction window design that will allow it to be pipelined to four stages without significant decrease in IPC. Scaling the pipeline depth of the window to 10 stages only decreases the IPC of SPEC 2000 integer benchmarks by 11% and floating-point benchmarks by 5%. Although this study uses the parameters of a 100nm technology, our use of the technology-independent FO4 metric will permit our results to be translated to other technologies. The 1 FO4 corresponds to 360 picoseconds times the transistor's drawn gate length. But, for highly tuned processes, such as the Intel 0.13-m process, the drawn gate length and effective gate length may differ substantially. However, our estimate of the optimal pipeline depth remains unchanged regardless of the exact value assigned to a FO4 delay though the actual cycle time will depend on the operating conditions and process technology specifications. While we did not consider the effects of slower wires, they should not affect this study, which uses a fixed microarchitecture. To first order, wire delays remain constant as a fixed design is scaled to smaller feature sizes. Although wire resistance increases, wire lengths decrease, thus preserving the absolute wire delay across technologies. However, long wires that arise as design complexity increases can have a substantial impact on the pipelining of the microarchitecture. For example, the high clock rate target of the Intel Pentium IV forced the designers to dedicate two pipeline stages just for data transportation. Microprocessor performance has improved at about 55% per year for the last three decades, with much of the gains resulting from higher

clock frequencies, due to process technology and deeper pipelines. However, our results show that pipelining can contribute at most another factor of two to clock rate improvements. Subsequently, in the best case, clock rates will increase at the rate of feature size scaling, which is projected to be 12-20% per year. Any additional performance improvements must come from increases in concurrency, whether they be instruction-level parallelism, thread-level parallelism, or a combination of the two. If the goal is to maintain historical performance growth rates, concurrency must start increasing at 33% per year and sustain a total of 50 IPC within the next 15 years.

The 32b, 2-phase domino, Han-Carlson (2004) adder design contains NMOS and PMOS sleep transistors inserted between the virtual and real supply grids on the chip (Fig. 6.1.2). The sleep transistors are ON during active mode and are turned OFF along with the local clock. To further improve on-resistance and leakage reduction capabilities of the sleep transistors, combinations of (1) gate overdrive during “active” and underdrive during “idle”, and (2) forward body bias (FBB) during “active” and zero body bias (ZBB) during “idle” are used. Low-VT devices are used for the sleep transistors to minimize performance and area impacts. The sleep transistors are distributed uniformly across the adder layout to prevent undesirable current crowding in the power grids. Metal levels M4 and M5 in the traditional power grid are re-connected through the sleep transistors instead of M4/M5 vias. The virtual supply transition waveform, following sleep transistor turn-off, can be measured directly using an on-chip 3b A/D converter using inverters of varying pre-characterized trip points. High-frequency testing is accomplished using a scan methodology for the ALU input vectors and at-speed output capture. Measurement of the virtual supply transition waveform, following sleep transistor turn-off, is done by capturing outputs of the two 3b A/D converters after a preset number of clock cycles. Separate supply pins are provided for the buffers driving the sleep transistors so that switching and leakage power of the adder can be measured independently of the energy required to switch the sleep transistors between active and idle. The overall performance impact of these dynamic leakage control techniques is dictated by the time required to switch the sleep transistor gate or the body node voltage during “idle” to “active” transitions. Measurements and simulations show that this time is 1 clock cycle for sleep transistors and 3-4 cycles for body bias, compared to less than a cycle for clock gating. For any of these dynamic leakage control techniques to achieve reduction in overall power, the leakage energy saved during the “idle” period must be larger than any energy overhead

incurred during transitions between “idle” and “active” modes. For sleep transistors, energy is consumed during active-idle transitions due to discharging and charging of the capacitances at the virtual supply and internal circuit nodes by adder leakage currents as they converge to steady-state values. This “leakage convergence” time is measured to be $1\mu\text{s}$ at 75°C and increases to 10ms when an external 133nF decap is added to the virtual supply. MTCMOS is a low-power technique that uses transistors with multiple threshold voltages to reduce leakage current in standby mode.

A theory has been presented (2003) of the optimum pipeline depth for a microprocessor taking into account both power and performance. It was found that only one optimal solution exists for this problem and the nature of that solution has been characterized. For some power/performance metrics one finds that the optimum design contains only a single stage in the processor, whereas for other metrics, including the performance only metric, a pipelined design results. Consideration of power in the optimization problem always leads to shorter pipelines than if power were no consideration. Simulations were performed for 55 workloads, covering traditional, SPEC, and modern workloads on a 4 issue, in-order superscalar microprocessor model. If one takes an average over the different analysis methods and workloads, the optimum pipeline depth was found to be 7 stages, corresponding to a 22.5 FO4 design point. Significant differences are seen between different types of workloads, which can be attributed to the different processing requirements of the different workloads. The SPEC integer workloads are seen to be less stressful of the processor than real workloads of various types. The parameters, which have the greatest impact on the optimum design point, are the two exponents, m and η . The theory is so sensitive to these parameters because they occur in the exponent. Unfortunately, these parameters are perhaps the least well characterized. The value of the latch growth factor depends on details of the logic and circuit design of the particular pipeline and microarchitecture being considered. Which particular exponent, m , to use in the power/performance metric is still an open question. Indeed these considerations led us to formulate a theory, which kept these values as parameters. The theory is applicable for all values, but the numerical results depend sensitively on the chosen values for these parameters. This theory can be used to investigate numerous dependencies as new microarchitectures, workloads, or technologies arise. This can be done without the need for the detailed simulations, which were used to verify the theory. At the circuit-level, the authors discuss techniques such

as supply voltage scaling, power gating, and multi-threshold voltage design. They also describe the use of sleep transistor networks, which can selectively turn off parts of the circuit during idle periods to reduce leakage. At the device-level, the authors discuss the use of different gate dielectric materials and gate stack engineering to reduce gate leakage. They also describe techniques such as channel engineering, source/drain engineering, and substrate engineering, which can help reduce sub-threshold leakage and drain-induced barrier lowering. Overall, the paper provides a comprehensive overview of the different leakage mechanisms in deep sub-micrometer CMOS circuits and the various techniques that can be used to reduce leakage. The information presented in the paper can be useful for circuit designers and researchers working on low-power circuit design in modern CMOS technology nodes.

With the reference of Mingoo seok (2011) circuit and architecture techniques to enhance energy efficiency in the subthreshold regime, with application to an FFT module. In contrast to the use of modest pipelining in past subthreshold designs, we advocate the use of super-pipelining to suppress leakage energy consumption while simultaneously improving performance. The latch-based Fig. 19. Die photograph of the FFT core in 65 nm CMOS. Pipelining scheme recovers delay averaging effects by introducing a longer effective stage length through time borrowing to effectively mitigate variations. A parallel-pipelined architecture maximizes CE and memory utilization, enabling greater voltage scaling range and further enhancing throughput. In addition, an energy efficient and robust FIFO design and variation-tolerant clock distribution network are employed. The proposed FFT design is fabricated in 65 nm CMOS and measurements indicate that it successfully operates at an optimal operating point of 270 mV at a clock frequency of 30 MHz. Its energy efficiency is 2.4 higher than previous state-of-art FFT designs, while performance is more than 10 higher than past ULV designs, demonstrating the feasibility of very low voltage design for moderate to high performance systems. In ultra-low voltage designs, the voltage levels are reduced to minimize power consumption, resulting in slower circuit performance. To address this issue, the authors propose a pipeline strategy that optimizes energy efficiency by minimizing the pipeline stages while still meeting the target throughput requirement. The pipeline strategy involves dynamically resizing pipeline stages based on the throughput requirements to minimize energy consumption. The proposed methodology includes an algorithm that calculates the optimal pipeline stage sizes, and a framework for generating pipelines that satisfy the given requirements.

The algorithm optimizes the pipeline stage sizes by considering the circuit delays, leakage power, and dynamic power, while the framework generates pipelines that satisfy the optimal pipeline stage sizes. The proposed pipeline strategy is evaluated using several benchmark circuits and compared with other design techniques such as voltage scaling and power gating. The experimental results show that the proposed pipeline strategy achieves up to 46% energy reduction compared to the baseline design, while still meeting the throughput requirement. Overall, the paper proposes a pipeline strategy that improves energy efficiency in ultra-low voltage design. The proposed approach can be useful in designing low-power and energy-efficient systems, especially in battery-operated devices where power consumption is critical.

The parasitic resistance (2004) in an MOS transistor used as on-chip decoupling capacitance has been determined. The presented theory agrees well with both simulations and measurements. At low frequencies, the resistance is $1/12$ th of the channel resistance. Therefore, the channel length of the transistors used as decoupling capacitance can be chosen 12 times larger than the channel length of the transistors in the logic circuits. This would still give an RC time constant of the decoupling capacitance lower than that of the logic circuits. The distributed nature of the capacitance in an MOS transistor causes a frequency-dependent parasitic resistance as demonstrated by measurements. This can be utilized to form a damping resistance reducing resonance oscillations while still achieving low resistive noise. Therefore, the large resistance, appearing at low frequencies, will contribute to the damping of the LC circuit while the low resistance at high frequencies will assure low voltage drop over the decoupling capacitance. On-chip decoupling capacitors are used to provide a stable supply voltage to the integrated circuits by filtering out the noise and ripple in the power supply. In this paper, the author highlights the issue of parasitic resistance in MOS transistors used as on-chip decoupling capacitance. The parasitic resistance in MOS transistors results from the resistance of the source, drain, and substrate contacts, as well as the channel resistance. The resistance increases with the reduction in the feature size, which leads to higher resistance and reduced decoupling capacitance. The author presents a theoretical model for analyzing the parasitic resistance in an MOS transistor and its effect on the decoupling capacitance. The model takes into account the device geometry, doping concentration, and temperature, and provides a means to estimate the parasitic resistance and its impact on the decoupling capacitance. The paper also presents

experimental results to validate the theoretical model. The experiments were performed on a 0.8 μm CMOS process, and the results show that the parasitic resistance in the MOS transistor can significantly reduce the decoupling capacitance, particularly at high frequencies. Overall, the paper highlights the issue of parasitic resistance in MOS transistors used as on-chip decoupling capacitance and presents a theoretical model to estimate its impact. The study is essential for designing high-performance integrated circuits with stable power supply, particularly in sub-micron technologies.

Dynamic V_{th} SRAM (DTSRAM) architecture can be used to reduce leakage energy dissipation in memory structures. (1997) Using body biasing, the subthreshold leakage can be reduced without sacrificing data stability [81]. In a time-based dynamic scheme, high is assigned to the cache lines which are not accessed for a certain period (30 s–100 s), and a low is assigned to the cache lines which are in frequent use to maintain high performance. Fig. 50 depicts the two dominant leakage paths for a conventional six-transistor SRAM cell, the -to-ground and the bitline-to-ground leakage paths. the performance loss due to negative body bias (i.e., high V_{th}) is considerable. To overcome these difficulties, properties of temporal and spatial locality of cache access can be used. In current deep-submicrometer devices with low threshold voltages, subthreshold and gate leakage have become dominant sources of leakage and are expected to increase with the technology scaling. GIDL and BTBT may also become a concern in advanced CMOS devices. To manage the increasing leakage in deep-submicrometer CMOS circuits, solutions for leakage reduction have to be sought both at the process technology and circuit levels. At the process technology level, well-engineering techniques by retrograde and halo doping are used to reduce leakage and improve short-channel characteristics. At the circuit level, transistor stacking, multiple V_{th} , dynamic V_{th} , multiple V_{dd} , and dynamic V_{dd} techniques can effectively reduce the leakage current in high-performance logic and memory designs.

By using local sleep devices and provided an analysis of sneak leakage paths to facilitate the approach.(2003) We presented a set of design guidelines for preventing common sneak leakage paths. A 0.13- μm MTCMOS test chip confirms that gate-level sleep devices can prevent sneak leakage and provide total standby leakage savings measured from 7.0 to 8.6 for different stored states. Fig. 13 shows the annotated die photo of the test chip. The three slices (containing 12 CLBs) cover 300 μm by 740 μm . We

also propose fine-grained sleep regions based on local sleep devices and implement them on the test chip. The sleep regions allow idle blocks to enter sleep mode while other circuit components remain active and with unaffected performance. The total steady-state power (clock-gated) for an active CLB reduces by from 10% to 2.2 for different configurations. The CLBs use three power supplies to facilitate measurement: flip-flop power supply, sleep power supply, and core supply. The sleep power supplies the sleep signal network, the FF supply powers the flip-flops, and the core supply powers the rest. All HSPICE simulations and measurements are for VDD at 1.0V. During the period that the clock pulse is high, the transmission gate of the latch is on, and the output of the latch (Q) takes the same value as the input (D). When the clock signal changes to low, the transmission gate is turned off. However, the transistors along one of the two feedback paths turn on, completing the feedback loop. The inverter and the feedback loop retain the sampled data value until the following clock cycle. The clock signal has a 50% duty cycle while the data signal is a simple step function. We simulated transistors at 100nm technology and performed experiments similar to those by Stojanović et al. [14], using the same P-transistor to N-transistor ratios. In our experiments, we moved the data signal progressively closer to the falling edge of the clock signal. Eventually when D changes very close to the falling edge of the Clk signal the latch fails to hold the correct value of D. Latch overhead is the smallest of the D-Q delays before this point of failure. We estimated latch overhead to be 36ps (1 FO4) at 100nm technology. Since this delay is determined by the switching speed of transistors, which is expected to scale linearly with technology, its value in FO4 will remain constant at all technologies. Note that throughout this paper transistor feature sizes refer to the drawn gate length as opposed to the effective gate length. In MTCMOS, the power supply to each logic block can be independently turned off using a sleep transistor to reduce leakage current. However, this technique suffers from the additional overhead of sleep transistor control and the resulting increase in design complexity. To overcome these limitations, the paper proposes a distributed MTCMOS approach, where each logic block has its own sleep transistor, resulting in simplified control circuitry. The authors also propose a leakage reduction methodology that uses optimal sleep transistor sizing, selective threshold voltage assignment, and judicious clock gating. The methodology is evaluated using a test chip in a 90nm CMOS process. The experimental results show that the distributed MTCMOS approach reduces leakage by 93% and dynamic power by 42% compared to a conventional design. The proposed methodology also shows significant improvements in

leakage reduction compared to the traditional MTCMOS approach. Overall, the paper provides a comprehensive methodology for reducing leakage current in distributed MTCMOS circuits. The proposed approach can be beneficial in the design of low-power circuits for portable devices and other applications where power consumption is critical.

2.2 LITERATURE SUMMARY

A variety of approaches to reduce power consumption in CMOS circuits have been presented which involve optimization at all levels of the system design. At the technology level, the key low-power consideration is threshold reduction which allows supply voltage scaling to below 1 V. An optimum threshold voltage selection based on trading leakage and switching currents was presented which indicates that the threshold voltage should be scaled to the range as opposed to current day technology which uses a threshold voltage. At next level, the physical, circuit, and logic levels are optimized for low power. This includes activity driven place and route, the use of minimum sized devices in cell design, use of reduced swing logic, and logic level optimization and power down. Optimization at the architectural level can have a major impact on power. An architecture driven voltage scaling strategy was presented in which parallel and pipelined architectures were used to scale the supply down into the 1-1.5 V range without loss in functionality.

A parallel-pipelined architecture maximizes CE and memory utilization, enabling greater voltage scaling range and further enhancing throughput. In addition, an energy efficient and robust FIFO design and variation-tolerant clock distribution network are employed.

CHAPTER 3

EXISTING METHODOLOGY

The existing system works on parallel and pipelined architectures, however, have an emphasis on nominal VDD designs, having little or no attention on a crucial issue that has greater significance in near- and subthreshold circuits: active-leakage dissipation. As VDD is scaled from nominal to near- and subthreshold levels, increasingly slowed-down circuits accumulate more leakage power per clock cycle. Eventually, leakage energy dissipation starts to offset the quadratic savings of dynamic energy dissipation. Active-leakage energy dissipation, consequently, becomes critical to runtime computing energy efficiency. The VDD level at which the total energy consumption starts to increase is defined as energy-optimal voltage (VOPT). The energy consumption at VOPT is denoted by EOPT.

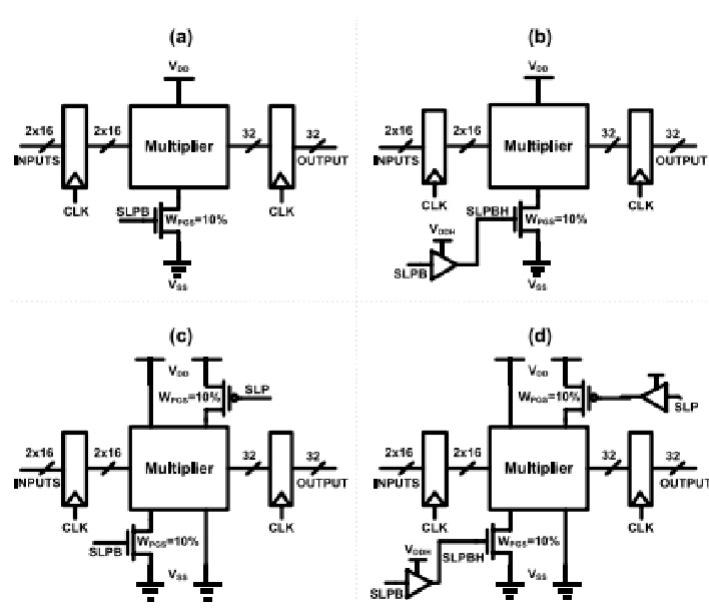


Fig. 3.1 Four PGS designs(a) Footer PGS.(b) Footer PGS with gate overdrive voltage.(c) ZSCCMOS.(d) ZSCCMOS with gate overdrive voltage

By comparing four PGS design techniques: a footer PGS, a footer PGS with gate overdrive voltage, a ZSCCMOS, and a ZSCCMOS with gate overdrive voltage (Fig. 3.1). In the two footer-only PGS designs, the PGS is sized to $\sim 10\%$ of the total nFET width of the main circuits (a 16-bit array multiplier). In the two ZSCCMOS designs, the footer is

sized to $\sim 10\%$ of the total nFET width, and the header PGS is sized to $\sim 10\%$ of the total pFET width. We include the overhead of level converters [25] in our simulations for the designs using overdrive voltage.

The ZSCCMOS can reduce T2WKU over the footer PGS design by more than one order of magnitude. In contrast, the improvements provided by gate overdrive voltage are marginal for both footer PGS and ZSCCMOS designs. These results make us again to confirm the importance of limiting the switching of CINT to scale T2WKU down. We also find the similar results for T2SLP, although it is less crucial for implementing temporarily fine-grained sleep modes. Fig. 2 also shows PSLP comparisons across the four structures. PSLEEP is defined as the average power dissipation, while we sleep circuits for ~ 1000 FO4 delays, and then, wake them up. The ZSCCMOS design consumes less PSLEEP than the footer PGS design by $1.74\times$ at 0.55 V and by $1.37\times$ at 0.35 V because of smaller ETRAN. The ZSCCMOS design with gate overdrive voltage consumes slightly more than the ZSCCMOS due to slightly larger ETRAN and the power dissipation of level converter circuits. Finally, leakage power is a strong function of process and temperature. Therefore, it is important for sleep techniques to adapt to process and temperature variations and determine when to enter and exit sleep modes. ZSCCMOS has a notable issue, which has limited its use in superthreshold voltage circuits. The matter in question is the sneak gate leakage path created by the use of both footer and header PGS. Further more, gate leakage through the paths can contribute to 83% of total leakage dissipation during a sleep mode at 1 V. the contribution of gate leakage to total leakage dissipation during a sleep mode becomes 23.8% at 0.5 V and only 6.5% at 0.3 V.

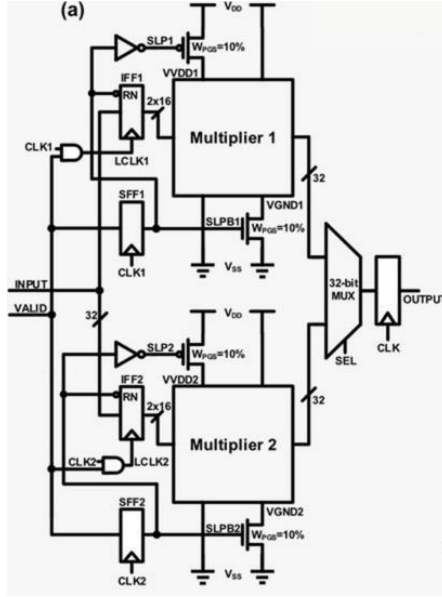


Fig. 3.2 A 16-bit two-way parallel multiplier with a temporarily fine-grained PGS technique.

This shows the test circuits that has been used in the experiment. It consists of two 16-bit array multipliers (M1 and M2), a 32-bit 2-to-1 multiplexer, input flip-flops (IFF1 and IFF2), and sleep flip-flops (SFF1 and SFF2). M1 and M2 employ the ZSCCMOS technique with no gate overdrive voltage. The multiplexer employs no PGS. CLK is a synchronous clock signal for nonparallel portions of the circuit, while CLK1 is a 2× slower clock. CLK2 is the inverted version of CLK1. The data input (INPUT) to the test circuits is accompanied with a VALID signal that indicates whether INPUT is valid or not. Both INPUT and VALID signals are synchronized with CLK. In addition, by AND-ing VALID and CLK1 (or CLK2), we can generate LCLK1 (or LCLK2), which is used in the input flip-flops of M1 (or M2) for clock gating. We present the detailed operating waveforms of the test circuits in Fig. 1(b). In the waveforms, at the clock cycle 1, the first input is not valid (VALID transitioning to LOW before the rising clock edge). SFF1, clocked by CLK1, captures VALID and generates SLPB1 = LOW and SLP1 = HIGH signals. This puts M1 into sleep mode and minimizes its active-leakage dissipation. The IFF1 of M1 is reset, forcing the inputs of M1 to LOW, so that every internal node in M1 settles to a predefined state. The IFF1 is also clock-gated, such that their dynamic energy consumption is reduced. Similarly, just before the next rising edge of CLK, another invalid input comes in (VALID is still LOW at the rising clock edge). VALID signal, this time, is captured by SFF2 at the rising edge of CLK2. This makes SLPB2 = LOW and SLP2 = HIGH to turn OFF M2 during that cycle. The IFF of M2 is reset, forcing internal nodes in M2 to predefined states. In

addition, IFF2 is clock-gated by LCLK2 to reduce the dynamic energy consumption. At Cycle 3, VALID becomes HIGH. The SFF1 captures the VALID signal at the rising edge of CLK1, which modulates SLPB1 and SLB1 to wake up M1. The wake-up process is finished before the IFFs of M1 sample INPUT at the rising edge of LCLK because of a very short T2WKU of <1 FO4 delays, and the time difference between the rising edges of CLK1 and LCLK1. The time difference is provided by the AND gate delay for generating LCLK1 from CLK1 and VALID. In addition, the difference between clock-to-q delay of the resettable IFFs and the regular SFFs (~ 3.3 FO4 versus ~ 2.3 FO4) can also contribute to the time difference. Although we use those intrinsic time differences for accommodating short T2WKU, it is also possible to intentionally add delay to ensure that the wake-up process finishes before IFF sample new input, or to latch the sleep signals with a transparent low latch, so that the sleep signals drive the PGSs before the rising edge of the clock (when a new input comes in). After the wake-up process, M1 can operate over the sampled inputs under usual conditions, while M2 is kept in the sleep mode to maximize energy-efficiency. At the beginning of Cycle 4, the SFF of M2 captures the VALID signal, which is still HIGH at the rising edge of CLK2. This makes SLPB2 and SLP2 to HIGH and LOW, respectively, waking up M2, which then performs computation over the new inputs, sampled at the rising edge of LCLK2. Fig. 3(c) shows the energy consumption per cycle for the same target TCYCLE of 20 ns. The VDDs to meet the target TCYCLE are 345 mV (proposed), 400 mV (baseline), 345 mV (simply pipelined), and 335 mV (simply parallelized). The proposed design exhibits less energy per cycle than the baseline and the simply parallelized designs across utilizations. At hardware utilization levels less than 50%, the proposed design outperforms the simply pipelined design as well. At the utilization of 10%, the proposed design consumes $2.4\times$ less energy than the simply parallelized design. We also investigate the gains in near-threshold voltage circuits. The target TCYCLE is set to 3.3 ns, which raises VDDs to 590 mV for the baseline, 505 mV for the simply pipelined, 470 mV for the simply parallelized, and 480 mV for the proposed design., the proposed design can achieve comparable or better energy efficiency across hardware utilizations than all the other architectures.while we focus on studying small-scale test circuits, i.e., multipliers, based on the results of the study, we can speculate the applicability of the proposed technique on larger scale systems. We think that the proposed technique can more benefit the hardware whose power dissipation is dominated by active leakage, particularly due to the use of a deeply scaled VDD and the use of massive parallelism.

CHAPTER - 4

PROPOSED SYSTEM

In order to improve efficiency, the existing system as parallel multiplexer architecture will be analyzed using 130nm, 32nm, and CMOS technology. Further, parallel multiplier architecture will be modified and analyzed using 130nm, and 32nm CMOS technology in terms of power, speed, and power delay product. This outperforms an efficient parallel multiplexer architecture for high-speed and low-power applications.

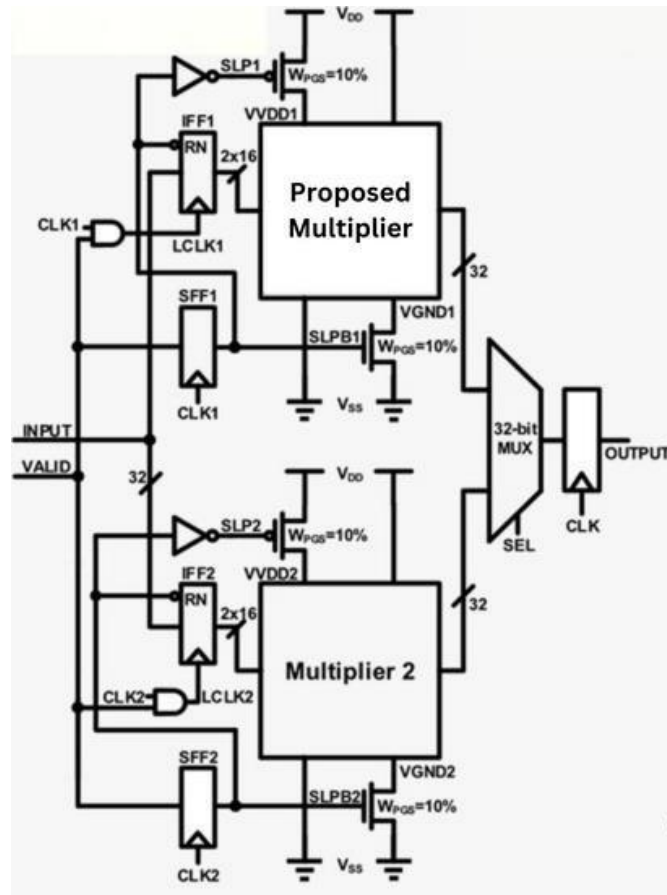


Fig. 4.1 *Proposed 16-bit two-way parallel multiplier with a temporarily fine-grained PGS technique.*

4.1 PROBLEM DEFINITION

To analyse an Performance evaluation of an efficient fine grained technique for parallel architectures

4.2 OBJECTIVES OF THE PROJECT

- To propose a PGS design technique, inspired by the so-called zigzag super cut off CMOS, in order to optimize the overheads of mode transitions of PGS in near- and subthreshold circuits.
- This technique enables to have circuits in sleep mode for as short as a single clock cycle with a negligible amount of energy and delay overheads.
- Comparison on performance will be analysed with existing and proposed systems in terms of power, delay and power delay product which will be used for low power applications.

4.3 PARALLELISM AND PIPELINING IN NEAR- AND SUBTHRESHOLD CIRCUITS

Increasing hardware-level parallelism and pipelining are notable architectural strategies for enhancing computing throughput. These approaches can also provide significant energy-efficiency gains, since the improved throughput can be traded off for energy savings via voltage scaling.

The investigation is to investigate the efficacy of those architectures in near- and subthreshold circuits. In nominal voltage circuits, the classic study has shown that parallelism and pipelining can allow the use of lower supply voltages for the same throughput, improving energy-efficiency. The results show that two-way parallel and two-stage pipeline architectures can reduce V_{DD} from 5 to 2.9 V and achieve 2.5× and 2.8×, respectively, energy savings over a baseline design. However, those architectures become significantly less effective on improving energy efficiency in near- and subthreshold circuits.

In order to investigate parallelism and pipelining in near- and subthreshold circuits, we use three test architectures based on 16-bit array multipliers in a 130-nm general-purpose CMOS. The baseline version, consists of 32 input D flip-flops and an array multiplier, which operates at the maximum clock frequency ($F_{CLK,BASE}$) at each V_{DD} . The two-stage pipeline architecture, which consists of 32 input and pipeline flip-flops. The two-stage pipeline can halve the critical path delay, which allows us to use the same clock frequency ($F_{CLK,PIPE} = F_{CLK,BASE}$) at V_{DD} . This low V_{DD} can improve energy-efficiency. Finally, The two-way parallel architecture. This design includes a 32-bit 2-to-1 multiplexer to recombine the outputs of the two multipliers (Multipliers 1 and 2). In the parallel architecture, while a new input comes at $F_{CLK,BASE}$, computation is interleaved by

clocking the input flop-flops at $F_{CLK, PARA}$, which is the half of $F_{CLK, BASE}$. Although clock frequency is reduced, throughput is still maintained. This slack, provided by parallelism, enables us to reduce V_{DD} to increase power and energy savings. The energy dissipation of output flip-flops is not included.

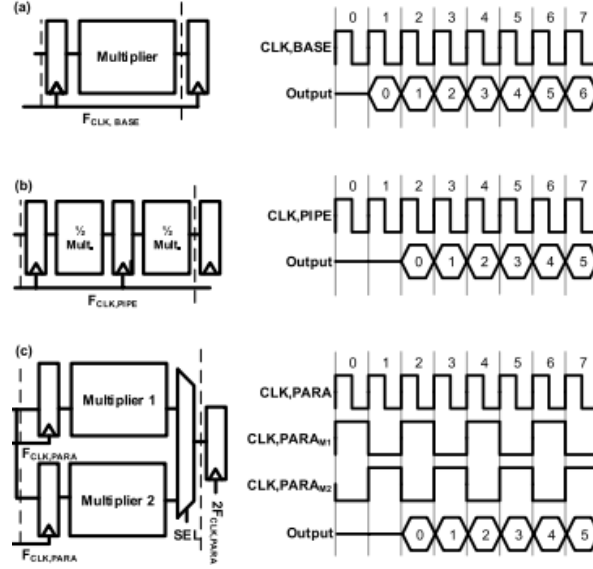


Fig. 4.3.1. Three test architectures based on a 16-bit multiplier. (a) Baseline, (b) two-stage pipelined, and (c) two-way parallel designs. Dashed lines: boundaries of the equivalent sequencing stage across three designs.

4.4 Mode Transition Overhead of PGS

The conventional nMOS-based PGS design, and the transient behaviors of the virtual ground potential (V_{VG}), and power dissipation of the main circuits when entering, exercising, and exiting sleep modes. Here, when the SLEEP BAR (SLPB) signal transitions to logic level LOW (i.e., entering sleep mode), the potential of VG starts to rise to a level close to V_{DD} . The elapsed time associated with this transition is defined as time to sleep (T_{2SLP}). After that, the circuits reach deep sleep, when they consume small leakage power, referred to as P_{SLEEP} . In order to exit a sleep mode, SLPB signal is set HIGH, and each node of the main circuits, including VG, returns to its stable state. The transition time from sleep to active mode is defined as wake-up time (T_{2WKU}). The total sleep time, T_{SLP} , is defined as the sum of T_{2SLP} , T_{SLEEP} , and T_{2WKU} . The energy dissipated during a mode transition is defined as E_{TRAN} .

CHAPTER – 5

5.1 LCG (LINEAR CONGRUENTIAL GENERATOR)

The most popular method for producing random numbers is a technique known as the linear congruential generator. Lehmer presented the concept in accordance with the subsequent equation in

$$X_{n+1} = (aX_n + c) \bmod m$$

The modulus is m , the multiplier is a , and the increment is c . To avoid duplication of similar values before m , the parameters a , c , and m must be carefully chosen together. The Park and Mill operator said that selecting $c=0$ will result in a respectable result. The multiplier a will be a value in the range $2, 3, \dots, m-1$, and the modulus m should be a very large prime number. LCG's cycle length will never exceed modulus m , however under the following three conditions, it frequently does:

- c is generally prime to modulus m ,
- multiplier $a-1$ is a multiple of each dividing modulus m ,
- multiplier $a-1$ is a multiple of four when modulus m is a multiple of four

LCG circuits are commonly used in VLSI for encryption, simulation, and testing applications. They can be implemented using a small amount of hardware, making them attractive for use in embedded systems with limited resources.

The following are the different types of gates used in LCG:

- **Multiplicative gate:** This type of gate multiplies the current state by a constant factor, which is typically a prime number. The result is then modulo reduced by a large prime number to ensure that the output falls within a certain range. The formula for the multiplicative gate is:

$$x_{n+1} = (a * x_n) \% m$$

where x_n is the current state, a is the multiplier constant, and m is the modulus.

- **Additive gate:** This type of gate adds a constant value to the current state to generate the next pseudo-random number. The formula for the additive gate is:

$$x_{n+1} = (x_n + c) \% m$$

where x_n is the current state, c is the constant value, and m is the modulus.

- **XOR gate:** This type of gate performs an exclusive OR operation on the current state and a constant value to generate the next pseudo-random number. The formula for the XOR gate is:

$$x_{n+1} = (x_n \text{ XOR } k) \% m$$

where x_n is the current state, k is the constant value, and m is the modulus.

- Shift register gate: This type of gate performs a bitwise shift operation on the current state to generate the next pseudo-random number. The formula for the shift register gate is:

$$x_{n+1} = (x_n \ll s) \% m$$

where x_n is the current state, s is the number of bits to shift, and m is the modulus.

5.2 LCG Circuit Diagram

The Block schematic of LCG operation overall is shown in Fig (seed is disregarded). It requires blocks for multipliers, adders, comparators, and subtractors. The use of a multiplier multiplies the preceding random number, X , with a before adding an increase of c . The modulus m is then contrasted with it after that. If the number is less than m or equal to m , it is regarded as a random number. In either case, m was used to subtract the integer. The result is then taken into consideration as a random number.

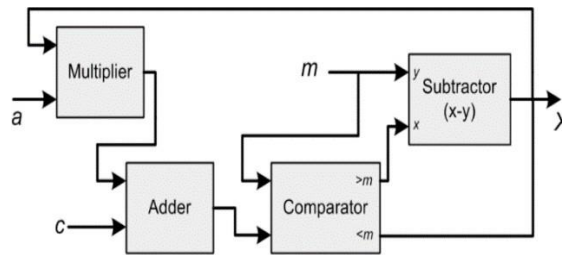


Figure 5.21.: Block diagram of LCG

The block diagram includes arithmetic operations like multiplication, addition, subtraction and comparison. Then subtractor and comparator blocks can be eliminated.

A multiplexer, a multiplier, an adder, a buffer (needed clear), and three buffers make up the circuit that was designed (required empower). The circuit's beginning value, multiplier, and increment are sent through ports Seed, A, and C, respectively. Two signals, enable and reset, limit the circuit. To clear the previous values in the buffer B4, signal reset must initially be HIGH (enable=LOW). Decide when to start the procedure using the signal enable. Before enable goes HIGH (reset=LOW), the pre-defined value (seed), increment, and multiplier must be present at the input ports. Once the clock reached HIGH, a random number was generated. To evaluate

power, delay, and PDP, LCG, which contains a D-flip flop, multiplier, adder, and mux, will be developed in code utilizing CMOS technology using MOSFETs under 130nm, 32nm, and CNTFETs under 32nm.

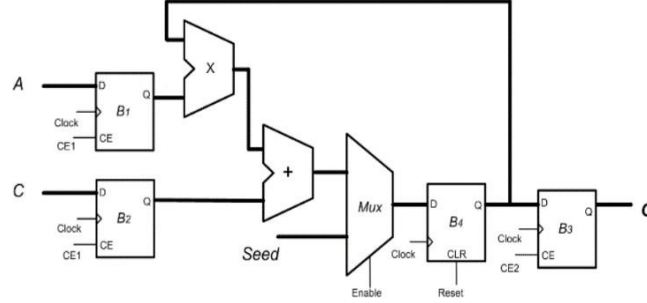


Figure 5.22: General circuit diagram of LCG

The LCG circuit diagram has DFF, multiplier, adder, and mux. To measure the power, delay, and PDP, first, we have to write the code for each device in the LCG block diagram in HSPICE. Coming to the Multiplier, we will write the nodes for each transistor as D=2, CK=3, and CKbar=4. After giving the nodes we will then code on the respective diode and stimulations are done in the HSPICE software respectively. After stimulation, we will see the power which is nearly 0.00158% and 0.0005293% which is high other than other multipliers in MOSFET under 32nm, 130nm, and 0.00001037% in CNTFET under 32nm. And the delay we get is 563.86p, 499.84p in MOSFET under 32nm, 130nm, and 0.077p in CNTFET under 32nm. Finally, the product of power and delay will be the PDP (Power-Delay product). The capacitance of a diode is directly proportional to its size. This enhances its speed in turn.

5.3 Parallel Architecture With The Temporarily Fine-Grained Sleep Technique

The design of parallel architectures in near- and subthreshold circuits, in which we employ the proposed temporarily fine-grained sleep technique. The goal is to wholly transform the throughput improvement of parallel architectures into energy savings without being penalized by active-leakage overhead. The test circuits that we use in the experiment. It consists of two 16-bit array multipliers (M1 and M2), a 32-bit 2-to-1 multiplexer, input flip-flops (IFF1 and IFF2), and sleep flip-flops (SFF1 and SFF2). M1 and M2 employ the ZSCCMOS technique with no gate overdrive voltage. The multiplexer employs no PGS. CLK is a synchronous clock signal for nonparallel portions of the circuit, while CLK1 is a 2× slower clock. CLK2 is the

inverted version of CLK1. The data input (INPUT) to the test circuits is accompanied by a VALID signal indicating whether INPUT is valid. Both INPUT and VALID signals are synchronized with CLK. In addition, by AND-ing VALID and CLK1 (or CLK2), we can generate LCLK1 (or LCLK2), which is used in the input flip-flops of M1 (or M2) for clock gating. The wake-up sequence of the circuits starts when the flipflops capture INPUT and VALID signals at the rising clock edge. However, we used the different flop-flops for INPUT and VALID signals: the flops for INPUT have reset input, and therefore, the D–Q delay of these flip-flops is longer than those of the flip-flops used for the VALID signals. The small difference in D–Q delay (~ 1 FO4) is found to be sufficient to wake up circuits because of the fast wake-up time of the ZSCCMOS. Even if the time difference is not fully sufficient and thus the INPUT signals propagate through gates, the impact on the wake-up time is found to be small since only the first few gates switch while waking up the whole circuits.

ZSCCMOS has a notable issue, which has limited its use in superthreshold voltage circuits. The matter in question is the sneak gate leakage path created by the use of both footer and header PGS.

PSLP comparisons across the four structures. PSLEEP is defined as the average power dissipation, while we sleep circuits for ~ 1000 FO4 delays, and then, wake them up. The ZSCCMOS design consumes less P_{SLEEP} than the footer PGS design by $1.74\times$ at 0.55 V and by $1.37\times$ at 0.35 V because of the smaller E_{TRAN} . The ZSCCMOS design with gate overdrive voltage consumes slightly more than Fig. 12. Comparisons among the four different PGS schemes. The ZSCCMOS due to slightly larger E_{TRAN} and the power dissipation of level converter circuits. , leakage power is a strong function of process and temperature. Therefore, it is important for sleep techniques to adapt to process and temperature variations and determine when to enter and exit sleep modes.

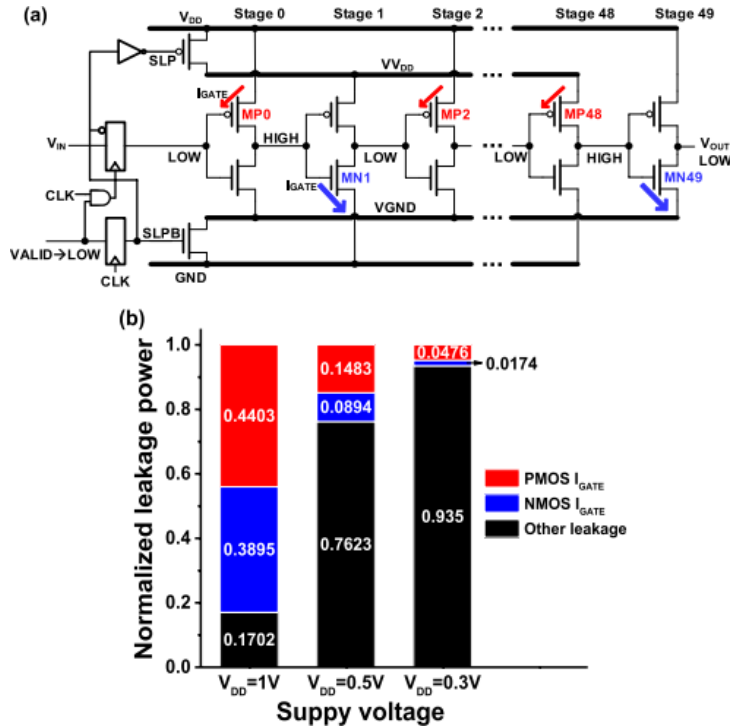


Fig.5.3.1 (a)) 50-stage inverter chain circuits with the ZSCCMOS scheme.
(b) Breakdown of leakage power dissipation during sleep modes across VDDs

5.4 Software Used

HSPICE is a widely used electronic circuit simulator software developed by Synopsys, Inc. It is used for simulating and analyzing analog and mixed-signal electronic circuits. HSPICE is one of the most popular circuit simulators used in industry and academia.

HSPICE uses a mathematical model to simulate the behavior of electronic circuits under different conditions, such as changes in input voltage or temperature. It can simulate a wide range of electronic circuits, including amplifiers, filters, oscillators, and digital circuits.

Some of the key features of HSPICE include:

- **Accuracy:** HSPICE provides highly accurate simulations of electronic circuits, taking into account the complex interactions between different components.
- **Flexibility:** HSPICE allows users to define their own models for electronic components, making it highly flexible and customizable.

- Analysis: HSPICE provides a range of analysis tools, such as DC analysis, transient analysis, AC analysis, and noise analysis, to help users understand the behavior of their circuits.
- Optimization: HSPICE includes optimization tools that enable users to find the best values for circuit parameters to meet specific performance requirements.
- Interface: HSPICE can be used with a range of different interfaces, including command line, graphical user interface, and scripting languages.

Overall, HSPICE is a powerful tool for simulating and analyzing electronic circuits, and is widely used in the design and development of electronic systems.

5.5 Implementayion of Gates in HSPICE

To implement a Gate in HSPICE software using CMOS technology,the following steps are:

- Draw the schematic of the full adder circuit using a CMOS inverter, a transmission gate, and a 2-input NAND gate. The circuit should have three inputs (A, B, and Cin) and two outputs (S and Cout).
- Specify the CMOS transistors in the circuit using the HSPICE "mosfet" command. You will need to specify the width, length, threshold voltage, mobility, oxide capacitance, and other parameters for the transistors.
- Set up the simulation parameters using the HSPICE "options" command. You will need to specify the analysis type, the simulation time, the time step size, and other options.
- Create a testbench file that defines the inputs to the circuit and sets up the simulation. You can use the HSPICE "vsource" command to define voltage sources for the inputs.
- Run the simulation using the HSPICE "run" command. The simulation will generate waveforms for the inputs and outputs of the circuit.

5.6 Netlist of Gates

Logic gates are the basic building blocks of digital circuits. They are electronic components that perform logical operations on one or more binary inputs (i.e., 0 or 1) to produce a single binary output. The output of a logic gate is determined by the

logical relationship between its input(s) and a truth table that defines the operation performed by the gate.

There are several types of logic gates, each with a different logical operation:

- NOT gate (inverter): A NOT gate has a single input and a single output. It produces a logical 1 output if its input is logical 0, and vice versa. The symbol for a NOT gate is a triangle with a circle at the output.
- AND gate: An AND gate has two or more inputs and a single output. It produces a logical 1 output only if all of its inputs are logical 1. The symbol for an AND gate is a dot (.) between the input and output lines.
- OR gate: An OR gate also has two or more inputs and a single output. It produces a logical 1 output if any of its inputs are logical 1. The symbol for an OR gate is a plus (+) between the input and output lines.
- XOR gate (exclusive OR): An XOR gate has two inputs and a single output. It produces a logical 1 output if only one of its inputs is logical 1. The symbol for an XOR gate is an o-shaped curve between the input and output lines.
- NAND gate (NOT AND): A NAND gate has two or more inputs and a single output. It produces a logical 0 output only if all of its inputs are logical 1. The symbol for a NAND gate is an AND gate with a circle at the output.
- NOR gate (NOT OR): A NOR gate also has two or more inputs and a single output. It produces a logical 0 output if any of its inputs are logical 1. The symbol for a NOR gate is an OR gate with a circle at the output.
- XNOR gate (exclusive NOR): An XNOR gate has two inputs and a single output. It produces a logical 1 output if both of its inputs are the same (both 0 or both 1). The symbol for an XNOR gate is an XOR gate with a circle at the output.

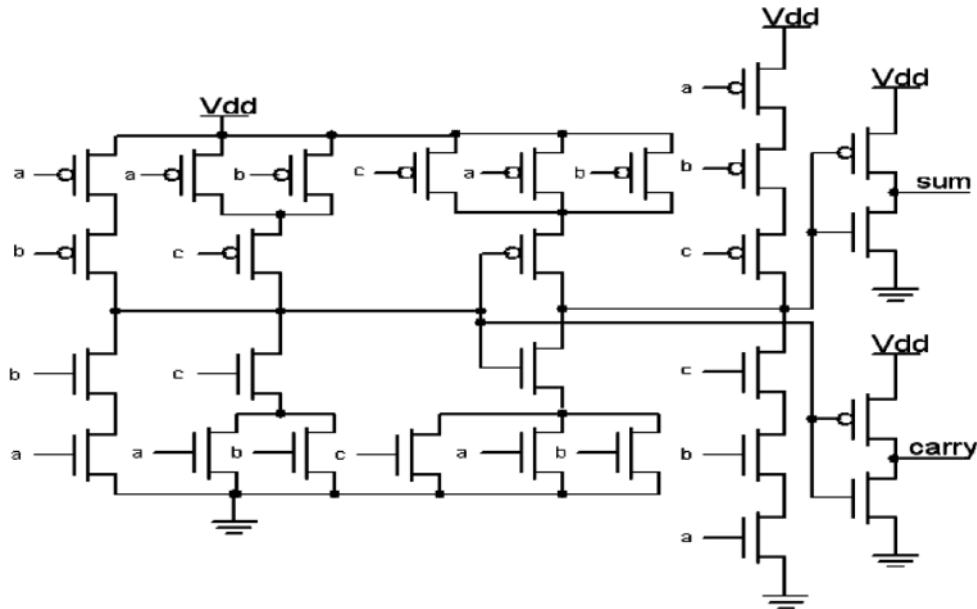
5.7 Implementation of Logic Gates

In digital electronics, gates are basic building blocks used to implement digital circuits. Gates are electronic devices that perform logical operations on input signals and produce output signals based on the rules of Boolean algebra.

There are different types of gates, such as AND gate, OR gate, NOT gate, NAND gate, NOR gate, XOR gate, and XNOR gate. Each gate has a unique truth table that specifies the output for all possible combinations of input signals.

Gates are used to implement various digital circuits, such as adders, multiplexers, demultiplexers, and flip-flops. These circuits are the basic building blocks of digital systems, such as computers, calculators, and digital controllers.

IMPLEMENTATION OF FULL ADDER



M1 - 5 3 2 0

M2 - 4 3 1 1

M3 - 5 2 3 3

M4 - 8 5 7 7

M5 - 8 7 5 5

M6 - 6 5 1 1

M7 - 9 6 7 7

M8 - 9 5 2 2

M9 - 5 4 2 2

M10- 4 3 0 0

M11- 5 2 4 4

M12- 6 5 0 0

M13- 8 6 7 7

M14- 8 7 6 6

M15- 9 5 7 7

M16- 9 6 2 2

To implement a full adder in HSPICE software using CMOS technology, you can follow these steps:

- Draw the schematic of the full adder circuit using a CMOS inverter, a transmission gate, and a 2-input NAND gate. The circuit should have three inputs (A, B, and Cin) and two outputs (S and Cout).
- Specify the CMOS transistors in the circuit using the HSPICE "mosfet" command. You will need to specify the width, length, threshold voltage,

mobility, oxide capacitance, and other parameters for the transistors.

- Set up the simulation parameters using the HSPICE "options" command. You will need to specify the analysis type, the simulation time, the time step size, and other options.
- Create a testbench file that defines the inputs to the circuit and sets up the simulation. You can use the HSPICE "vsource" command to define voltage sources for the inputs.
- Run the simulation using the HSPICE "run" command. The simulation will generate waveforms for the inputs and outputs of the circuit.

CHAPTER – 6

RESULTS AND DISCUSSION

The existing and proposed architecture are simulated by using HSPICE simulator. Output of the existing inexact speculative adder and the proposed inexact speculative adder are analysed from the results and it is concluded that the proposed architecture reduces the delay compared with the existing system.

Four-bit pipelined-carry look-ahead adder (PCLA) output simulation:

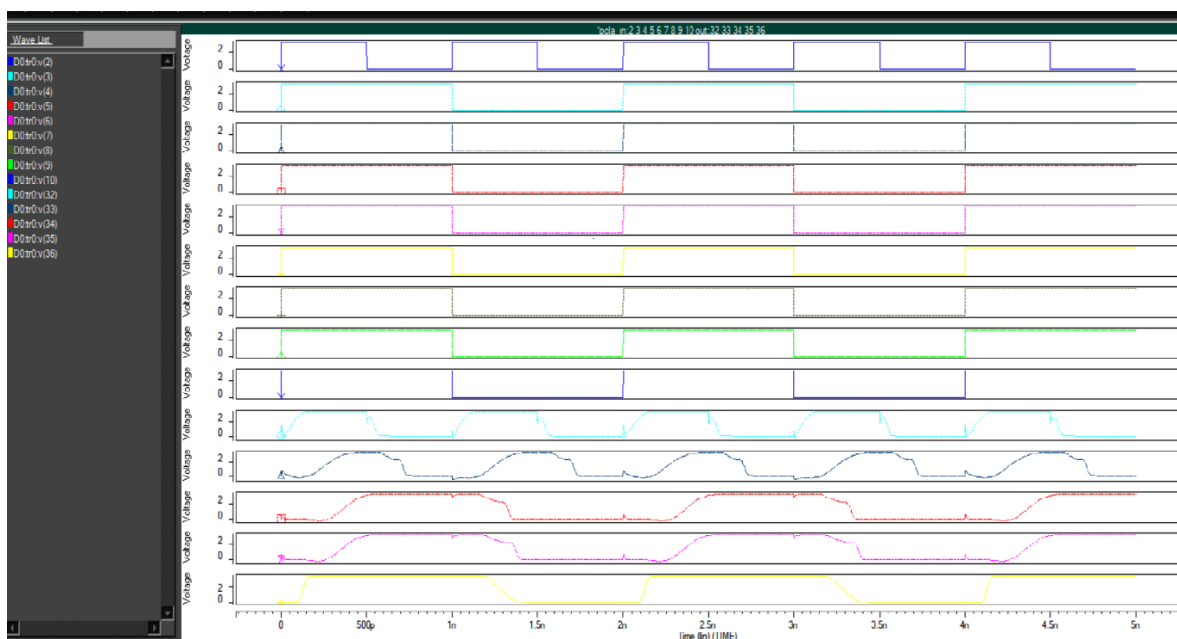


Figure:6.1 The simulation output of a four bit pipelined carry look-ahead adder.

Four-bit pipelined-carry look-ahead adder (PCLA) output using sleep technique simulation:

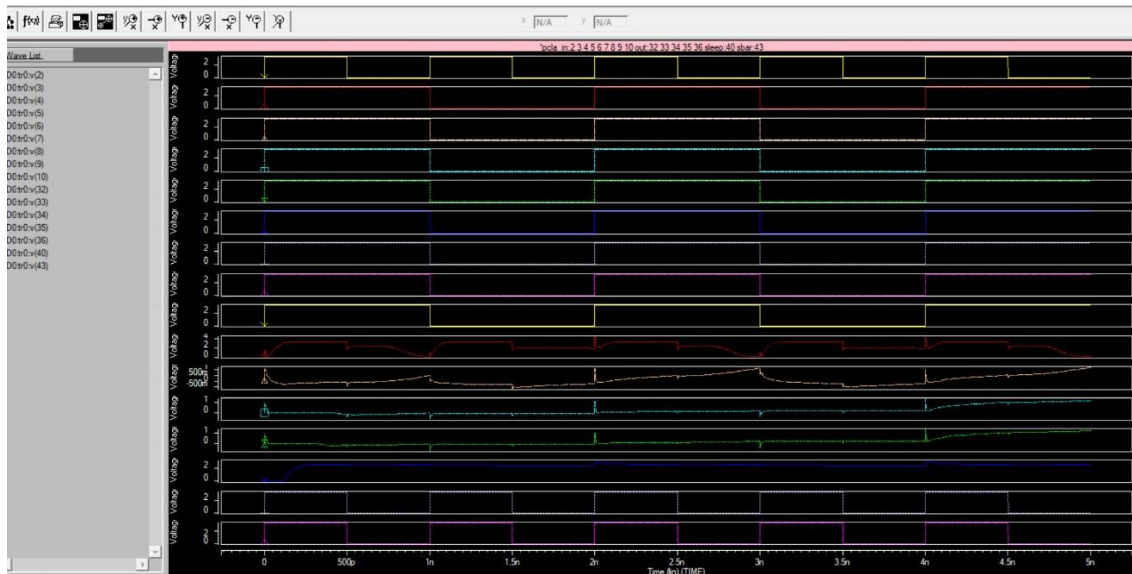


Figure:6.2 The simulation output of a four-bit pipelined carry look-ahead adder using sleep technique.

Pipelined speculator (PSPEC) output simulation:

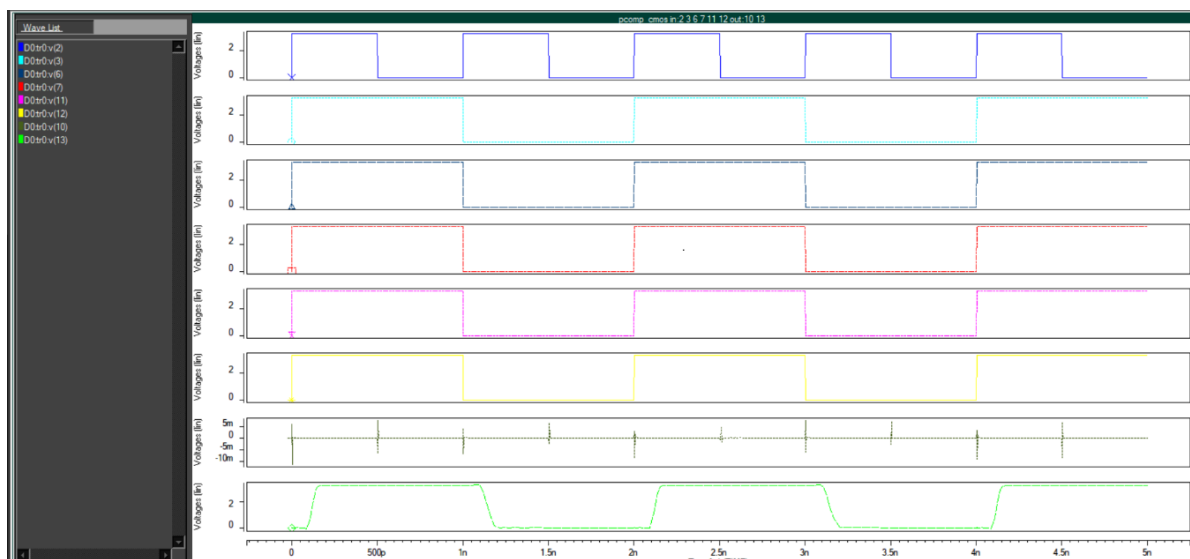


Figure: 6.3 The simulation output of a four bit pipelined carry look-ahead adder using sleep technique.

Pipelined speculator (PSPEC) output simulation using sleep technique:

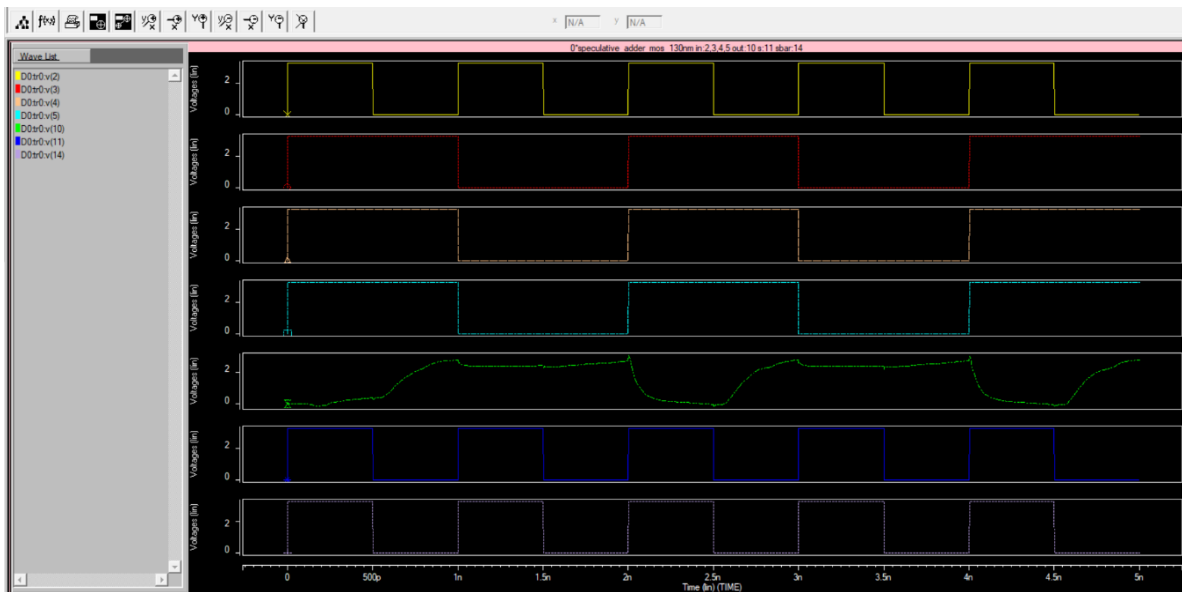


Figure:6.4 The simulation output of a pipelined speculator using sleep technique.

Pipelined compensator (PCOMP) output simulation:

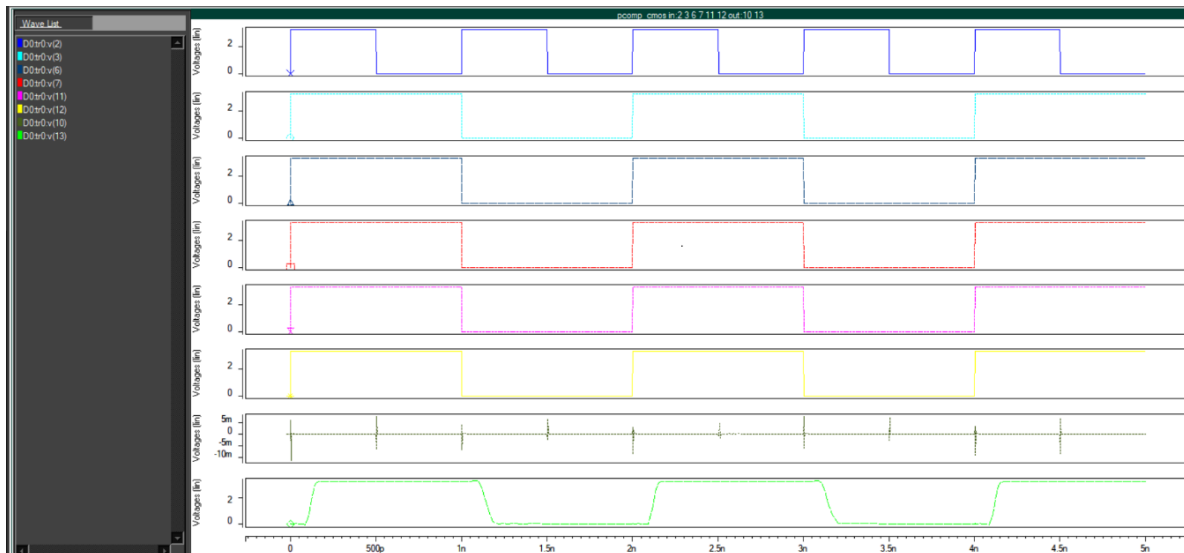


Figure: 6.5 The simulation output of pipelined compensator.

Pipelined compensator (PCOMP) output simulation using sleep technique:

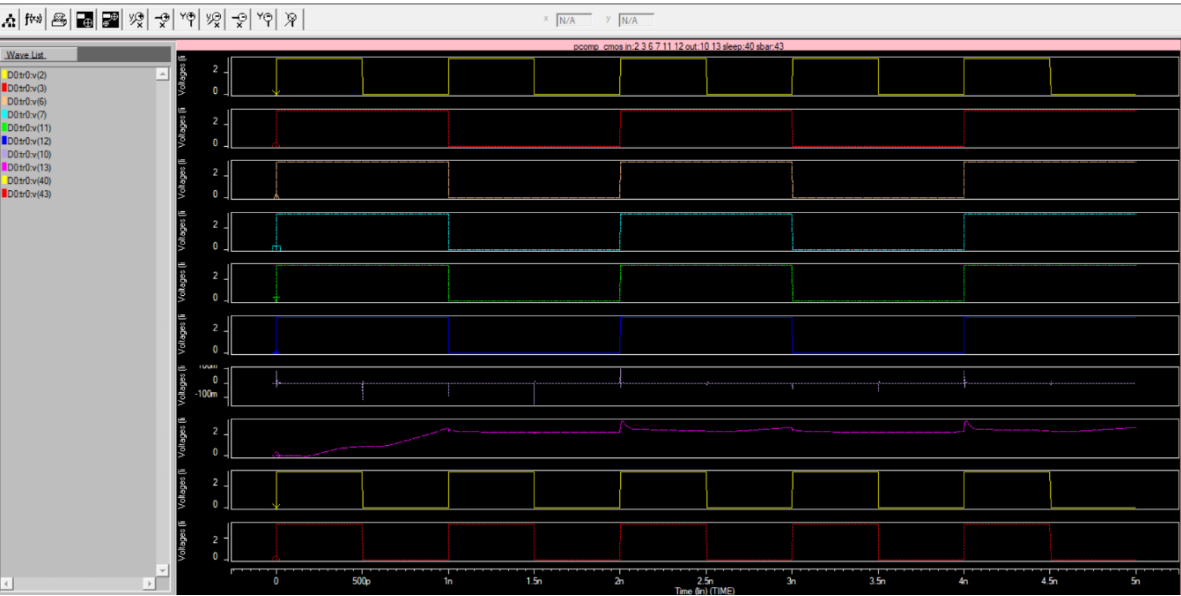


Figure 6.6 The simulation output of a pipelined compensator using sleep technique.

OUTPUT SIMULATION OF EXISTING SYSTEM:

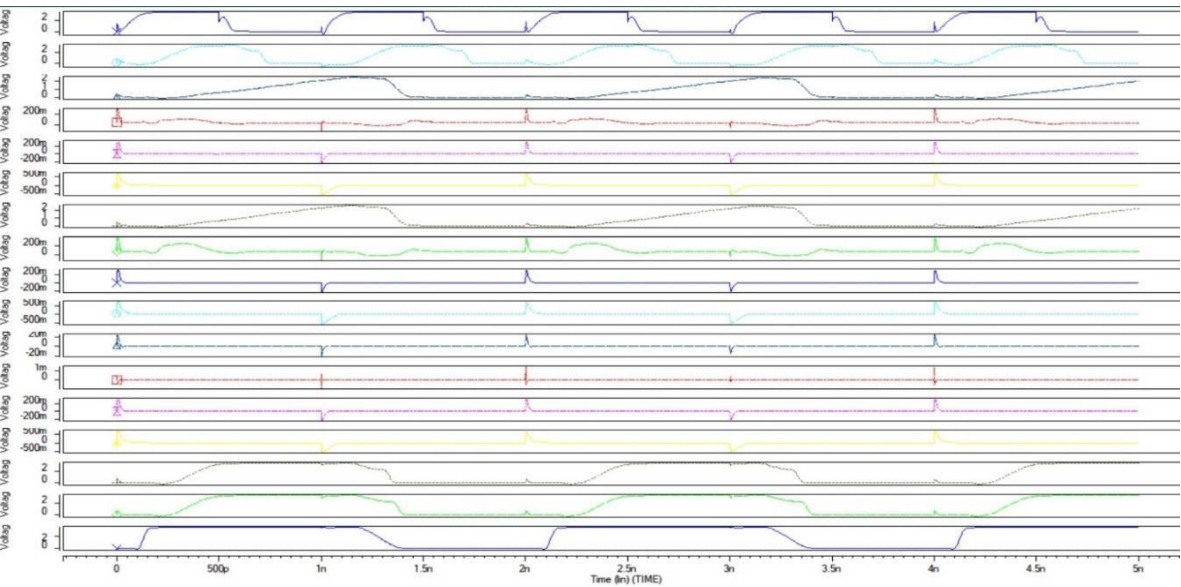


Figure: 6.7 The simulation output of a pipelined in extract speculative adder of existing system.

OUTPUT SIMULATION OF PROPOSED SYSTEM:

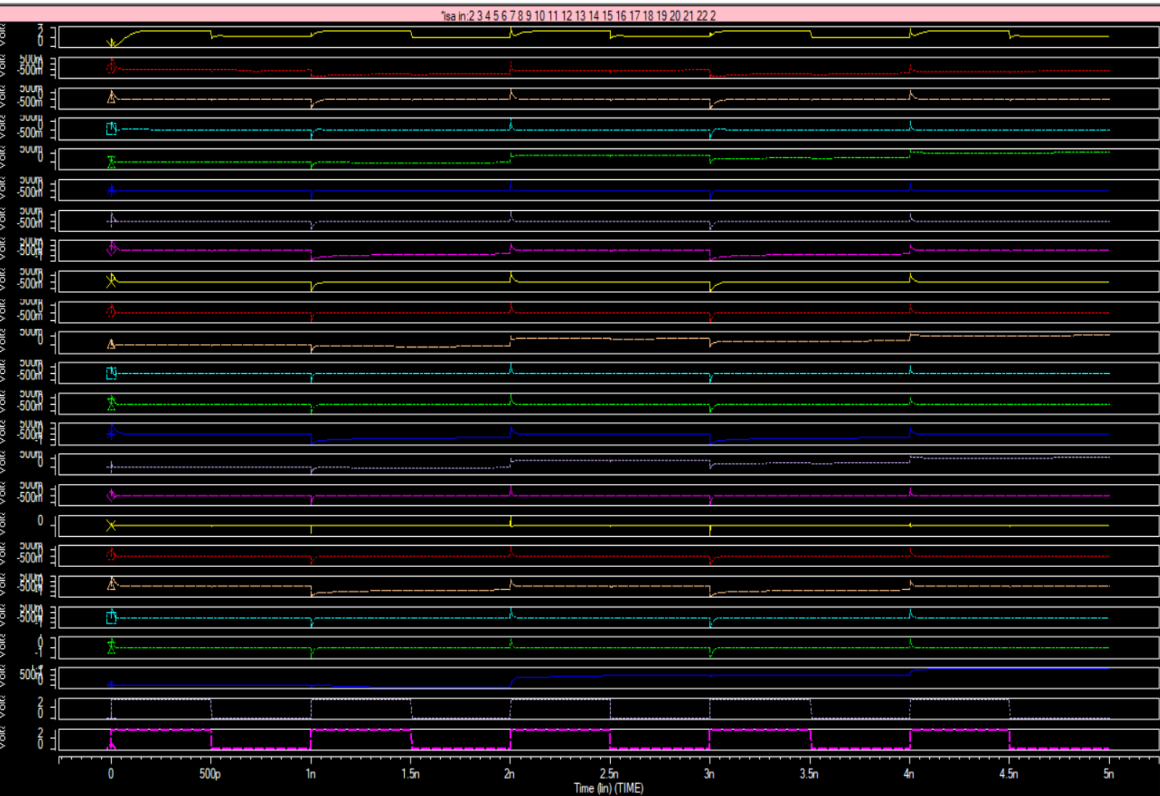


Figure:6.8 The simulation output of a pipelined inextract speculative adder of proposed system.

Four-bit pipelined-carry look-ahead adder (PCLA) output simulation using stack technique:

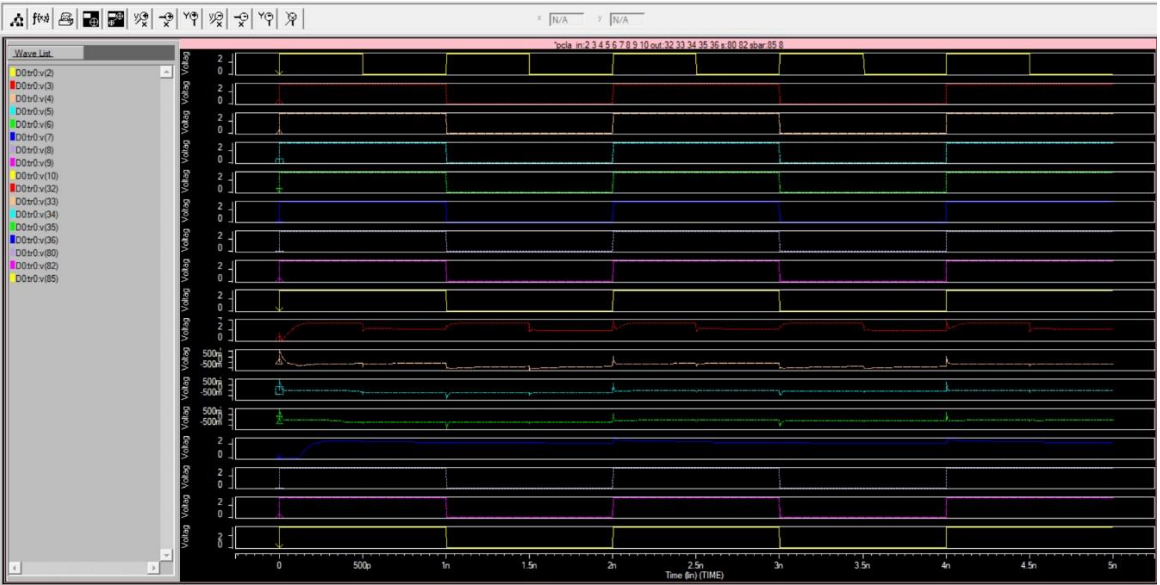


Figure:6.9 The simulation output of a Four-bit pipelined-carry look-ahead adder (PCLA) output simulation using stack technique.

Pipelined compensator (PCOMP) output using stack technique:

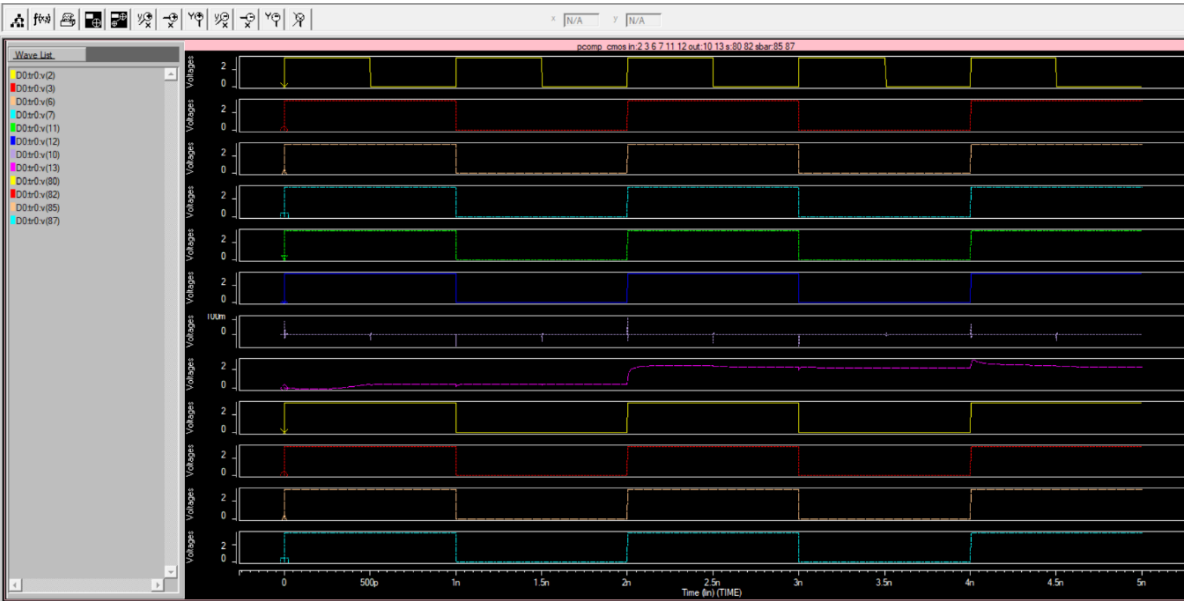


Figure: 6.10 The simulation output of Pipelined compensator (PCOMP) output using stack technique

Pipelined speculator (PSPEC) output simulation using stack technique:

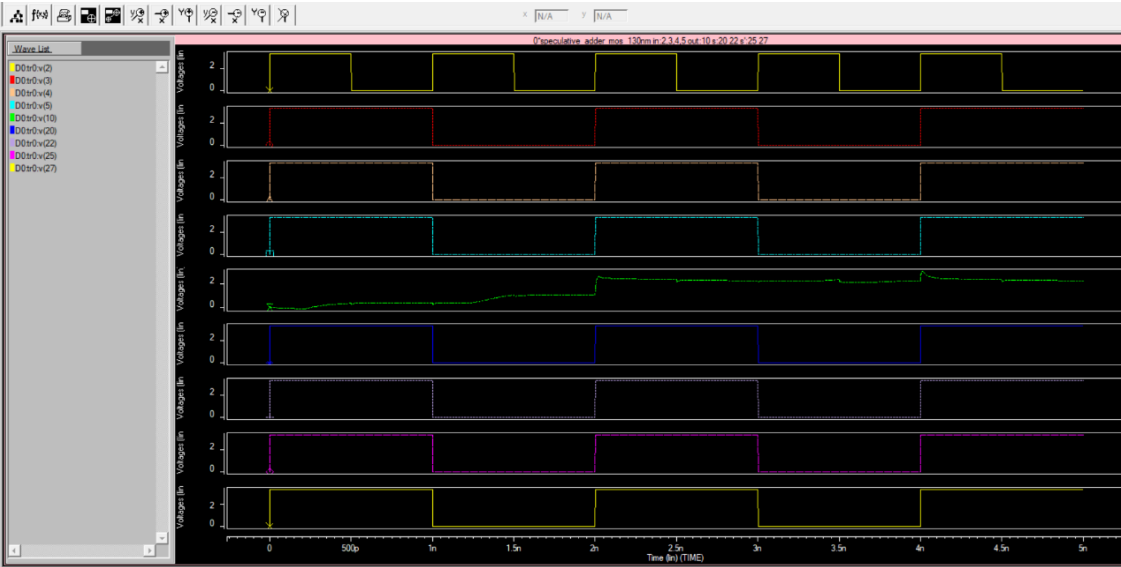


Figure:6.11The simulation output of Pipelined speculator (PSPEC) output using the stack technique.

6.1 PERFORMANCE ANALYSIS:

The performance analysis of Pipelined speculator (PSPEC), Pipelined compensator (PCOMP), and Four-bit pipelined-carry look-ahead adder (PCLA) in terms of delay has been given below.

DELAY ANALYSIS OF EXISTING SYSTEM:

Table – 6.1.1 Delay Analysis Of Existing System

DEVICE	AVERAGE POWER(W)	AVERAGE DELAY(S)	POWER DELAY PRODUCT(J)
Inexact Speculative Adder (ISA)	3.707e-04	41.546P	2.8208P
Four-bit pipelined-carry look-ahead adder (PCLA)	9.805e-05	268.45P	17.735P
Pipelined speculator (PSPEC)	1.245e-03	500.95P	31.05P
Pipelined compensator (PCOMP)	3.491e-03	1.65P	0.286P

DELAY ANALYSIS OF PROPOSED SYSTEM USING SLEEP TECHNIQUE:

Table – 6.1.2 Delay Analysis Of Proposed System Using Sleep Technique

DEVICE	AVERAGE POWER(W)	AVERAGE DELAY(S)	POWER DELAY PRODUCT(J)
Inexact Speculative Adder (ISA)	8.073E-04	17.418p	2.57p
Four-bit pipelined-carry look-ahead adder (PCLA)	7.859E-04	198.12p	28.5p
Pipelined speculator (PSPEC)	6.690E-05	341.39p	15.388p
Pipelined compensator (PCOMP)	1.193E-04	1.60p	0.0364p

DELAY ANALYSIS OF PROPOSED SYSTEM USING STACK TECHNIQUE:

Table – 6.1.3 Delay Analysis Of Proposed System Using Stack Technique.

DEVICE	AVERAGE POWER(W)	AVERAGE DELAY(S)	POWER DELAY PRODUCT(J)
Inexact Speculative Adder (ISA)	7.250e-04	1.2994p	0.175p
Four-bit pipelined-carry look-ahead adder (PCLA)	6.871e-04	149.46p	18.80p
Pipelined speculator (PSPEC)	3.484e-04	250.29p	5.875p
Pipelined compensator (PCOMP)	8.391e-05	4.229p	0.2390p

DELAY ANALYSIS OF PROPOSED SYSTEM USING SLEEPY STACK TECHNIQUE:

Table – 6.1.4 Delay Analysis Of Proposed System Using Sleepy Stack Technique.

DEVICE	AVERAGE POWER(W)	AVERAGE DELAY(S)	POWER DELAY PRODUCT(J)
Inexact Speculative Adder (ISA)	7.548e-04	7.659p	1.0588p
Four-bit pipelined-carry look-ahead adder (PCLA)	7.155e04	250.5p	32.827p
Pipelined speculator (PSPEC)	4.220e-05	498.4p	0.1417p
Pipelined compensator (PCOMP)	9.435-05	1.7199p	10.933p

CHAPTER 7

CONCLUSION

Newly proposed a new technique of in extract speculative adder using low power techniques which give better response in terms of speed, delay, average power, and area to a conventional pipelined speculative adder. The total power consumption has been reduced from $3.707e-04$ to $6.871e-04$. Total delay has been reduced from $41.546p$ to $1.2994p$ and power delay product from 2.8208 to $0.175p$. Hence we achieved our objective. So, we practically saw that inexact speculative adder with a modified low-power technique is much more efficient.

The design is to combine parallel architectures with temporarily fine-grained sleep modes for improving the energy-efficiency of near- and subthreshold digital circuits. The two classic throughput and energy-efficiency techniques, pipelining and parallelism, in the context of near and subthreshold circuits. By investigating them, we find that active-leakage overhead makes it significantly inefficient to trade off throughput improvement for energy savings. In order to mitigate active-leakage overhead, we propose a temporarily fine-grained PGS technique by analyzing and optimizing mode transition overhead. Simulations with multiplier-based test circuits show that the proposed technique can significantly reduce energy consumption per cycle, especially for the cases in which hardware utilization is low. The result also confirms the importance of suppressing active-leakage overhead for scaling the practical limit of energy efficiency in the near and subthreshold digital circuits.

Finally, while we focus on studying small-scale test circuits, i.e., multipliers, based on the results of the study, we can speculate the applicability of the proposed technique on larger scale systems. We think that the proposed technique can more benefit the hardware whose power dissipation is dominated by active leakage, particularly due to the use of a deeply scaled VDD and the use of massive parallelism.

REFERENCES

1. A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.
2. A. Hartstein and T. R. Puzak, "Optimum power/performance pipeline depth," in *Proc. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2003, pp. 117–125.
3. A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, no. 4, pp. 498–523, Apr. 1995.
4. B. H. Calhoun, F. A. Honore, and A. P. Chandrakasan, "A leakage reduction methodology for distributed MTCMOS," *IEEE J. Solid-State Circuits*, vol. 39, no. 5, pp. 818–826, May 2004.
5. J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1838–1845, Nov. 2003.
6. K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep sub micrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.
7. M. Fojtik et al., "A millimeter-scale energy-autonomous sensor system with stacked battery and solar cells," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 801–813, Mar. 2013.
8. M. S. Hrishikesh, N. P. Jouppi, K. I. Farkas, D. Burger, S. W. Keckler, and P. Sivakumar, "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," in *Proc. ACM/IEEE Int. Symp. Comput. Archit.*, May 2002, pp. 14–24.
9. M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, "Pipeline strategy for improving optimal energy efficiency in ultra-low voltage design," in *Proc. ACM/EDAC/IEEE Design Autom. Conf.*, Jun. 2011, pp. 990–995.
10. P. Larsson, "Parasitic resistance in an MOS transistor used as on-chip decoupling capacitance," *IEEE J. Solid-State Circuits*, vol. 32, no. 4, pp. 574–576, Apr. 1997.
11. A. R. Trivedi, W. Yueh, and S. Mukhopadhyay, "In situ power gating efficiency learner for fine-grained self-adaptive power gating," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 5, pp. 344–348, May 2014.

12. P. Mishra, A. N. Bhoj, and N. K. Jha, "Die-level leakage power analysis of FinFET circuits considering process variations," in Proc. IEEE Symp. Quality Electron. Design, Mar. 2010, pp. 347–355.
13. D. Jeon, M. Seok, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A superpipelined energy efficient subthreshold 240 MS/s FFT core in 65 nm CMOS," IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 23–34, Jan. 2012.
14. B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in Proc. ACM/IEEE Design Autom. Conf., Jul. 2004, pp. 868–873.
15. Y. Kim, Y. Lee, D. Sylvester, and D. Blaauw, "SLC: Split-control level converter for dense and stable wide-range voltage conversion," in Proc. Eur. Solid-State Circuits Conf., Sep. 2012, pp. 478–481.
16. A. R. Trivedi, W. Yueh, and S. Mukhopadhyay, "In situ power gating efficiency learner for fine-grained self-adaptive power gating," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 5, pp. 344–348, May 2014.
17. B. H. Calhoun, F. A. Honore, and A. P. Chandrakasan, "A leakage reduction methodology for distributed MTCMOS," IEEE J. Solid-State Circuits, vol. 39, no. 5, pp. 818–826, May 2004.
18. T. Miyazaki, T. Q. Canh, H. Kawaguchi, and T. Sakurai, "Observation of one-fifth-of-a-clock wake-up time of power-gated circuit," in Proc. IEEE Custom Integr. Circuits Conf., Oct. 2004, pp. 87–90.
19. K.-S. Min, H. Kawaguchi, and T. Sakurai, "Zigzag super cut-off CMOS (ZSCCMOS) block activation with self-adaptive voltage level controller: An alternative to clock-gating scheme in leakage dominant era," in Proc. IEEE Int. Solid-State Circuits Conf., Feb. 2003, pp. 400–502.
20. A. Hartstein and T. R. Puzak, "Optimum power/performance pipeline depth," in Proc. IEEE/ACM Int. Symp. Microarchitecture, Dec. 2003, pp. 117–125.