Multi-Threshold Asynchronous Pipeline Circuits

by

RAGHID SHREIH, B.Sc. (COMPUTER),

UNIVERSITY OF OTTAWA

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Electronics

Carleton University

Ottawa, Ontario

December, 2006

Your file Votre référence
ISBN: 978-0-494-23347-4
Our file Notre référence
ISBN: 978-0-494-23347-4

Abstract

Power consumption of integrated circuits has been rapidly increasing over the past decades, and this is expected to continue for the foreseeable future due to increasing integration and higher frequencies of operation. High power consumption results in shorter battery life and excessive heat generation, which negatively affects reliability and necessitates the design of complex cooling systems. Therefore, it is essential to develop techniques for low power design while maintaining high performance.

This thesis is focused on using multi-threshold circuit-level techniques that maintain high performance while allowing a reduction in the power supply voltage in order to reduce power consumption. These techniques are applied to asynchronous circuits, including micropipelines and GasP pipelines. The resulting circuits are then simulated and compared.

This is the first attempt, as far as we know, that combines multi-threshold and asynchronous circuit techniques. As such, multi-threshold C-Element and GasP circuit structures are proposed to take advantage of multi-threshold voltage techniques to minimize static power dissipation and to maximize performance. Applications of multi-threshold voltage techniques to delay elements and dual edge-triggered flip flops, both of which are used in asynchronous micropipelines, are also examined. Also, two application circuits are designed including custom layout. The selected application circuits are a micropipeline FIFO and a pipelined GasP 16-bit Brent Kung adder. Post-Layout extracted simulations are then run on these application circuits to prove the concept.

The proposed multi-threshold C-Element based pipelines were found to outperform standard micropipelines by achieving higher performance and lower dynamic and static power dissipation. These include new C-Element structures that were designed specifically to take advantage of multi-threshold techniques. The proposed multi-threshold GasP pipeline outperformed standard GasP pipelines as well. It also performed better than the proposed multi-threshold C-Element based pipelines.

The information used in this thesis comes in part from the research program of Dr. Maitham Shams and his associates in the VLSI group. The research results appearing in this thesis represent an integral part of the ongoing research program. All research results in this thesis including tables, graphs and figures but excluding the narrative portions of the thesis are effectively incorporated into the research program and can be used by Dr. Maitham Shams and his associates for educational and research purposes, including publication in open literature with appropriate credits.

Acknowledgements

First and foremost, all praise be to God the almighty, the Lord of the heavens and the earth and all that exists within them. This thesis would not have been possible without the guidance, patience and strength bestowed upon me by God.

Acknowledgement is due to my supervisor, Dr. Maitham Shams, who helped to shape my work and guide my research. His kindness and support were very important in allowing me to complete this thesis successfully. I would also like to acknowledge my friends and colleagues at the Department of Electronics, who kept me company and worked with me during my course work, which provided me with the foundation of knowledge on which my research was built.

My research was strongly supported by the government of Ontario through the Ministry of Training, Colleges and Universities and the Ontario Graduate Scholarship program. The Department of Electronics also supported my research and is therefore dutifully acknowledged.

This thesis would not have been possible without the constant encouragement of my parents, grandmother and siblings. Their unwavering support and confidence in my ability to succeed in this endeavor gave me the strength needed to complete this work.

Most importantly, I acknowledge and thank my wife, who supported and encouraged me and was patient while I completed this thesis. Thank you for all the days and nights you spent with me at the University while I worked on my research. You introduced meaning into my life, and so I dedicate this thesis to you.

Dedication

to my loving wife, Raghad

Table of Contents

List of Tables

List of Charts

List of Figures

List of Equations

List of Abbreviations

C-Element: An asynchronous circuit primitive and the control circuit for asynchronous micropipelines.

CMOS: Complementary Metal-Oxide Semiconductor

CMOSP13: The 0.13μm process technology used for this research

CPL: Complementary Pass Transistor Logic

DC: Direct Current

DETFF: Dual-Edge Triggered Flip-Flop

DRC: Design Rule Check. This is a check run on a circuit layout to ensure that it does not violate any of the design rules of the process technology being used.

DSSCCMOS: Dual-Sleep Super-Cutoff CMOS. This is a multi-threshold technique proposed in this thesis to be used with C-Element circuits. It allows sub-threshold leakage currents to be minimized in sleep mode while maintaining the state of the C-Element

EDA: Electronic Design Automation

FIFO: First-In First-Out. This is a structure frequently used in digital hardware design that allows processing data in stages.

GasP: A simplified asynchronous pipeline control circuit

LVS: Layout vs. Schematic. This is a check that is run once a layout is complete to verify that it matches the schematic in functionality.

Micropipelines: Asynchronous pipelines that use C-Element circuits to implement their control circuitry.

MISCCMOS: Multiplexed Inputs Super Cutoff CMOS. This improvement to the SCCMOS technique is proposed in this thesis. It uses existing transistors in the circuit as sleep control transistors to save area and improve performance

MOSFET: Metal-Oxide-Semiconductor Field Effect Transistor

MTCMOS: Multi-Threshold CMOS

Multi-Threshold Circuits: Circuits that use transistors of more than one threshold voltage value.

PT: Pass-Transistor Logic

SCCMOS: Super-Cutoff CMOS

Spectre: A circuit simulator similar to SPICE and HSPICE

VLSI: Very Large Scale Integration

VTCMOS: Variable Threshold CMOS

CHAPTER 1

Introduction

1.1 Motivation

High performance and smaller circuit area have always been important factors in building digital integrated circuits. Recent technological advancements have made it possible to put entire systems on one chip (SOC). However, as digital integrated circuits become denser, high power dissipation levels are becoming a major concern. Many new applications require devices to be portable, which inevitably means that they must be battery operated. High power dissipation levels translate into larger batteries and the need for more frequent charging, both of which are not convenient for portable applications. High power dissipation also produces more heat, which can affect the reliability and lifetime of circuits. More advanced and expensive cooling systems and packaging are required to dissipate the extra heat.

Since power dissipation in digital integrated circuits is a result of high static and dynamic current, the way to reduce power consumption is to reduce the current. This can

be done in various ways, such as lowering the power supply voltage or using high-threshold transistors that have negligible leakage currents when they are turned off. However, doing this reduces power consumption at the expense of degraded performance of the circuits. This is because digital circuits work by charging and discharging capacitances in the circuit, and a lower current takes a longer time to charge or discharge a load capacitance.

As a result, there is a need for circuit techniques that can help in alleviating this trade-off between high performance and lower power consumption. This thesis discusses such techniques that allow circuit designers to maintain high performance while significantly reducing power consumption of digital integrated circuits. In addition, applications for these techniques in asynchronous circuits are explored.

The techniques discussed in this thesis are all based on the idea of using transistors of different threshold voltage values in order to achieve the goal of high performance with low power. Transistors that have high threshold voltage values require a higher voltage at their gate terminal to turn on. They are therefore slower and conduct less current. Transistors that have a lower threshold voltage require a lower gate voltage to turn on, making them faster and giving them a higher current driving capability. Many new circuit fabrication processes now have high and low threshold transistors available for use within the same integrated circuit. This opens the door for implementing techniques that take advantage of the qualities of each type of transistor in order to achieve low power without sacrificing high performance. Several of these multi-threshold techniques were examined in this thesis, and they are applied to asynchronous circuits.

Asynchronous circuits have been proposed as an alternative to the commonly used synchronous circuits to help improve performance and reduce power dissipation. The main advantage of using an asynchronous design methodology is that the overall asynchronous system performs at the average speed of the stages within it. In synchronous systems, the performance is limited to the speed of the slowest stage, since the clock signal must be shared by all stages. Asynchronous circuits use handshaking signals instead of a system clock, which enables each stage of the overall asynchronous system to run at maximum speed, regardless of the speed of the other stages in the system.

Asynchronous circuits are well suited to a modular design philosophy [31]. Once an asynchronous component is designed, it can be placed in any asynchronous circuit that uses the same communication protocol and can be expected to work with no need for any modifications. In a synchronous system, the components must be re-designed for the new clock rate if they are to be re-used effectively. Asynchronous circuits do not suffer from clock skew, since they do not use a clock. Also, as clock rates continue to increase to meet the rising performance demands of digital integrated circuits, the clock tree is starting to consume large amounts of power. In addition, the clock tree draws a large peak current regularly as it switches, contributing to the noise generated by the digital circuit, which can have negative effects on analog parts of the chip.

Asynchronous circuits can be a better alternative to synchronous circuits for applications that require circuits that have high performance but that are only active for a small percentage of their lifetime. This is because while an asynchronous circuit remains

idle, there is no switching and power dissipation is kept at a minimum. Most synchronous circuits have a continuous overhead of power dissipation, since the clocks are constantly running regardless of whether or not the circuit is active or idle.

For these reasons, asynchronous circuit design has been chosen as an area to study in this thesis in conjunction with the low power techniques discussed earlier. A couple of styles of asynchronous circuits are studied. Low-power techniques are applied to these circuits in order to make them operate at low power supply voltages while maintaining high performance. Application circuits based on asynchronous circuits are simulated and the results are presented in the coming chapters of this thesis.

1.2 Objectives

The following is a brief summary of the main objectives of this thesis.

- Combine the advantages of multi-threshold voltage circuit techniques with asynchronous circuits
- Compare the various multi-threshold design techniques that have been proposed in literature, specifically with regard to their suitability for asynchronous circuits
- Attempt to improve some of the multi-threshold design techniques proposed in literature
- Apply multi-threshold design techniques to asynchronous circuit primitives and pipelines
- Propose improvements on asynchronous circuits to take better advantage of multi-threshold techniques
- Prove the concepts in some application circuits

1.3 Thesis Organization

Chapter 1 of this thesis provides an introduction to the thesis and describes the organization of the chapters within it.

Chapter 2 starts with a discussion of the components of power dissipation in a typical digital integrated circuit. The reasons for using a lower power supply voltage to reduce power dissipation are presented, and the challenges that this introduces are discussed. A discussion of the various sources of static leakage current is also presented. This is followed by a literature study of multi-threshold techniques proposed for reducing power dissipation. Each technique is then described in more detail. Finally, a brief discussion of on-chip charge pump circuits is presented.

Chapter 3 introduces asynchronous circuits, their applications, advantages and disadvantages. A literature study of proposed asynchronous control circuits is presented, followed by a more detailed discussion of asynchronous micropipelines and the C-Element. The GasP asynchronous control circuits is also introduced and discussed. Finally, the benefits of pipelining are briefly discussed.

In chapter 4, the multi-threshold techniques discussed in chapter 2 are applied to asynchronous pipeline primitives, including the C-Element, Dual Edge-Triggered Flip-Flop, Delay Element and GasP control circuits. Also, new multi-threshold techniques for asynchronous circuits are proposed. Schematic simulation results are presented and discussed.

While chapter 4 focused on the behavior of each primitive individually, chapter 5 examines the overall behavior of multi-threshold asynchronous pipelines. The asynchronous pipeline primitives designed in chapter 4 are put together to form complete asynchronous pipelines. This includes conventional C-Element, symmetric C-Element and GasP pipelines. Simulation results are again presented and discussed.

Chapter 6 presents the two application circuits selected to prove the research of this thesis. An 8-bit micropipeline FIFO circuit and a pipelined GasP 16-bit Brent Kung adder were designed using the primitives and asynchronous pipelines developed in chapters 4 and 5. Layout was performed on these two application circuits and post-layout simulations are carried out to provide more accurate simulation results. The results are presented.

Finally, chapter 7 provides a conclusion to this thesis and a discussion of possible future work that can be carried out based on the results obtained here.

CHAPTER 2

Low-Power Digital CMOS Circuits

2.1 Introduction

The techniques for building low power circuits that we will consider in this thesis all revolve around the threshold voltage of the transistors and the power supply voltage of the circuits. This is because lowering the power supply voltage is the most effective way to reduce the power consumption in digital CMOS circuits, for reasons that will be further explained in the next sections. However, a reduction in the power supply voltage of a digital CMOS circuit also has a negative effect on the performance of the circuit. Therefore, simply reducing the power supply voltage to reduce power consumption is not a practical method.

As will be seen in the coming sections, the reduction of the threshold voltage of the transistors in the circuit along with the power supply voltage allows circuit performance to be maintained without sacrificing the power consumption savings of reducing the

power supply voltage. This, in turn, leads to a large increase in static leakage current, which results in a significant increase in static power dissipation. The main focus of this thesis is on circuits that can operate at low power supply voltages while maintaining performance, without increasing static power dissipation significantly.

2.2 Power Dissipation in CMOS Digital Circuits

Power dissipation in CMOS logic circuits has three main components. These are the dynamic, static and short-circuit currents, as discussed in the following sections.

2.2.1 Dynamic Power Dissipation

Dynamic power dissipation occurs as a result of the current drawn to charge the output and parasitic capacitances during switching of the output. This is usually referred to as the switching current. Most of the power dissipation in a CMOS circuit arises from this switching current. When the load capacitance, C_L, of a CMOS logic circuit is charged by the pull-up network to a voltage v_{out}, the current drawn, I, is given by the following formula [2]

$$I = C_L \cdot \frac{dV}{dt} \tag{1}$$

The energy stored during the charging of the output is therefore given by

$$E_C = \int_0^{\infty} i_{VDD}(t) v_{out}\, dt = \int_0^{\infty} C_L \frac{dv_{out}}{dt} v_{out}\, dt = C_L \int_0^{V_{DD}} v_{out}\, dv_{out} = \frac{1}{2} C_L \cdot V_{DD}^2 \tag{2}$$

where V_{DD} is the power supply voltage of the circuit. Since the energy dissipation during the discharging of the output is the same, the average power dissipation over the charge/discharge cycle, P_{dyn}, is given by

$$P_{dyn} = C_L \cdot V_{dd}^2 \cdot f \tag{3}$$

This equation shows that the dynamic power consumption of a digital gate equals the load capacitance that is being charged and discharged, multiplied by the power supply voltage squared and the frequency of switching of the circuit. Frequency of switching is basically the speed at which the circuit is operating, and therefore reducing this parameter is usually not acceptable if our purpose is to lower power consumption while maintaining high speed performance. The load capacitance is usually dependent on the technology being used, because the load of a digital gate normally consists of its diffusion capacitance and the gate capacitance of the next stage in the circuit. Also, since the power supply voltage factor in the equation is squared, reducing the power supply voltage has a quadratic effect on the power consumption. Therefore, reducing the power supply voltage is the most efficient way to reduce dynamic power consumption [1].

2.2.2 Short-Circuit Current Power Dissipation

Short-Circuit current, also known as the crowbar current, is the current that is drawn when both the pull-up and pull-down networks are on at the same time. This occurs for a brief period of time during switching of the output of the circuit, but is a significant contributor to overall power dissipation. Power dissipation due to the short-circuit current, P_{SC}, is given by the following equation

$$P_{SC} = C_{sc} \cdot V_{DD}^2 \cdot f \tag{4}$$

where f is the switching activity of the gate and C_{SC} is a short-circuit capacitance that is a function of power supply voltage, transistor sizes and the input/output slope ratio [1].

Decreasing the value of V_{DD} clearly results in reducing the short-circuit power dissipation. Short-circuit current can also be minimized by sizing the PMOS and NMOS transistors in a CMOS logic circuit such that the rise and fall times are approximately equal. In this case, short-circuit current is typically expected to be less than 20% of the switching current [2].

2.2.3 Static Power Dissipation

This is power dissipation that occurs even when the inputs of the circuit are constant and there is no switching. It is due to leakage and sub-threshold currents of the transistors in a circuit. Even when the gate-source voltage, V_{GS}, of a MOS transistor is below the threshold voltage, a small sub-threshold current flows through the transistor. The transistor is said to be in the sub-threshold region, where the current flowing from drain to source, I_{DS}, is given by the following equation

$$I_{DS} = I_0 \cdot e^{\left(\frac{qV_{GS}}{nkT} - 1\right)} \tag{5}$$

where I_0 is a constant that depends on process parameters, n is a constant typically between 1 and 2, k is boltzmann constant, T is the absolute temperature in Kelvin and q is the elementary charge of an electron [2].

Clearly, the current flowing through a MOS transistor is never zero. Even in the sub-threshold region of operation, when V_{GS} is lower than the threshold voltage and the transistor is considered to be "off", there is still some current flow. Sub-threshold current is also called "weak-inversion" conduction. Strong inversion implies that there are enough carriers for conduction, but some current can still flow although it is small. In other words, the transistor starts conduction gradually even when the gate-source voltage is lower than the threshold voltage, but significant current only occurs when the gate-source voltage is higher than the threshold voltage and strong inversion occurs [1].

In digital circuits, it is usually desirable for MOS transistors to behave as closely as possible to ideal switches, where the transistors would have only two distinct states "on" and "off". Therefore, sub-threshold current is undesirable as it makes the transition between the "on" and "off" states more gradual than desired. This has made the rate of decline of the current in the region where the gate-source voltage is lower than the threshold voltage an important parameter to be considered when using MOS transistors. A slope factor is usually used to quantify this rate of decline. It is defined as the amount that the gate-source voltage needs to be reduced in order to reduce the drain current by a factor of 10, and is given by the following equation [1]

$$S = n\left(\frac{kT}{q}\right)\ln(10) \tag{6}$$

where k is the Boltzmann constant, T is the absolute temperature of operation, q is the elementary charge, and n is a factor that is dependent on the process technology. For a

typical transistor where the constant n is assumed to be $n = 1.5$, the slope factor works out to be 90 mV/decade [1]. This means that the gate-source voltage must be reduced by 90 mV in order to reduce the drain current by a factor of 10, which means there is still significant drain current even when the gate-source voltage is well below the threshold voltage. Sub-threshold current poses some challenges in digital circuit design. In addition to contributing to static power dissipation, sub-threshold current also has a negative impact on dynamic circuits that require charge to be stored on a capacitor.

Static power dissipation is particularly undesirable in circuits that typically function is bursts. An example of this type of circuit is a circuit in a cell phone that only functions during a call, and otherwise remains in a stand-by mode. If a person using this cell phone talks for one hour per day, then the circuit is expected to be operational and switching for approximately *4.2%* of the time. In other words, the circuit is in stand-by for over *95%* of the time, meaning the only current drawn is the static sub-threshold and leakage currents. Clearly, for this type of circuit, reducing the sub-threshold and leakage currents will have a major impact on the overall power dissipation of the device. This is especially important for portable devices such as a cell phone, where power dissipation is directly related to battery life. A longer battery life means that the device needs to be plugged in for charging less often, leading to a more convenient to use, and therefore better selling, product.

2.3 Effect of Lowering Power Supply Voltage on Performance

As discussed in the previous sections, the most efficient method of reducing the power dissipation of a circuit is to reduce the power supply voltage. However, simply reducing the power supply voltage to reduce power consumption is not a practical method because it also reduces the performance of the circuit. The propagation delay t_{pd} through a CMOS gate is given by [1]

$$t_{pd} = \frac{C_L \cdot V_{DD}}{I_{DS}} = \frac{C_L \cdot V_{DD}}{A(V_{dd} - V_{th})^2} \tag{7}$$

where C_L is the load capacitance of the gate, V_{DD} is the power supply voltage, I_{DS} is the drain to source current flowing through the gate and V_{th} is the threshold voltage of the transistors.

The equation for current, which appears in the denominator of the propagation delay equation, contains the term $(V_{dd} - V_{th})^2$. Clearly, as the supply voltage and threshold voltage values get closer together, the delay will increase rapidly. Therefore, it is not possible to maintain high speed performance when we reduce the power supply voltage, especially as it approaches the threshold voltage.

In order to maintain high speed performance while lowering the power supply voltage, the threshold voltage must be scaled down along with the power supply voltage. Keeping the term $(V_{dd} - V_{th})^2$ constant by reducing both the power supply voltage and the threshold voltage at the same time maintains the same delay characteristics while using a

lower power supply voltage. This results in power savings because of the lower power supply voltage while the circuit performance is not affected.

2.4 Effect of Lowering Threshold Voltage on Sub-Threshold Current

Another problem arises as the threshold voltage of the transistors used to build digital circuits is reduced. As discussed earlier in the section on static power dissipation, sub-threshold current causes static power dissipation. Reducing the threshold voltage has a very significant effect on sub-threshold current. It has been shown that for a minimum size NMOS transistor in a 0.25 μm technology, lowering the threshold voltage from 0.5 V to 0.3 V multiplies the off-current of the transistors by a factor of 170 [1].

Therefore, transistors with lower threshold voltages are more "leaky", and so they have significant static current that continues to flow through the circuit even when there is no switching activity. This static current flow causes static power dissipation as well. This must be taken into account if low threshold transistors are used in a circuit where the goal is to lower power dissipation by lowering the power supply voltage. This is because the static power dissipation introduced by the sub-threshold current of the low threshold transistors eliminates part of the power savings that resulted from the lowering of the power supply voltage. In other words, although we must reduce the threshold voltage of the transistors in order to maintain high performance when we reduce the power supply voltage, this reduction in threshold voltage causes an increase in static current which reduces the effectiveness of reducing power consumption by lowering the power supply voltage.

In order to combat this sub-threshold current problem, several techniques have been proposed in research literature on this topic. These techniques will be described in later in this chapter.

2.5 Leakage Currents in MOS Transistor Circuits

As mentioned in the previous sections, reducing the power supply voltage is the most effective method to reduce power consumption. As the power supply voltage is scaled down, the transistor threshold voltage must be scaled down as well in order to maintain the required performance levels. Leakage currents occur during both the off-state and on-state of a MOS transistor. Several mechanisms contribute to leakage current, and these will be briefly discussed in this section.

In addition to reducing power consumption, power supply voltages are also being lowered as a result of process technology down-scaling of CMOS device dimensions. Therefore, the overall trend in industry is towards using lower power supply voltages and lower threshold voltages in order to maintain high performance. As the threshold voltage of transistors is lowered, leakage currents can no longer be ignored and special circuit techniques must be used to enable us to suppress leakage current in order to keep power dissipation levels under control.

When a transistor gate-source voltage is zero, it is said to be turned off. However, a small leakage current, referred to as off-state current, still passes through the transistor. Many factors influence the amount of leakage current in a short-channel deep submicron transistor, including the threshold voltage, the physical dimensions of the channel and its doping profile, gate oxide thickness, the supply voltage and the drain and gate voltages

[4]. There are several leakage mechanisms as well, which will be discussed in the following sections.

2.5.1 pn Reverse Bias Current

The *pn* reverse bias current has two components. The first is minority carrier drift near the edge of the depletion region. In fabricating an NMOS transistor, the drain and source diffusions are doped $n+$, which means the majority carrier is electrons. The minority carriers are the holes, and the drift of the minority holes near the depletion region forms this component of *pn* reverse bias current. The second component is due to the electron-hole pair generation in the depletion region of the reverse bias junction. There can be other minor contributors to *pn* reverse bias current as well, but overall this is a minimal component of the off-state leakage current in a transistor.

2.5.2 Weak Inversion Current

A more significant contributor to off-state leakage current is the weak inversion current, which occurs between the source and drain in a MOS transistor when the gate voltage is below the threshold voltage. In this situation, the horizontal electric field across the channel is very weak, and therefore the carriers move by diffusion. The relationship between the gate voltage and the drain current is exponential in this case. This weak inversion current, also known as the sub-threshold leakage current because it occurs when V_{GS} is below the threshold voltage, is the major factor in off-state leakage current for modern devices.

2.5.3 Gate Induced Drain Leakage

Gate induced drain leakage (GIDL) is another contributor to the overall off-state leakage current in a transistor, which occurs at low gate and high drain bias voltages. This generates carriers into the substrate and drain from surface traps or band-to-band tunneling. GIDL occurs in the gate/drain overlap region. As the thin-oxide layer gets thinner, the supply voltage is increased and having a lightly doped drain all increase the GIDL current [4].

2.5.4 Drain Induced Barrier Lowering Effect

The drain induced barrier lowering effect is not a leakage current component. It is a phenomenon that results in a lowering of the threshold voltage, which in turn causes an increase in the overall leakage current. The lowering of the threshold voltage occurs when the depletion region of the drain interacts with the source near the channel surface. This causes carriers to be injected into the channel surface regardless of the gate voltage, which enhances leakage currents. Drain induced barrier lowering is more pronounced with higher drain voltages and shorter channel lengths [4].

2.5.5 Drain Punch-through

A very similar phenomenon to drain induced barrier lowering is the punch-through effect. The mechanism is very similar; however punch-through occurs deep below the gate and the channel surface. It occurs when the drain and source depletion regions electrically make contact deep in the channel. This happens when the drain-source voltage is high enough, causing the depletion region around the drain to expand and extend to the source [3].

2.5.6 Gate Oxide Tunneling

As the thin oxide layer used under the gate gets thinner with the down-scaling of process technology, gate oxide tunneling is becoming more significant. It is not a major contributor to overall leakage current, but its contribution may continue to grow with the use of thinner oxides. It occurs as a result of high electric field which causes direct tunneling through the gate. This results in a current flowing from the gate to the source or drain. It could especially be a concern for techniques where overdriving the gate voltage is required, as this increases the electric field applied through the gate voltage [3].

2.5.7 Hot Carrier Injection

Finally, hot carrier injection should be mentioned. It occurs in short channel transistors, where hot carriers (holes and electrons) are injected into the oxide. These are measured as gate and substrate currents, and increase as the effective channel length is reduced. Scaling the supply voltage down along with the channel length can help to mitigate this effect, which is another reason for trying to implement circuits that use low power supply voltages, in addition to the power consumption savings discussed earlier [4].

2.6 Low Power Supply Voltage Circuit Techniques

The goal in reducing the power supply voltage is to reduce the power dissipation of a circuit. However, reducing the power supply voltage negatively affects performance, so the threshold voltage must be scaled down along with the power supply voltage. This in turn causes a significant increase in leakage currents, which eliminates or reduces the power dissipation savings of reducing the power supply voltage. Multi-threshold

techniques can be used to achieve lower power dissipation while maintaining high performance.

2.7 Multi-Threshold Process Technology

Many of the techniques that can be used to lower leakage currents in circuits operating with low power supply and threshold voltages rely on using transistors with different threshold voltages in the same circuit. It is becoming more common in integrated circuit fabrication processes to provide for transistors of various thresholds in the same process. This gives circuit designers the flexibility to select the threshold voltage of the transistors used to optimize power dissipation and performance of their designs.

Newer fabrication technology provides for two types of NMOS transistors, a high threshold and a low threshold version. The same applies for PMOS transistors. In addition, depletion type transistors may also be provided, which have a threshold voltage of that is negative or zero. The threshold voltage can be adjusted by varying the doping concentration at the silicon insulator interface through ion implantation, or by using a different insulation material for the gate. This changes the effect of the gate voltage on the channel in the transistor, causing a change in the threshold voltage. The process technology used while performing the research leading to this thesis was a 0.13 μm process that provided for high and low threshold transistors.

When low threshold transistors are used, higher performance is expected because it takes less time for the transistors to turn on. However, leakage currents are also higher as

discussed earlier. Leakage currents are not a major problem when the circuit is active and during switching, in fact they can help switch the output faster. However, they do represent a major overhead when the circuit is in an idle or standby mode. This overhead is particularly severe for circuits that are in a standby or idle state most of the time. When high threshold transistors are used, performance is degraded but leakage currents are significantly lower than low threshold transistors. Therefore, each type of transistor is useful in a certain application in a circuit.

In general, when the circuit is active, low threshold transistors are preferred, while in standby modes of operation, high threshold transistors are preferred. If the advantages of both types of transistors can be combined in the same circuit, high performance can be maintained while standby leakage currents would be greatly reduced, keeping static power dissipation to a minimum. The remaining sections in this chapter will discuss techniques to achieve just that.

2.8 Literature Survey of Existing Techniques

A literature survey was conducted in order to collect information about existing techniques that have been proposed in literature. A brief overview of these techniques is presented here, while the following sections of this chapter will explain each technique in more detail.

One of the most obvious techniques to apply multi-threshold technology that comes to mind is to design the critical paths of a circuit using low threshold transistors in order to maintain the performance of the circuit, while using high threshold transistors for the non-critical paths in order to minimize leakage current and static power dissipation. An

algorithm for choosing the paths to implement using high and low threshold transistors is presented in [6], where a 0.5μm process technology was used to prove this concept. Simulation results of ISCAS benchmark circuits showed both active and standby leakage power savings of more than 80% for some circuits, without incurring any penalty in terms of area or performance.

Another technique presented in [7] takes advantage of the fact that the input vector to a circuit determines which transistors are turned on or off while the circuit is in standby. This in turn determines the amount of standby leakage power dissipation. The paper presents an algorithm that selects an appropriate input vector to force the circuit into a low-leakage state during standby periods. The method was demonstrated using ISCAS-89 benchmark circuits and showed that leakage power reductions of up to 54% can be achieved.

The Multi-Threshold Voltage CMOS (MTCMOS) technique is proposed in [8]. This technique uses high threshold transistors to gate the power supply when the circuit is in standby, thus minimizing standby leakage current. A PLL circuit based on a standard cell library developed in this paper was designed to demonstrate the effectiveness of this technique using a 0.5μm CMOS process. The chip consists of 5,000 gates and has dimensions of $4\times5\ mm^2$, and was operated at 1 V. Power dissipation was reduced to 5% of a similar conventional circuit operated at 5 V. Leakage current during standby mode is drastically reduced from 30 μA to below 50 nA. The MTCMOS technique is also used in [11] to implement a 1 V CMOS DSP for a mobile phone application. A 0.5μm MTCMOS process technology is used to implement the chip which demonstrates

performance that is 4 times faster than that possible using conventional CMOS technology at 1 V. The energy delay product is 1/6 of the same conventional circuit operated at 3.3 V, while standby power can be reduced to 600 nW during "sleep" mode.

One of the problems with the MTCMOS technique, which will be discussed in more detail later in this chapter, is the fact that a flip-flop implemented using the MTCMOS technique is not able to hold data when the circuit is placed in sleep mode due to the gating of the power supply voltage. A possible solution to this problem is presented in [16], which proposes intermittently cutting off and reconnecting the virtual power supply lines to the real power supply lines during sleep mode. This refreshes the virtual power supply lines and prevents them from losing their voltage levels due to leakage, which ensures that the data holding circuits powered by the virtual power supply lines continue to operate during sleep mode. Experimental MTCMOS chips using this technique were fabricated in 0.35μm MTCMOS process technology achieving 30% smaller area, 10% shorter delay time and 10% lower active power consumption compared with conventional MTCMOS technology.

One criticism of the MTCMOS technique has been that it requires the use of a CMOS process that provides transistors with low and high threshold voltages. Not all process technologies have this option available, and it is usually more expensive to have multi-threshold voltage capability in a process technology due to the extra steps required during fabrication. The Super Cut-Off CMOS (SCCMOS) scheme is proposed in [17] to address this issue. It has a very similar structure and concept to MTCMOS, but it uses only low threshold voltage transistors. The transistor used to gate the power supply

voltage during standby mode is over-driven in order to minimize leakage current. A test circuit consisting of inverters, 2-input NAND gates, flip-flops and pass transistor logic gates is fabricated using a 0.3μm CMOS process technology to demonstrate SCCMOS. It was shown that the SCCMOS circuits operate at nearly the same speed as conventional circuits with no cut-off MOSFET. At the same time, standby current was measured below 1 pA per logic gate, while active energy consumption of the 2-input NAND was shown to be 8 fJ per switching.

Finally, the Variable Threshold Voltage (VTCMOS) scheme is proposed in [24], which presents a 0.9 V 2-D discrete cosine transform core processor fabricated using a 0.3 μm CMOS technology. The VTCMOS scheme uses circuit techniques to dynamically vary the threshold voltage of the transistors in the circuit. This is done by varying the voltage applied to the bulk connection of the transistors, and can achieve active and standby power dissipation savings with minimal overhead. The circuit achieved a speed of 150MHz and power dissipation of 10 mW while operating at a power supply voltage of only 0.9 V. The power dissipation is only 2% of a previous generation of the same circuit implemented using conventional techniques and operated at a power supply voltage of 3.3V.

It is also important to note that there are other techniques that have been used to achieve low power dissipation such as using multiple supply voltages. This involves operating different parts of the chip using different power supply voltages depending upon the required performance and power dissipation. Circuits that need to have high

performance are supplied with a higher voltage, while those where the main constraint is on power dissipation can use a lower supply voltage. Although multi-V_{DD} can be a promising technique, it has several disadvantages that prompted us not to include it in this research.

Multi-V_{DD} circuits require level converters to interface between the different power supply regions. Also, multi-V_{DD} circuits require multiple power supply voltages to be available, which is not always possible. Most multi-threshold techniques use a single power supply voltage, with some exceptions that will be discussed further in the coming chapters of this thesis. Further information on multi-V_{DD} circuits and techniques can be found in literature such as in [50] and [51]. However, this type of circuit is not discussed further in this thesis because we decided to focus our research on multi-threshold techniques.

2.9 Mixed V_{th} Circuits

Digital circuits usually have a critical path which constrains the maximum speed of the circuit, while other parts of the circuit may not be timing critical at all. This intuitively introduces the idea of using low threshold transistors in the critical path of the circuit, and using high threshold transistors for non-critical parts. Although leakage currents will continuously be drawn in the critical path, this is usually a small part of the overall circuit. This simple method can lead to a large reduction in leakage currents, depending on the circuit type and the number of critical paths it contains.

This process of intelligently assigning transistors to be high or low threshold transistors can be done within a single circuit at the transistor level, or even more simply

at the gate level. Gates in the critical path of the circuit are implemented using low threshold transistors, while high threshold transistors are used for the rest of the circuit. This method reduces leakage current by using low threshold devices only where necessary to maintain performance.

However, applying this method at the gate level becomes more difficult as the circuits become more complex. Simply building all gates in the non-critical path of the circuit with high threshold transistors can lead to a change in the critical path. Therefore, more complex algorithms have been developed, such as the one proposed in [6], to determine which gates should be implemented with low threshold transistors and which ones can be safely implemented using high threshold transistors.

This method can be a simple and very effective method to cut down on stand-by leakage current in the overall circuit. At the transistor level, transistors that are not used to switch the output of the circuit have no effect on the speed performance. Hence, they can be made high threshold transistors without affecting performance. Transistors required in switching do affect performance and are therefore made low threshold transistors.

2.10 Gate Level Input Vector Leakage Current Reduction

Another method for reducing leakage current in circuits is proposed in [7]. This method is based on the fact that the leakage current dissipated in a circuit in steady state depends upon the input vector of the circuit. The input vector to a CMOS circuit determines which transistors in the pull up and pull down networks of the gate will be on or off. As a result, different input vectors produce different leakage current amounts,

depending on the paths generated in the circuit due to the input vector. Based on this observation, an algorithm was developed in [7] to determine the input vector that will put the circuit in a state where the leakage current is minimized. The algorithm takes a random sample of input vectors, and chooses the one that produces the smallest leakage current. The authors have reported leakage power reductions of up to 54% using this method.

2.11 Multi-Threshold CMOS (MTCMOS)

Multi-Threshold CMOS is a transistor level technique for combating leakage current in circuits that use low threshold transistors. MTCMOS employs transistors with both high and low threshold voltages. This does require extra fabrication steps which increases the cost of fabrication, and is not an available option in some processes. However, it provides a solution to the leakage current problem. MTCMOS can be used in specific applications that require low power and high speed operation, and where the additional cost and complexity can be justified.

The idea behind MTCMOS is to use two high threshold transistors to gate the power supply and ground connections. The rest of the circuit is implemented normally in CMOS using low threshold transistors [8]. The low threshold transistors guarantee high speed operation of the circuit at the reduced power supply voltage. The reduced supply voltage results in a significant reduction in power consumption. The logic circuit implemented using low threshold transistors is connected to virtual power supply and ground lines. Two high threshold devices are used to connect these virtual power supply and ground lines to the real power supply and ground lines.

The two high threshold transistors can effectively cut-off leakage current when they are turned off. Therefore, MTCMOS circuits can operate in two states. When the high threshold transistors, called the sleep transistors, are turned off, the circuit is said to be in sleep mode. In this mode, the power supply and ground rails are disconnected from the circuit, and therefore the circuit is not operational. However, since the sleep transistors have high threshold voltages, they are not leaky and have a very small static leakage current passing through them when they are off.

On the other hand, the circuit can be put into active mode by turning on the two sleep transistors. In this case, the virtual power and ground lines are connected to the power supply and ground through the sleep transistors since they are turned on. The virtual power line is charged up to nearly the full power supply voltage, while the ground line is discharged to nearly the ground voltage. The circuit is now powered by these virtual power supply and ground lines, and it operates normally. The circuit also does not sacrifice high speed performance because it is implemented in low threshold voltage transistors. The structure of an MTCMOS circuit is shown in Figure 2.1 applied to a conventional CMOS circuit. The transistors Q_1 and Q_2 are the sleep control transistors, and they are implemented as high threshold transistors. All other transistors in the circuit's pull-up and pull-down networks are implemented as low threshold transistors.

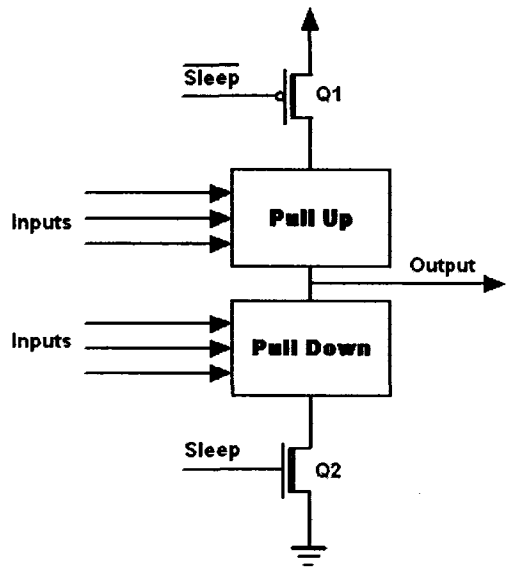

Figure 2.1: Structure of an MTCMOS Circuit

When studied closely, we realize that there are two effects that cause a reduction in stand-by leakage current in MTCMOS circuits. The first is the reduction in the effective leakage width to the width of the single transistor, and the increase in the effective channel length of the pull-up and pull-down networks of the CMOS circuit. The second effect is that of the increased threshold voltage, which results in an exponential reduction in leakage current.

MTCMOS tries to combine the speed advantage of low threshold devices with the low leakage current benefits of high threshold devices. The sleep transistors can be controlled by a simple rail-to-rail sleep voltage that puts the circuit either in sleep mode or active mode. No overdriving of the transistor gates is required in MTCMOS circuits, as will be required in some other techniques. This simplifies the circuit and eliminates the need for a charge pump to generate an overdrive voltage. However, if further leakage current reduction is required, overdriving the sleep transistors can provide additional

small reductions. So this can remain as an additional option with MTCMOS circuits. The concept of overdriving transistors will be discussed in further detail in later sections.

MTCMOS can be further optimized by noting that in fact, only one polarity sleep control transistor is actually required, either just an NMOS or just a PMOS device. Since these are high threshold voltages devices, having one sleep transistor turned off in the path between the power supply and ground may be sufficient to greatly reduce the stand-by leakage current through the circuit. In addition, having only one sleep control transistor degrades the speed of only one transition. An NMOS only sleep transistor will affect the speed of the high to low transition, while a PMOS only sleep transistor will degrade the low to high transition speed of the circuit.

Using NMOS sleep transistors is more effective due to their lower on-resistance compared to PMOS transistors. The reason for NMOS transistors having lower on-resistance is because electron mobility is larger than hole mobility, which causes more current to flow in an NMOS transistor than a similarly biased PMOS transistor. This effect can be thought of and modeled as a lower on-resistance. Therefore, NMOS transistors can be made smaller for the same current drive as compared with PMOS transistors.

However, having two transistors turned off can reduce the stand-by leakage current even further because of the stacking effect of having two transistors in series that are off in the current path. Also, since we are examining the SCCMOS technique which uses a single sleep control transistor, it was decided to use the original MTCMOS with two sleep control transistors for this research.

The MTCMOS technique does have some drawbacks. Since the circuit needs to be put in sleep mode in order to realize the stand-by current savings, special flip flops are required as no data can be held during sleep mode since the power supply and ground are disconnected from the circuit. One such flip flop is shown in the next section. Another drawback is that MTCMOS requires the introduction of at least one series device to act as a sleep control transistor. This incurs a performance penalty, which becomes worse if the device is not sized large enough.

The amount of current supplied by the sleep control transistor when it is turned on is dependent upon the device dimensions. Increasing the width of the sleep control transistor ensures that it is able to provide the required amount of current for the circuit to obtain maximum performance. Therefore, there is a trade-off between low power and high performance, and it depends upon the sizing of the sleep control transistors. [9]

Optimal sizing of the sleep control transistors can be difficult and increases design complexity. The performance penalty of the extra series transistors can only be alleviated by making them relatively large, usually much larger than the transistors used in the actual circuitry. This introduces an area overhead that must be contended with. Sleep control transistors can be shared among a large number of circuits, so for some applications this extra overhead can be acceptable in exchange for the benefits of high speed and low power circuit operation. On the other hand, sizing the sleep control transistor too small degrades performance because of the increased series resistance.

Ground bounce due to the rise of the virtual ground voltage as a result of the sleep control transistor negatively impacts performance. A new design methodology to

minimize the impact of virtual ground parasitic resistances on the performance of an MTCMOS circuit was proposed in [42]. Gate resizing and logic restructuring is used in this methodology to make the MTCMOS circuits more robust with respect to these parasitic resistance effects.

Another drawback which was mentioned earlier is that MTCMOS circuits require two different types of transistors, high and low threshold, to be fabricated on the same chip. This involves extra steps in the fabrication process and increases fabrication costs. Not all fabrication processes support multiple threshold transistors on the same chip. Also, a sleep signal is required for MTCMOS circuits, meaning that additional control circuitry is required to put the circuit in and out of sleep mode. Finally, the charging and discharging of the virtual rails costs energy, although this is usually well compensated for by the stand-by leakage, and hence static power dissipation, savings that we gain using MTCMOS.

As device dimensions get smaller with newer process technologies, deviations in leakage and delay due to process variations have increased significantly. This makes it more difficult to design MTCMOS circuits due to the variation in threshold voltage of the transistors. Two schemes for compensating for process variations are proposed in [41]. In the Variable-Gate Voltage MTCMOS (VGV-MTCMOS) scheme, the voltage applied to the gate of the sleep control transistor is varied according to the process conditions. This adjusts the resistance of the sleep control transistor, and hence the associated ground bounce. The result is an adjustment to the threshold voltage and therefore the leakage current through the circuit. The second scheme is Variable-Width MTCMOS (VW-

MTCMOS), where several sleep control transistors are placed in parallel. As the process tilts from one corner to another, the number of sleep control transistors that are turned on is adjusted to vary the current sinking capability. These schemes compensate for process variations and allow circuits to operate at the desired nominal design point.

Therefore, it is clear that MTCMOS is a very promising technique. Although it has some drawbacks which make it impractical for some applications, it can be a very suitable solution where the requirements warrant the additional cost and complexity in order to gain in power savings while maintaining high performance. MTCMOS was studied further and applied to several circuits, and the results will be discussed in later chapters.

2.12 An MTCMOS Latch

As mentioned in the previous section, one of the limitations of the MTCMOS technique is that sequential circuits lose their state during sleep mode due to the fact that the power supply and ground lines are cut-off from the circuit. A special MTCMOS latch was proposed in [8] that allows the MTCMOS technique to be applied while enabling the latch to hold data during sleep mode. The MTCMOS latch using transistors of both high and low thresholds, and has only two transistors more than a conventional latch. The MTCMOS latch is shown in Figure 2.2 below:

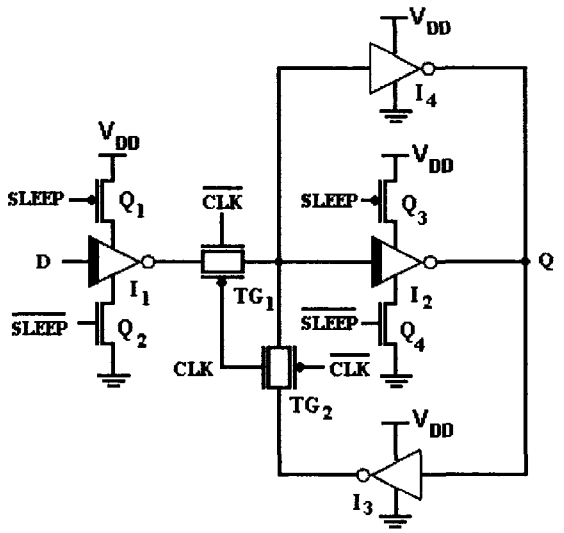

Figure 2.2: MTCMOS Latch

The MTCMOS latch structure is very similar to conventional latch structures. The latch consists of a forward path, where the data goes from the D input of the latch to the Q output through two inverters. A latching path also exists forming a loop through an inverter from the output back to the input of the second inverter. Two transmission gates select the input of the second inverter in the forward path. When the latch is transparent, the D input is connected. When the latch is holding data, the loop inverter is selected as the input of the second inverter.

What is different about the MTCMOS latch is that low threshold and high threshold transistors are used. Inverters I_1 and I_2 and both transmission gates are built using low threshold transistors. This is because these devices are in the critical path of the circuit. Inverter I_3 on the other hand has no effect on the speed of the latch, and is therefore implemented using high threshold transistors. In order to cut down on the

leakage currents flowing through the inverters I_1 and I_2, the MTCMOS technique is applied by adding high threshold sleep transistors to cut off leakage current.

In order to allow the latch to hold its value during sleep mode, a new inverter is added to the latch, I_4. This inverter forms a loop with inverter I_3, even during sleep mode, allowing the latch to hold its value even when the inverter I_2 has been cut-off from the power supply and ground. Inverter I_2 provides the high speed response, while inverter I_4 simply provides the latching required during sleep mode and therefore does not need to be fast and is implemented using high threshold transistors.

The results reported for the MTCMOS latch in [8] are quite promising. Two 1 V latches were simulated. One used all high threshold transistors, while the other was an MTCMOS latch. The MTCMOS latch was found to be twice as fast as the conventional high threshold latch. In addition, the static leakage current was found to be almost the same between the two circuits. This result shows how promising the MTCMOS technique can be in achieving high speed using low threshold transistors without noticeably increasing static leakage currents. [8]

2.13 Intermittent Power Supply Scheme

One of the problems with the MTCMOS technique mentioned in the previous section was the need for special flip-flops to preserve data when the MTCMOS circuit is placed into "sleep" mode. The flip-flop discussed in the previous section required an extra inverter as well as four sleep transistors compared to a normal flip-flop. The sleep transistors can be shared among a number of flip-flops and even with logic circuits. But

this still represents additional overhead and complexity. The Intermittent Power Supply Scheme (IPS), reported in [16], provides an alternative method. The authors reported fabricating a latch circuit in a 0.35 μm process using IPS and achieved 30% smaller area, 10% lower delay and 10% lower power consumption compared to conventional a MTCMOS latch.

The IPS idea is very simple. Instead of completely cutting off the virtual power supply lines from the real power supply lines during sleep mode, IPS connects them intermittently during this time. Because the power supply is connected to the circuit intermittently, the state of the circuit remains unchanged and so no special flip-flops or other circuitry is needed to remember the state of the various nodes in the circuit. The power supply needs to be connected frequently enough to maintain circuit state and to prevent discharge of the nodes causing the circuit state to be lost. At the same time, because the power supply is not continuously connected, the amount of leakage current flowing through the circuit during sleep mode is greatly reduced. When properly optimized, IPS allows just enough current to flow through the circuit to maintain its state.

Further consideration of the IPS technique reveals several shortcomings. Although the overhead of an extra inverter per flip-flop is eliminated using IPS compared to using the MTCMOS flip-flop discussed earlier, another type of overhead is added for IPS. A circuit, named the ACT-Gen-Block by the authors of the paper on IPS, needs to be available to generate the intermittent pulses required for the IPS scheme to work. This circuit will not be simple, and so the additional area and power overhead will be

considerable, perhaps even overshadowing any savings over the MTCMOS flip-flop latch technique presented in the previous section.

In addition, determining the appropriate frequency of the pulses, and the pulse widths is not a simple matter, and will be different for each circuit. This adds to the design complexity. Therefore, it was decided not to pursue the IPS technique any further in the research leading to this thesis. The MTCMOS latch technique that was presented in the previous section was used for MTCMOS circuits.

2.14 Super-Cutoff CMOS (SCCMOS)

Super-Cutoff CMOS, proposed in [17], is another method that can be used to combat static leakage current in CMOS circuits built with low threshold transistors. The idea behind SCCMOS is very similar to MTCMOS, which was discussed in the previous section. The main differences are that SCCMOS relies on using one sleep transistor, and that the sleep transistor in SCCMOS is implemented as a low threshold voltage transistor. This is done to reduce the fabrication steps required and therefore reduce the cost of fabricating chips using the SCCMOS technique. Figure 2.3 shows the structure of an SCCMOS circuit. Only one sleep control transistor, Q_1, is required, and this transistor is a low threshold transistor just like all other transistors in the circuit's pull-up and pull-down networks.

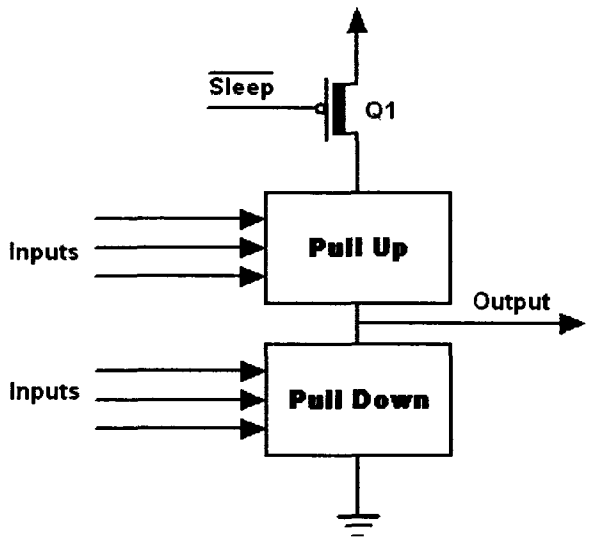

Figure 2.3: Structure of an SCCMOS Circuit

SCCMOS introduces a new challenge due to the fact that low threshold voltage transistors are leaky. A significant static leakage current will continue passing through the low threshold sleep transistor even when it is turned off. In order to overcome this problem, the SCCMOS technique requires that the gate voltage of the sleep transistor is over-driven in the case of a PMOS sleep transistor, and under-driven in the case of an NMOS sleep transistor. It is more practical and common to implement SCCMOS circuits using a PMOS sleep transistor, in which case a charge pump circuit is required to produce an overdrive voltage. The overdrive voltage is simply a voltage that is higher than the power supply voltage of the circuit. This ensures that the sleep transistor is strongly turned off, and it almost completely turns off the static leakage current passing through it [17].

Just as in MTCMOS circuits, SCCMOS circuits can be put into one of two modes. In sleep mode, the circuit no longer functions and static leakage current is suppressed,

and in active mode, the circuit operates normally and at high speed. The circuit does not sacrifice high performance because it is implemented using low threshold transistors. At the same time, it can be put into sleep mode in order to suppress static leakage current.

Like MTCMOS circuits, SCCMOS circuits can be the most appropriate choice when there is a requirement to build circuits that operate on very low supply voltages, and where circuit operation occurs in bursts. SCCMOS circuits can operate at power supply voltages that are even lower than MTCMOS circuits, because they can function at full efficiency using only one sleep transistor. This removal of one of the series sleep transistors makes it possible to lower the power supply voltage further than possible with MTCMOS circuits that use two sleep control transistors.

SCCMOS and MTCMOS circuits are particularly useful in cases of battery powered portable devices, where using a lower supply voltage and reducing power consumption is important to making battery life longer. This also applies to medical devices that are battery operated and require very long operational lifetimes. In addition, many portable devices remain idle most of the time, and only operate in bursts. An example of this is a cell phone. It remains idle most of the day, waiting for a phone call to be made or received. Therefore, applying SCCMOS or MTCMOS techniques to some of the circuits in such devices can greatly improve battery life while at the same time not compromising on high performance.

The results reported in [17] are promising. A test chip fabricated in a 0.3 μm CMOS process technology with a V_{TH} of 0.2 V was used to prove the SCCMOS

technique. The charge pump circuit required to generate the overdrive gate bias voltages was $100\ \mu m^2$ in area, and consumed $0.1\ \mu A$ at a supply voltage of $0.5\ V$. The sleep transistor was designed with a width of $10\ \mu m$. The test chip tested two input NAND and inverter gates. It was found that the SCCMOS technique achieved the same speed as using only low threshold transistors with no sleep transistor, meaning that sizing the sleep transistor large enough removes any penalty incurred from adding a series transistor. Static leakage current was found to be below 1 pA per logic gate.

When compared to MTCMOS, SCCMOS can achieve the same leakage current as MTCMOS at lower power supply voltages. The area overhead of the sleep transistor in SCCMOS is also not as large as MTCMOS since it is a low threshold device, so it does not have to be sized as large. A low threshold device needs much less area for same resistance compared to high threshold devices. However, SCCMOS retains the same drawbacks as MTCMOS regarding the area overhead of the sleep transistor, and the difficulty in optimally sizing the sleep transistor. SCCMOS circuits also require special flip flops if they are to hold information during sleep mode.

Another additional disadvantage specific to SCCMOS is the extra area-overhead of the charge pump circuit, which is required to generate the overdrive voltage to put the circuit into sleep mode. The MTCMOS technique did not require a charge pump circuit. Charge pumps will be discussed in a later section.

Overdriving the gate in SCCMOS to strongly cut-off the sleep transistor can also cause problems. As devices are scaled down and the gate-oxide continues to become

thinner, a gate-oxide reliability issue will become more problematic for SCCMOS circuits. The virtual power supply voltage of the SCCMOS circuit will drop to ground when the circuit is placed in sleep mode, as the static leakage currents flowing through the low threshold transistors will discharge the virtual power supply rails. With a large voltage at the gate, the gate-oxide is prone to breakdown due to hot carrier injection into the oxide, degrading reliability of the circuit [5].

Another effect of overdriving the gate terminal of the sleep transistor in SCCMOS circuits is the increase in Gate Induced Drain Leakage (GIDL) current. The high electric field across the gate-drain overlap region causes deep depletion, triggering the GIDL static leakage current to increase. The GIDL current can then become a significant contributor to the overall static leakage current, which degrades the benefits gained from applying the SCCMOS technique [4].

A solution is proposed to the problems of gate oxide reliability and GIDL current in [17]. A second series transistor is added with the same gate width as the sleep control transistor. During sleep mode, the gate of this second transistor is connected to the normal power supply voltage, while the gate of the sleep control transistor is connected to the overdrive voltage. The result is that both transistors are turned off, and since the device dimensions are the same, the voltage drops across each of them is equal. The result is that the gate-drain voltage of each of these transistors will be half of what it would have been in the case of using one PMOS sleep transistor. This greatly reduces the gate-oxide reliability problem and the GIDL current problem.

The disadvantage of this proposed solution, however, is that it doubles the area overhead of the sleep transistors because two transistors are now required instead of just one. In fact, since the effective transistor length of the combined sleep transistors is now doubled, the widths of the transistors must be doubled in order to maintain the same current driving capability. This means that the area overhead is quadrupled if we use two sleep transistors compared to one. This makes it more difficult to justify the area overhead for many applications.

In order for the SCCMOS technique to be used in a real application, the wake-up time of the circuit must be fast. This is especially true for applications where the circuit will be put placed into and out of sleep mode frequently. A slow wake-up time can degrade the overall performance of the circuit in such cases. A possible solution to this is proposed in [43] as an improvement to SCCMOS. The technique merges Zigzag SCCMOS (ZSCCMOS), proposed in [44], and clock gating to greatly improve the wake-up time of the circuits. The ZSCCMOS technique is applied to an FPGA Look Up Table (LUT) in [45]. A fabricated chip shows that in addition to significant leakage current reductions, the wake up time of the proposed LUT is 10 times faster than that of an LUT using SCCMOS. Therefore, it has been shown in literature that the issue of wake-up time can be addressed if it becomes a problem.

2.15 Variable Threshold CMOS (VTCMOS)

VTCMOS (Variable Threshold CMOS) takes a slightly different approach to reducing static leakage currents than the previously mentioned methods. VTCMOS takes advantage of the fact that the substrate to source voltage has an effect on the threshold

voltage of a transistor. Therefore, substrate biasing enables electrical modulation of the threshold voltage. A circuit can therefore be put in various modes, with each mode biasing the substrate differently and therefore adjusting the threshold voltage of the transistors [20].

The effect of applying a voltage at the substrate is similar to the body effect. This effect can be expressed by an equation for threshold voltage based on the empirical parameter V_{T0}, which is the threshold voltage at a source-body voltage (V_{SB}) of zero. V_{T0} is a function of the fabrication process, and depends on material constants such as the oxide thickness, Fermi voltage and the dosage of ions implanted for threshold adjustment. The expression for threshold voltage V_T is given by [1]

$$V_T = V_{T0} + \gamma(\sqrt{|(-2)\phi_F + V_{SB}|} - \sqrt{|2\phi_F|}) \quad (8)$$

In this expression, the parameter γ is called the body effect coefficient, and it represents the impact of the body-bias on the threshold voltage, while ϕ_F is the Fermi potential, which is a value that depends on physical device parameters.

Figure 2.4 shows the basic structure of a VTCMOS inverter. The same idea can be applied to more complex circuits with more complicated pull-up and pull-down networks. A simple inverter was shown in this figure simply because it's pull-up and pull-down networks consist of a single transistor. The figure shows a substrate bias control circuit, which takes the sleep control signal as an input, and outputs the appropriate p-well and n-well substrate biasing voltages. More complex schemes can be devised which

put the circuit into multiple levels of operation. This can be used to allow finer control of the speed vs. power consumption tradeoff.

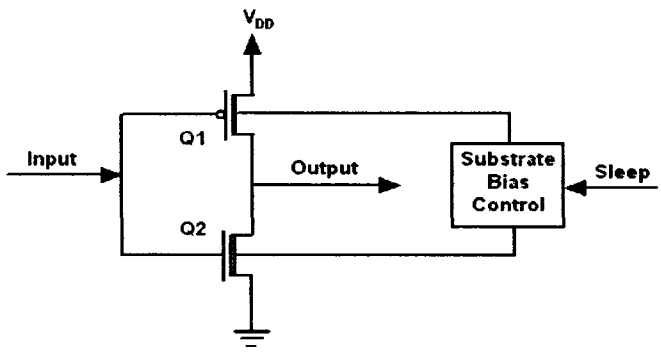

Figure 2.4: VTCMOS Inverter Circuit Structure

The concept behind the VTCMOS technique is similar to that of the Mixed-Threshold technique discussed earlier. When high speed operation is required, lower threshold transistors should be used. When the static leakage current is to be minimized, higher threshold transistors should be used. However, while the mixed-threshold technique used transistors with different thresholds in different parts of the circuit, the VTCMOS technique utilizes substrate biasing to change the threshold voltage of the transistors. The mechanism to accomplish this adjustment in the threshold voltage is explained in the following paragraphs.

Normally, for an NMOS transistor, the substrate is connected to the source terminal of the transistor. This means that the source to substrate voltage is zero. This ensures that electrons, the majority charge carrier in an NMOS transistor, are not attracted to the substrate and more of them can travel to the channel region when a positive gate voltage is applied. This makes forming a channel easier, in effect reducing

the threshold voltage of the device. If it is required to increase the threshold voltage of the transistor, the substrate can be biased at a higher voltage. This makes it more difficult to attract electrons to the channel region, requiring a higher gate voltage, and therefore the effective threshold voltage is increased.

The same theory can be applied to PMOS transistors. Normally, the substrate is connected to the source of the transistor, which is usually connected to the power supply or to the higher voltage level. This repels holes and ensures that they can easily form a channel. Reducing the bias voltage of the substrate has the effect of increasing the threshold voltage of a PMOS transistor.

It is possible to have two modes of operation for VTCMOS circuits, just like we had in MTCMOS and SCCMOS circuits. Low threshold voltage transistors are used to build the circuit. In active mode, the substrate bias is kept normal, with the NMOS transistor substrates kept low while the PMOS substrates are held high. In standby mode, a deeper reverse body bias is applied to increase the threshold voltage and reduce leakage currents.

VTCMOS circuits have the advantage that only one type of device is required, making design and fabrication simpler and less expensive. Also, no sleep control transistors are added in series to the main circuit as in the MTCMOS and SCCMOS techniques. This eliminates the area overhead of the sleep control transistors and avoids the performance penalty that is incurred if they are not sized large enough to maintain high performance. This also means that there is no performance penalty as a result of adding a series transistor. Also, since the power supply and ground are never

disconnected from the circuit, there is no need for special flip flops to hold data during sleep mode.

However, VTCMOS circuits require additional circuitry to control the substrate voltage. In addition, in many fabrication processes, it might not be easy or possible to alter the substrate pin of the transistors to connect to the substrate bias control circuitry. Layouts of existing circuits need to be completely re-done if the VTCMOS technique is to be applied, since the body terminal of every transistor must be connected to the substrate bias control circuitry. Applying the MTCMOS or SCCMOS technique does not require a new layout. It only requires the addition of the overhead circuitry without modifying the original layout. Also, as the oxide thickness decreases, the maximum applicable source-bulk potential will decrease. This will limit the range over which the threshold voltage can be modulated. VTCMOS requires triple-well structure. In addition, charging and discharging the substrate capacitance costs energy.

Another possible issue with VTCMOS is that with some new process technologies, a fluctuation of the supply voltage occurs due to IR-Drop and inductance effects. This degrades performance of the circuits and causes a large variation in the power consumption. Conventional VTCMOS techniques set the threshold voltage of the device in advance, which means that they are unable to compensate for supply voltage fluctuations. An adaptive threshold voltage control (ATVC) scheme is proposed in [46] to minimize power consumption for a given timing constraint even under conditions where supply voltage fluctuations occur.

Experimental results from papers researching the VTCMOS technique have been quite successful. In [20], the authors claim that the VTCMOS technique achieved a static power reduction of 50% with little effect on speed and minimal area overhead. Their experimental circuit consisted of a gate array in a 0.3 μm process technology. The results of [24] were even better, with the experimental 2-D Discrete Cosine Transform Core Processor circuit at 0.9 V power supply consuming only 2% of a previous similar circuit that operated at 3.3 V.

2.16 Multiplexed Inputs SCCMOS (MISCCMOS)

One of the shortcomings of SCCMOS is that it requires the addition of series transistors to gate the power supply and cut it off from the main circuit during sleep mode. This has two negative effects. The first is that the designer must size this sleep transistor appropriately depending on how many gates are drawing current through it. This can become tricky, and the transistor must be sized quite large if many gates are sharing one sleep transistor. The other more significant negative effect is on circuit performance. The addition of a series transistor degrades circuit performance and adds unnecessary capacitances that must be charged and discharged. It also limits the available headroom which means that the power supply voltage cannot be lowered as much because of the drop across the sleep transistor.

In certain applications, these negative effects can be eliminated by the usage of the Multiplexed Inputs SCCMOS (MISCCMOS) technique proposed in this thesis. Instead of connecting an overdrive voltage to the sleep transistor during sleep mode, it is applied to all or some of the PMOS transistors in the pull up network of the circuit. This eliminates

the need for adding a series sleep control transistor and the negative effects that come along with it. Simulation results presented in this thesis show that MISCCMOS can outperform conventional SCCMOS in some circuit configurations. Figure 2.5 shows how the MISCCMOS technique can be applied to a circuit with the use of a multiplexer.

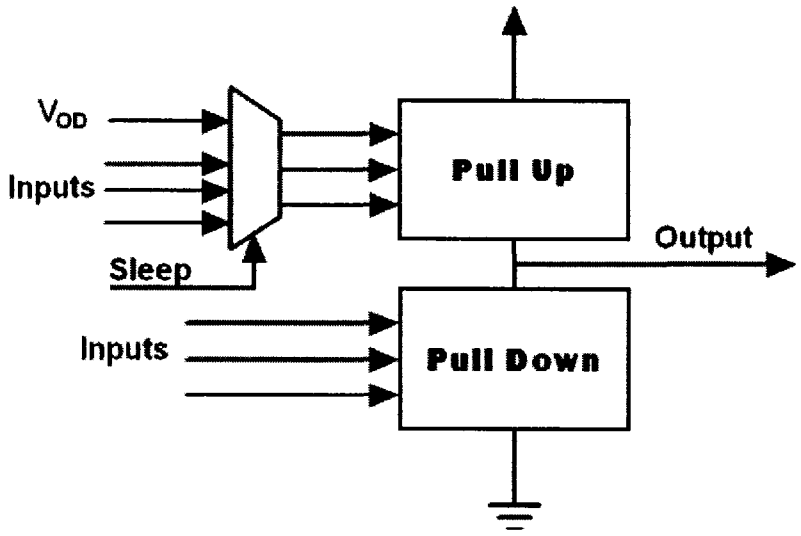

Figure 2.5: Multiplexed Inputs SCCMOS (MISCCMOS) Technique

In order to make it possible to eliminate the sleep control transistor, multiplexers are used to switch the inputs of the PMOS pull up network transistors between the overdrive voltage in sleep mode and the normal inputs of the circuit during normal mode. These multiplexers are controlled by a sleep control signal that switches the circuit between sleep and normal mode.

In the SCCMOS technique, the same sort of configuration is required. A multiplexer controlled by the sleep control signal switches the input of the sleep control transistor between the overdrive voltage and ground. Therefore, the MISCCMOS

technique does not necessarily add extra complexity or overhead. It is just used more intelligently to take out the extra series sleep control transistor. However, in cases where the PMOS pull-up network consists of several branches, one multiplexer is required per branch for the MISCCMOS technique. This may present additional overhead compared to SCCMOS, but this may be justified for certain applications where SCCMOS performance needs to be improved and the extra area overhead of the additional multiplexers is not an issue.

In circuits where the PMOS pull up network consists of multiple levels of PMOS transistors, only one level in each branch needs to be strongly cut-off to achieve the same leakage current reduction as SCCMOS. Therefore, the level with the least number of PMOS transistors in parallel, i.e. the level with the least number of branches, should be selected to minimize the number of multiplexers that will be required. It is also important to note that the multiplexers in this technique must be built using high threshold voltage transistors. Otherwise, there is potential for a leakage current path to be created between the overdrive voltage and the inputs when they are tied for ground or pulled low through the pull down network of the previous stage in the circuit.

By removing the extra series transistor used in SCCMOS, the MISCCMOS technique eliminates the performance penalty incurred by its inclusion. Another benefit of the MISCCMOS technique is that it relieves the designer of having to contend with the sizing problem of the sleep control transistor. Since the MISCCMOS technique uses existing transistors in the pull-up network in place of the sleep control transistor, these

transistors have already been designed and sized appropriately. No additional work needs to be done to size these transistors when applying the MISCCMOS technique.

The SCCMOS technique allows the sharing of one sleep control transistor among many gates, as long as the sleep control transistor is sized large enough to accommodate the current draw demand of all the gates. The result is that large sleep control transistors must be used, but this provides savings in the overhead multiplexer circuitry since only one multiplexer is required per sleep transistor. In MISCCMOS, multiplexers can be shared among inputs that are used in more than one gate. One multiplexer can switch the input between its original value and the overdrive voltage, regardless of how many gates this input is connected to. This sharing of multiplexers allows the designer to limit the circuitry overhead of the multiplexers required for MISSCMOS. However, MISCCMOS does not require large sleep control transistors like SCCMOS, so MISCCMOS can still potentially provide savings in terms of area, in addition to the improved performance it provides.

In some cases, the SCCMOS technique can be applied to existing circuits, where only the sleep control transistors and their associated overhead need to be added. The existing circuit does not need to be modified. If the performance afforded by conventional SCCMOS is acceptable for the target application, it may be more suitable than MISCCMOS in such cases. However, this depends on the circuit application, and it is not always the case that applying SCCMOS is simpler than applying MISCCMOS, even if the improved performance of MISCCMOS was not required. The MISCCMOS

technique was studied among the others already discussed in previous chapters, and the results are presented in a later chapter.

2.17 Charge Pump Circuits

Some of the techniques presented to reduce leakage currents require an overdrive voltage higher than the power supply voltage. This section provides a very brief introduction to charge pumps, which are the circuits used to provide an overdrive voltage. The research towards this thesis was focused on the circuit techniques themselves and on comparing them. However, some basic research was done on charge pumps in order to better understand how they work and how much effort and circuit area they consume when used in the low-power techniques discussed in this thesis.

Low-power techniques such as SCCMOS require an over-drive voltage that is higher than the power supply voltage in order to strongly turn off a low threshold voltage PMOS transistor. By applying a voltage higher than the power supply voltage at the gate of the PMOS transistor, the voltage difference between the gate and source terminals of the PMOS transistor can be increased which causes the transistor to be turned off more strongly and hence greatly reduces leakage current that is present when the transistor is turned off by applying a voltage equal to the power supply voltage at it's gate terminal.

The Dickson charge pump circuit can be used to generate a voltage higher than the supply voltage on chip, and this is the type of charge pump that was briefly investigated in this chapter. Charge pump circuits have been used in power IC's that require a high voltage to switch MOS transistors. They are also used on EEPROM circuits where the high voltage is used to rewrite data.

The basic concept behind charge pump circuits is shown in the circuit of the voltage doubler shown in Figure 2.6. The circuit consists of a capacitor and three switches that are controlled by a clock signal. When switches S_1 and S_3 are closed, the capacitor is charged to the power supply voltage. During the other phase of the clock cycle, switches S_1 and S_3 are open while switch S_2 is closed. The bottom plate of the capacitor now assumes a potential of V_{DD}, equal to the power supply. However, the capacitor still maintains the charge it gained of V_{DD} from the previous phase. Therefore, the voltage across the capacitor is now twice V_{DD}.

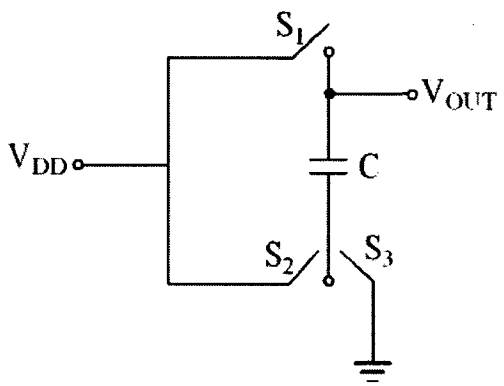

Figure 2.6: Voltage Doubler Concept

A practical Dickson charge pump circuit that can be implemented in CMOS technology was proposed in [25] and is shown in Figure 2.7. This type of circuit was simulated at the schematic level and studied briefly. This helped to develop an understanding of charge pump circuits, how they can be implemented and the area and power overhead they would introduce into a circuit. The Dickson charge pump implementation was selected because it can be implemented on-chip, using NMOS

transistors and capacitors. However, it does require the application of two clock signals that are out of phase.

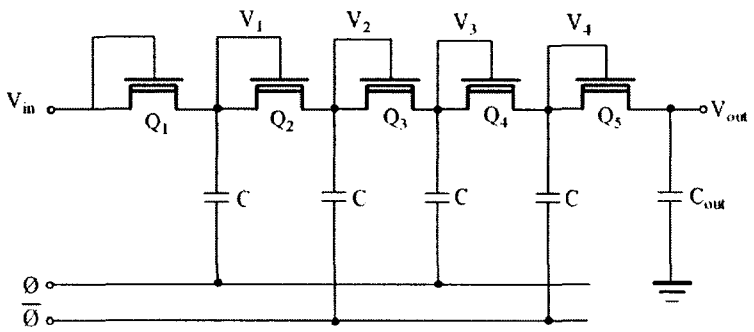

Figure 2.7: Practical Dickson Charge-Pump Schematic

The Dickson charge pump circuit shown was simulated at the schematic level and was found to work as expected. The transistors Q_1 to Q_5 are diode-connected, so they behave like diodes. This allows charge to flow in one direction only. The pumping clocks are out of phase, with amplitude of V_ϕ, which is usually the same as the supply voltage. The voltage at the node after each stage of the charge pump increases by ΔV, under the same mechanism described earlier using the simplified circuit showing a voltage doubler. This incremental increase in voltage is given by [26]

$$\Delta V = V_\phi \cdot \frac{C}{C + C_S} - \frac{I_O}{f \cdot (C + C_S)} \tag{9}$$

where V_ϕ is the amplitude of the pumping clocks, C_S is the parasitic capacitance at each node, C is the capacitance value of the capacitors at each stage, f is the pumping frequency and I_0 is the output current loading. As long as this incremental voltage

increase at each node is higher than the threshold voltage of the NMOS transistors being used as diodes, the charge pump circuit will produce an output voltage higher than the input voltage. The over-drive voltage required for SCCMOS and similar techniques does not have to be double the supply voltage. In fact, it can usually be just 0.3 V – 0.4 V higher than the power supply voltage and it will produce the expected results.

CHAPTER 3

Asynchronous Circuits

3.1 Introduction

The previous chapter introduced techniques for lowering the power supply voltage in order to reduce power consumption in circuits. In order to be able to investigate how practical these techniques are for real world applications, it was decided to apply some of them to actual circuits. Asynchronous circuits represent an interesting area of research, and they have some promising applications. Therefore, it was decided to apply the low power supply techniques to asynchronous circuits in order to be able to compare these techniques and to show that they can be applicable and beneficial in at least some applications.

However, before discussing the specifics of how to apply the low power techniques to asynchronous circuit elements, it is necessary to provide some background information about asynchronous circuits, their applications, advantages and disadvantages. It is also important to learn about the circuit elements used to implement asynchronous circuits, most importantly the C-Element. This chapter provides the motivation for choosing

asynchronous pipelines for our research and gives the necessary background information before the specific circuit implementations are presented.

3.2 Literature Study of Asynchronous Circuits

The focus of this thesis is to apply multi-threshold techniques to asynchronous pipelines. Therefore, a literature survey of asynchronous circuit concepts and the various types of asynchronous pipelines was conducted and is presented here. This section provides an overview of what is found in existing literature on these topics, while the remaining sections of this chapter provide more detailed discussions of asynchronous circuits.

A general discussion of the main advantages and disadvantages of asynchronous circuits is presented in [31]. The main concepts behind "self-timed" logic circuits are presented, including the concept of "handshaking" and the Muller C-Element, which is an instrumental circuit element in asynchronous pipelines.

One of the proposals of an asynchronous pipeline was made in [30]. In this paper, an argument is made for what is called "transition signaling", where both rising and falling edges of a control signal have the same meaning. This is in contrast to synchronous circuits where the rising edge of the clock usually signals an event. An asynchronous handshaking protocol that uses a request and acknowledge signal, in addition to the data bus is discussed. The Muller C-Element is presented, in both dynamic and static versions. The Muller C-Element, the operation of which will be discussed in more detail later in this chapter, acts as a logical AND for transition events. The C-Element circuits can then be used to implement the control circuitry for an asynchronous

micropipeline. A number of circuits including multipliers, decoders and memory controllers were designed using micropipelines. Another implementation of the C-Element, named the symmetric C-Element, was proposed in [39]. This performs the same logic function as the Muller C-Element, but has a more symmetric construction which makes it more suitable for layout.

A dynamic logic implementation is suggested for asynchronous micropipelines in [32] using a four-phase handshake protocol. This new implementation was compared with a static pipeline implementation and was found to be much faster. However, dynamic circuits are not a very good match with low power techniques, and so this implementation may not be very useful for the purposes of this thesis, as we are interested in achieving low power while maintaining high performance.

An asynchronous pipelined ripple-carry adder is presented in [33], along with a circuit that detects the completion of the addition operation. The results seem to be promising, although the micropipeline implementation presented in [30] is much simpler and there is no real purpose to implementing a complicated detection circuit.

A design methodology for asynchronous circuits is presented in [35]. A five step process is introduced that helps to design fast asynchronous circuits. The process also helps in transistor sizing which makes the design process easier. The method is based on logical effort, and it works by ensuring that the loads on each part of the circuit are equal. The ideas in this paper proved to be very helpful in the design of the asynchronous pipelines that will be discussed in later chapters.

A good comparison of several CMOS implementations of the C-Element circuit is presented in [36]. Three different implementations, including the conventional and symmetric C-Elements previously mentioned in this section, are compared. Simulations were carried out in HSPICE using a 0.8 μm BICMOS technology library, at a power supply voltage of 3V. It shows that the symmetric C-Element was the most energy-efficient implementation, achieving the same delay as the conventional C-Element using 45% less energy. A similar study is also found in [37], which uses first-order analysis in addition to SPICE simulation. The results are similar as well, finding that the symmetric C-Element is the best choice in terms of both energy and performance. Several new implementations of the C-Element are introduced in [38] and optimization methods for these circuits are discussed. Once again, the symmetric C-Element is found to be the best choice, with a frequency of 0.375 GHz achieved at a cost of 12.6 pJ in terms of energy consumption.

An MCML micropipeline implementation presented in [47], where MCML C-Element and dual-edge triggered flip flop circuits are presented. Post-layout simulation results show that an MCML FIFO implemented in a 0.18 μm process technology achieves a throughput of 4 GHz and dissipates 3.7 mW.

A very interesting approach to integrate micropipelines into FPGA circuits is presented in [48]. This is done by introducing delay pads that can be customized for each stage of the micropipeline. By making these delays reconfigurable, the doors are opened for using micropipelines in reprogrammable devices such as an FPGA. A 5-stage load-store reconfigurable machine was developed to demonstrate the validity of this approach.

The GasP asynchronous pipeline technique is presented in [40]. The idea behind GasP is to use the minimum amount of control circuitry in order to maximize speed and minimize power and area. However, GasP circuits suffer from dependence on time, which requires very careful and accurate transistor sizing to match delays through various branches in the circuit. This makes GasP pipelines more difficult to design than asynchronous micropipelines, and the dependence on time goes against the basic principle of delay insensitivity for asynchronous circuits. However, the results presented in [40] are very promising.

A simple model based on Logical Effort for calculating GasP and other asynchronous control circuits is presented in [49]. The model also enables one to estimate the energy consumption and the cycle time of the circuits independent of process technology.

3.3 Synchronous vs. Asynchronous Design Methodology

Synchronous circuits are usually defined as sequential circuits that use a clock signal for synchronization. Most digital integrated circuits designed today are therefore synchronous circuits, as they depend on a central clock signal distributed to all parts of the chip in order to synchronize all operations across the chip. The edge of the clock signal is used to determine the appropriate time when all stored variables are updated with their new values. Asynchronous circuits are sequential circuits that do not use a clock. Asynchronous design techniques have been proposed as a way of advancing digital circuits to higher density and performance. There are arguments for and against asynchronous design techniques, and some of these will be explored in this chapter.

One of the main advantages of synchronous design techniques is that they eliminate the danger of logic hazards as long as they are guaranteed to have disappeared before the next active clock edge arrives. The synchronous design paradigm has been used for many years, and is therefore widespread. Computer-aided design tools have been developed to simplify the design of synchronous circuits. Standard cell libraries have been designed and are ready to use for synchronous designs. All of these are reasons why synchronous design techniques are so dominant in current digital integrated circuits.

However, there are also some problematic issues with synchronous designs. One of these issues is that of metastability when moving data from one clock domain to another, since the data generated in one clock domain cannot be guaranteed to be valid at the edges of the clock in another clock domain. Another issue is that in synchronous design, the clock rate must be selected to be suitable to the slowest stage in the design. For a pipelined design, data passes through the stages of the pipeline from one flip-flop to the next through some combinational logic. The clock rate for the entire pipeline must be set according to the delay of the slowest stage in the pipeline. Also, process variations can cause a fluctuation in the actual delay in the fabricated circuits. The clock rate must be set to accommodate the slowest possible fabricated circuit.

On the other hand, asynchronous circuits can be designed such that the issue of metastability is not as important. This is because handshaking between stages in an asynchronous circuit can be used to ensure that there is valid data to be transferred. Also, for an asynchronous pipeline, each stage of the pipeline can operate at its own maximum speed since there is no global clock that is applied to all stages. This is usually referred to

as average-case delay, because the delay across the pipeline will be the average delay of all the stages. Synchronous pipelines, however, operate at the worst-case delay, because all stages must run at the same speed, which is that of the slowest stage.

Another problem that is being encountered more and more as integrated circuits become more complex is that of clock skew. As transistor feature sizes decrease and the density of transistors on a chip increases, the capacitive loading on the global clock signal increases. This causes a variation in the time of occurrence of the active clock edge in different parts of the chip, which is referred to as clock skew. Although computer-aided design software can optimize the layout of digital circuits such that the time delays are equal throughout the clock tree, this becomes more difficult as circuits become more complex and the clock tree design becomes inefficient. Again, this problem is not found in asynchronous circuits simply because there is no global clock required, and so there is no need for a clock tree.

Power dissipation is another significant difficulty with synchronous designs. As the complexity of integrated circuits increases, the load capacitance at each node and the number of nodes increase. Integrated circuits are operating at higher speeds than ever before, which means that the frequency of operation is also on the rise. Therefore, as integrated circuits become faster and more complex, power dissipation is increasing rapidly. This problem is complicated even further by the fact that in synchronous circuits, all nodes are required to switch at the same time with the active clock edge. This creates large peaks in the power drawn. This must be taken into account during layout, meaning that the clock lines must be increased in size to allow for the peaks in current flow and to

prevent metal migration. It was reported in [31] that for one contemporary processor design, one third of the silicon area was used by the clock drivers!

On the other hand, since asynchronous circuits do not have a clock tree, there is no problem of having a great overhead in area to accommodate it. Also, each part of an asynchronous circuit operates independently. This eliminates the peaks in current flowing through the circuit at clock edges. Therefore, power consumption does not peak at the clock edges and power consumption is evenly distributed over time in asynchronous circuits. This reduces noise generated by the digital part of the integrated circuit, which frequently causes problems for the analog parts of the integrated circuit. Furthermore, asynchronous designs will have significant savings in power consumption during periods were the circuit is not actively processing new data. In a synchronous design, there is always a clock signal that is switching at regular intervals whether or not the circuit is currently actively processing new data.

After considering all of these possible pitfalls with synchronous circuits, it can be seen that asynchronous circuits are a promising and interesting area of research. Their importance will continue to increase in the future as integrated circuits become denser and more complex. Although they do require some overhead circuitry for handshaking which replaces the global clock, asynchronous circuits can often be implemented on a smaller chip area than synchronous designs because they do not need a global clock tree.

There are other interesting features that are unique to asynchronous circuits. Asynchronous circuits are speed-independent. Any of the components in an asynchronous design can have a delay added or decreased and the overall circuit will still

perform correctly. There is no need to re-simulate the circuit or to adjust the clock period to accommodate the new delay characteristics. In fact, asynchronous circuits can be designed to be delay insensitive. This means that even wire delays are not important to the correct operation of the circuit. Therefore, the circuit will work correctly regardless of how it is routed and what the routing delays are, although performance can be optimized with proper routing techniques.

Asynchronous circuits fit very well into a modular design style. Asynchronous building blocks can be used to build a larger system. Improvements in one of the building blocks will immediately be reflected in the overall system performance. There is no need to re-design the entire circuit or re-adjust the clock period to make the system with the improved building block. All these are benefits that can be gained by following an asynchronous design methodology.

Of course, there are also some drawbacks to asynchronous design. It was earlier mentioned that synchronous design is dominant today, and so the tools, design methodology and standard cell libraries available for synchronous designs make it the easier and more logical choice for most designs. It was also mentioned that asynchronous circuits require additional synchronization signals and logic which take the place and function of the clock. This makes asynchronous circuits more difficult to design, and sometimes requires a larger chip area to implement the same function, depending on the type of circuit. It also adds overhead that may reduce the performance of asynchronous circuits compared to synchronous circuits, which reduces their attractiveness for some designs.

3.4 Asynchronous Design Concepts

Now that asynchronous circuits have been introduced and discussed in comparison with synchronous circuits, a more detailed discussion of asynchronous circuit elements and design techniques is required. One of the main concepts required to make asynchronous circuits work is that of handshaking. Instead of using a global clock signal that loads all registers in the circuit at known intervals, asynchronous circuits use handshaking, which is a process that occurs between each two stages in the circuit.

3.4.1 Handshaking

Handshaking is a process that occurs between two adjacent stages in a circuit. When the register of a stage has been loaded with a new data item, it informs the register of the next stage of the circuit that a new data item is ready using a request signal. When the next stage has read in the new data item, an acknowledge signal is sent back to the previous stage to inform it that the data item has been read, and this allows a new data item to come in and the process continues.

There are two common handshaking schemes. In the 4-cycle handshake, the request and acknowledge events are conveyed by the level of the request and acknowledge signals. For example, consider the case of all signals initially starting low. The sender initiates the handshaking process by raising the request signal. Once the receiver has read the data, it responds by raising the acknowledge signal. Upon detecting the acknowledge signal, the sender lowers the request signal. The receiver detects the lowered request signal and lowers the acknowledge signal. Thus the circuit returns to the original state where both the request and acknowledge signals are low, and this completes

the handshaking process. The process took four stages: raising the request signal, raising the acknowledge signal, lowering the request signal and finally lowering the acknowledge signal. This is why it is called a 4-cycle handshake.

The 4-cycle handshake requires four signal changes to complete, which is not really necessary. The 2-cycle handshake is more efficient, and is based on transitions as opposed to signal levels. Any transition on the request or acknowledge lines is interpreted as an event. Therefore, only two signal transitions are required as opposed to four, which saves both time and energy. Consider the same example as before, with all signals starting out low. The sender signals the request by transitioning the request signal to the high logic level. The receiver detects this request event, and responds when it is ready by transitioning the acknowledge signal to the high logic level. The handshaking process is complete. On the next cycle, both request and acknowledge signals start out at the high logic levels. They now use high to low signal transitions to generate the request and acknowledge events. This type of transition based signaling is more common because of its simplicity and the resulting savings in time and energy.

Circuit elements have been developed to perform common logic functions on transition signals. For example, an XOR gate performs the OR logic function on transition signals, since a transition on one of the signals produces a transition on the output. The Muller-C element, which will be described in more detail in the next section, performs the AND logical function on transition signals. A simple implementation of an asynchronous pipeline has been introduced in [30], and named a micropipeline. The

micropipeline uses 2-cycle handshaking and Muller-C elements to implement an asynchronous pipeline.

3.4.2 C-Element Implementations

Two implementations of the C-Element are examined in this thesis. These are the conventional C-Element and the symmetric C-Element. The idea behind the C-Element is to function as an AND gate on transition or toggle based signals. While a normal AND gate is level-sensitive, a C-Element is only sensitive to transitions on its inputs regardless of their level. Since using 2-cycle toggle based handshaking is simpler and more efficient, the C-Element is needed to perform the AND function on these signals. The C-Element produces an event at its output when both inputs have experienced an event.

The C-Element is used in asynchronous pipelines to produce an event when both the request and acknowledge signals have been triggered. Before loading a new data item into a register, it is required to wait until the next stage has read the current data item and sent back an acknowledge event. It is also required to wait until the previous stage has sent a request event informing the current stage that a new data item is available. The C-Element produces an output event only when these two input events have occurred. This output event is used as a clock input to the flip-flops of the register of the current stage of the pipeline.

The conventional and symmetric C-Element transistor level implementations are shown in figures 3.1 and 3.2.

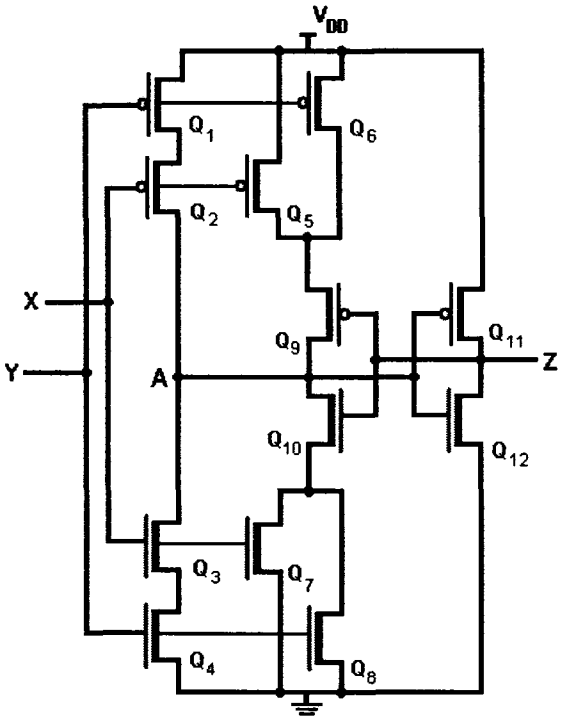

Figure 3.1: Conventional Muller C-Element

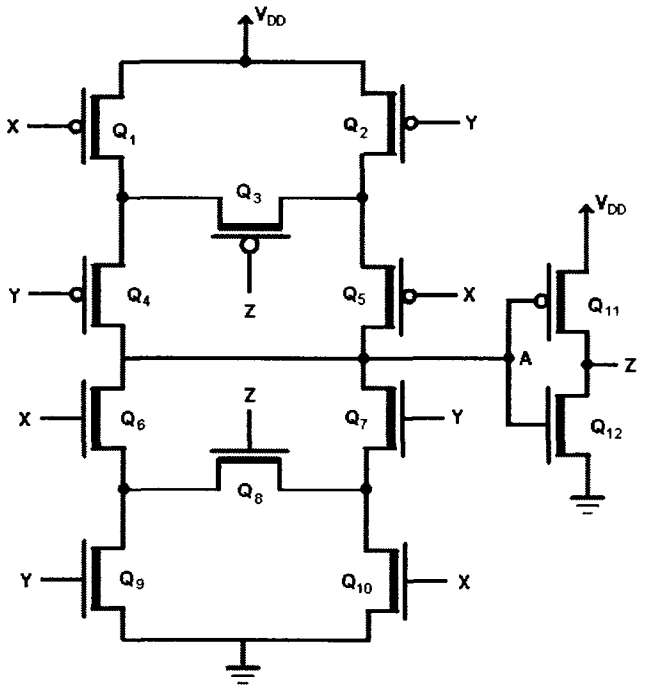

Figure 3.2: Symmetric C-Element Implementation

3.4.3 Conventional C-Element Operation

When both inputs X and Y are low, transistors Q_1 and Q_2 pull up the internal node A, which sets the output of the C-Element low. When one of these inputs changes to high, no change will occur to the output. When both inputs go high, transistors Q_3 and Q_4 pull down the internal node A, which switches the output from low to high. Therefore, the output of the C-Element does not change until both inputs have transitioned.

It is interesting to study the effect of transistors Q_5, Q_6, Q_7 and Q_8. The main reason for including these transistors is to prevent contention when the inputs are trying to overwrite the latch with a new value. These transistors will cut off inverter I_1, which allows the transistors Q_1 and Q_2 (in the pull-up case) or Q_3 and Q_4 (in the pull down case) to easily switch node A and the output without resistance from the latch. An interesting observation is that these four transistors can in fact be omitted if the designer carefully sizes the remaining transistors such that the saved value in the latch can be easily over-written without much resistance or contention. This sort of contention is not desirable because it can cause high current to be drawn during the switching, and it will make the time to switch the output longer, which slows down the C-Element. However, if the remaining transistors are sized carefully, the designer can trade-off the area used up by these transistors with a slight increase in the current drawn and a small degradation in the speed of the C-Element.

3.4.4 Symmetric C-Element Operation

The symmetric C-Element produces the same function as the conventional implementation, but it may be desirable in some cases. It works in the same way as well, except that the latching is done differently. By using the transistors Q_3 and Q_8 to latch, the geometry of the entire circuit is made more symmetric. To understand how Q_3 and Q_8 are used, consider the following scenario. If both the X and Y inputs start out low, there are two PMOS branches that pull up the internal node A, which is inverted at the output Z. Transistor Q_3 is now turned on as its gate is connected to the output Z. Now when a transition happens on one of the inputs from low to high, two of the transistors in the pull up network will be turned off. The other two will remain on, and they will be connected by transistor Q_3 which provides a path to continue pulling up node A. This latching mechanism ensures that the output does not change until a transition happens on both inputs.

Although both the conventional and symmetric implementations use the same number of transistors, the symmetry of the symmetric C-Element makes it easier to handle in the layout phase of the integrated circuit design. It may also make it easier on the layout engineer to pack the C-Elements together since they can be made to have a regular shape. Also, the performance and power consumption of the two implementations are different, and this is one of the aspects that were studied in this research. In the coming chapters, these two implementations of the C-Element are compared in terms of power and speed, and the various low power techniques are applied to both implementations as well.

3.5 Pipelining

Pipelining is a technique used in high speed circuits to optimize the performance of circuits and get the highest level of efficiency out of them. For example, consider a 64-bit adder. In a non-pipelined version of the 64-bit adder, the period of the clock running the adder must be long enough to allow the full addition operation to be completed before the next set of operands is clocked into the adder and the result is stored or passed on to the next block in the circuit. This constrains the maximum clock speed for the entire chip, resulting in decreased overall performance. Applying the pipelining technique to the adder can make more efficient use of the adder and improve the overall performance.

In a pipelined 64-bit adder, the adder is divided into multiple stages. During each clock cycle, the 64-bit operands pass through one stage of the adder. As soon as the first set of operands pass to the second stage of the adder, a second set of operands can enter the first stage. This gives an improvement in performance, the exact size of which depends on the number of stages the adder is divided into. In addition, the adder can now be run at a much faster clock rate since it is now constrained by the speed of the slowest stage of the adder and not the full adder from input to output.

It is interesting to note that the latency of the adder is actually degraded because of the addition of registers between each stage. Latency is defined as the time for a single operation to be completed. However, the throughput is greatly improved with pipelining, which is defined as the number of operations completed per unit of time (usually per second). Figure 3.3 is a graphical representation of how pipelining can provide a performance improvement. It shows how the first operation enters and exits both

pipelined and non-pipelined versions of the circuit at the same time. However, after the first operation is completed in the non-pipelined version of the circuit, the second operation is still entering into the first stage. In the pipelined version of the circuit, the second, third and fourth operations have entered the pipeline and have passed through some stages of the circuit. Therefore, by the time the non-pipelined circuit has completed the second operation, the pipelined version will have completed up to the fourth operation and will already have the fifth, sixth and seventh operations partially processed.

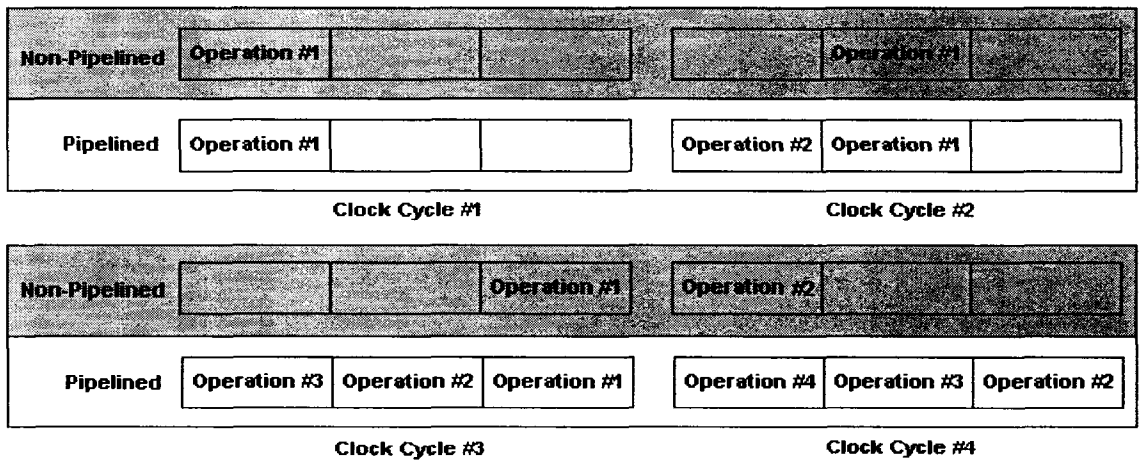

Figure 3.3: Pipelining Example

3.5.1 Asynchronous Micropipelines

The asynchronous micropipeline is based on the C-Element circuit discussed previously. A special kind of flip-flop, an event controlled storage element, was proposed to be used in conjunction with the C-Element to build the micropipeline. However, it was found that it is much easier to implement the micropipeline using a dual-edge triggered flip-flop. So in the research leading to this thesis, all micropipelines were built with dual-edge triggered flip-flops instead.

Asynchronous pipelines using both symmetric and conventional Muller C-Elements were studied in this thesis. Figure 3.4 shows the basic structure of an asynchronous micropipeline as proposed in [30]. This structure was used for the micropipelines studied in this thesis, while using different C-Element implementations and comparing the results.

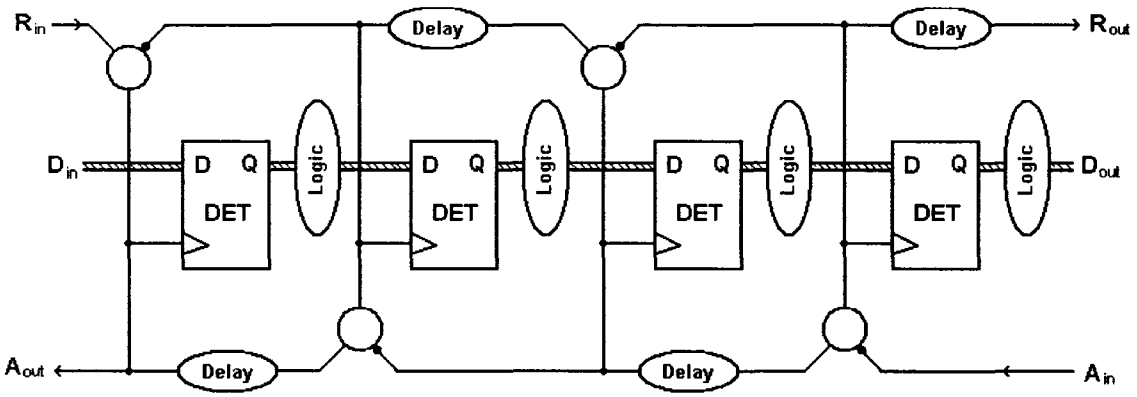

Figure 3.4: Asynchronous Micropipeline Structure

The micropipeline consists of dual edge triggered flip-flops and logic as the data path, and C-Elements and delay elements as the control path. The delay elements are used to match the delay through the control path to the delay through the flip-flop and logic of each stage. The data enters the micropipeline at D_{in}, and exists at D_{out}. The data path can be a bus and can be as wide as necessary. If it is more than one bit wide, then for each bit a flip-flop would be connected in parallel. The micropipeline shifts in the first data item when the request input R_{in} signal transitions. This triggers the first C-Element to transition its output. Since the flip-flops are dual-edge triggered, the flip-flops will shift in the new data-items on both rising and falling edges. At the same time, the R_{in} signal is

looped back out to the data generating circuit as an acknowledge signal, A_{out}. This signal informs the data generating circuit that the current data item has been shifted into the first stage of the micropipeline, and that the next data item can be placed onto the data bus at D_{in}. The output of the first C-Element also causes an event on one of the inputs of the second C-Element, causing it to transition. This shifts the data into the second stage of the micropipeline and triggers the next C-Element. This process continues until the micropipeline is full.

It is important to realize that each C-Element requires two events in order to trigger an event at its output. First, the previous C-Element needs to trigger one of its inputs to signal that a new data item is ready. Secondly, the next C-Element needs to trigger the other input to acknowledge that the next stage has already shifted in the current data item and that it is not needed anymore. At that point, the current data item is overwritten by the new one. At the output of the micropipeline, the same procedure needs to take place for the last stage. The circuit that is taking data from the micropipeline interfaces with the micropipeline control circuitry using two signals, R_{out} and A_{in}. R_{out} signals that a new data item is ready at the output of the micropipeline, and A_{in} signals that the data item at the output of the micropipeline has been successfully used and is no longer needed.

3.5.2 GasP Asynchronous Pipelines

Another type of asynchronous pipeline investigated in this thesis is the GasP pipeline, proposed in [40]. The idea behind GasP is to improve the speed of asynchronous

pipelines by reducing the complexity of the control circuitry. By doing this, the associated overhead in terms of circuit area, power and delay is reduced. In addition, GasP pipelines use simple data latching circuits composed of only three inverters, which are much simpler than the dual edge triggered flip-flops required by micropipelines. This contributes significantly to improving the speed and reducing the power dissipation of GasP pipelines.

However, GasP circuits suffer from dependence on time, which requires very careful and accurate transistor sizing to match delays through various branches in the circuit. This makes GasP pipelines more difficult to design than C-Element based asynchronous micropipelines, and the dependence on time goes against the basic principle of delay insensitivity for asynchronous circuits.

Advocates of GasP pipelines claim that GasP combines the best of synchronous and asynchronous design techniques, and have reported improved results compared to other asynchronous pipelines. The basic structure of a GasP pipeline is shown in Figure 3.5.

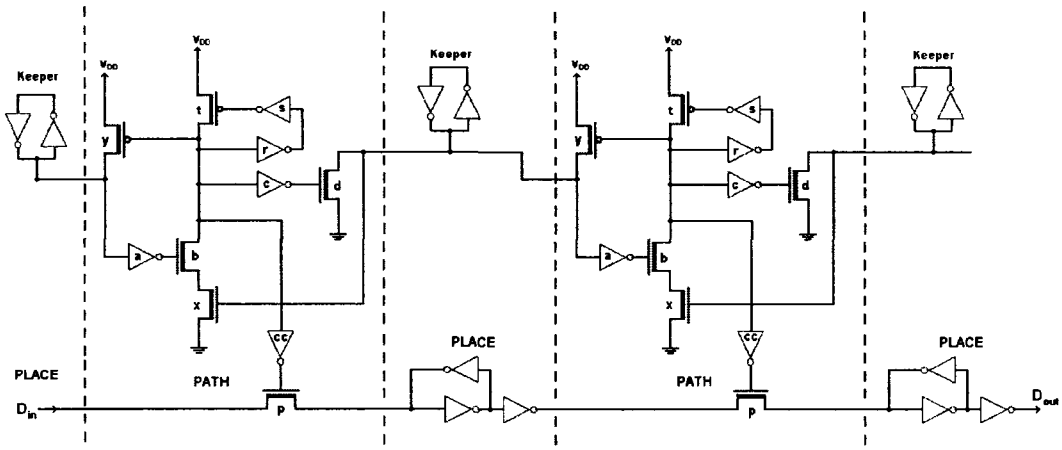

Figure 3.5: GasP Asynchronous Pipeline

GasP circuits rely on a "keeper" circuit to store the state of each stage in the pipeline. The "keeper" consists of two inverters in a loop, and the node where the state is stored is called a "state conductor". There is also a data latch consisting again of three inverters, two in series to maintain polarity and a third that loops around to latch the data. The combination of the "keeper", "state conductor" and the data latch is called a "Place" in GasP terminology. The "Place" is where each stage of the pipeline stores its current state and data value.

The control circuitry of GasP pipelines is referred to as the "Path". This consists of two NMOS transistors in series, labeled b and x in Figure 3.5, that trigger when the previous state is "full" and the next state is "empty". When these conditions are fulfilled, this means that a new data item should be copied to the next stage. This is done through the inverter cc, which turns on transistor p long enough to pass a data item to the next stage of the pipeline. At the same time, the state of these two stages needs to be updated. Inverter c and transistor d pull the state conductor of the next stage low, indicating that it

is now "full". The PMOS transistor y pulls the previous state conductor high, indicating that it is now "empty". Finally, inverters r and s, and transistor t reset the "Path" circuit high to prepare for the next cycle.

In order for all of this to happen, very careful transistor sizing must be performed. For example, the self-reset circuitry must be sized such that it does not reset too quickly, which would interrupt the data and state transfer process. Also, there is a lot of contention in GasP pipelines. The keeper must be overwritten by force, meaning that the keeper inverters need to be sized small to allow the transistors d and y to overwrite them quickly and without excessive current draw. Also, to optimize the speed of the GasP pipeline, inverter a, transistors b and x, inverter cc and transistor p must be appropriately sized as they are what determine the delay of the data and state transfer. Incorrect sizing of these components leads to a non-functioning or poorly performing pipeline. However, since the data path and control circuitry is simpler that that used in asynchronous pipelines, GasP pipelines can achieve greater performance.

CHAPTER 4

Multi-Threshold Asynchronous Pipeline Primitives

4.1 Introduction

This chapter discusses the primitives of asynchronous pipelines in more detail. The asynchronous micropipeline introduced in the previous chapter is composed of three main components. The C-Element is the main control circuit that performs the handshaking between the stages of the pipeline. Event triggered storage elements are needed to capture and hold data as it is transferred from one stage to the next. For the purposes of this thesis, we chose to use dual edge-triggered flip-flops for this purpose. Finally, delay elements may also be required for an asynchronous micropipeline to function properly.

In the case of a simple FIFO, where there is no logic between the storage elements in the pipeline, no delay element is required. This is because the delay of the control signals passing through the C-Element is large enough to ensure that the setup time of the

flip-flop is not violated, meaning that the data is ready at the input of the next stage of the pipeline when the control signal is capture the data arrives. However, when additional logic is inserted between the pipeline stages, such as in the case of a pipelined adder for example, this is unlikely to work. Therefore, delay elements need to be inserted in the control path of the micropipeline in order to match the delay of the control path and the data path.

The GasP pipeline uses a control circuit that is similar in function yet simpler than the C-Element. It also uses a simpler data latch in place of the event triggered storage element. Delay elements may be used or delay can be added to the GasP control circuitry by transistor sizing. The GasP pipeline primitives were intentionally designed to be simpler in order to achieve higher speed than the asynchronous micropipeline.

The multi-threshold low power supply voltage techniques discussed in previous chapters of this thesis were applied to these asynchronous pipeline primitives. Also, improvements to some primitives were proposed. A 0.13 μm CMOS technology with multi-threshold capability was used to run schematic level simulations in order to help us better understand the behavior and performance of the various primitives and techniques. This chapter presents the primitives and the simulation results, and it compares the results to help form a better understanding of which techniques work best.

4.2 The C-Element

The C-Elements studied in this research thesis are the conventional C-Element as described in [30] and the symmetric C-Element as described in [39]. Techniques for

minimizing leakage current and reducing the power supply voltage in order to lower power consumption were applied to these two C-Element implementations. In addition, new C-Element implementations that minimize power dissipation while maintaining high performance were explored. The following sections will describe the circuits that were studied and simulated. The results of the simulations will then be presented and discussed.

4.2.1 Simulation Methodology

The C-Elements discussed in this chapter were all designed for and simulated with a power supply voltage of 0.9 V as opposed to the 1.2 V normally used for 0.13 µm CMOS process technology. This significant reduction in power supply voltage achieves the overall goal of lowering dynamic power dissipation. Multi-Threshold transistor techniques were applied to achieve a balance between performance and static leakage power dissipation.

The test bench circuit used to simulate the C-Elements is shown below in Figure 4.1. As the figure shows, inverters were used to drive the inputs and as a load on the output of the C-Element in order to make the simulation environment realistic. The inverters at the inputs ensure that the input waveforms have realistic rise and fall times, and realistic drive strength. The inverter at the output of the C-Element ensures that the output of the C-Element has a realistic load.

The input waveforms used, shown in Figure 4.2, were designed to test all possible input combinations to the C-Element. This includes the order of change in the inputs, as

this has an effect on the performance of the circuits. Propagation delays from input to output were measured, as well as the energy consumed per cycle of the C-Element. These results were then used to compare the different C-Element implementations.

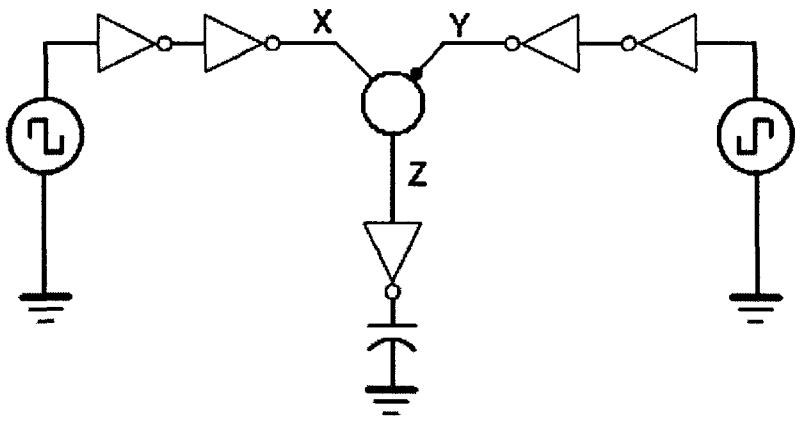

Figure 4.1: C-Element Test Bench Circuit

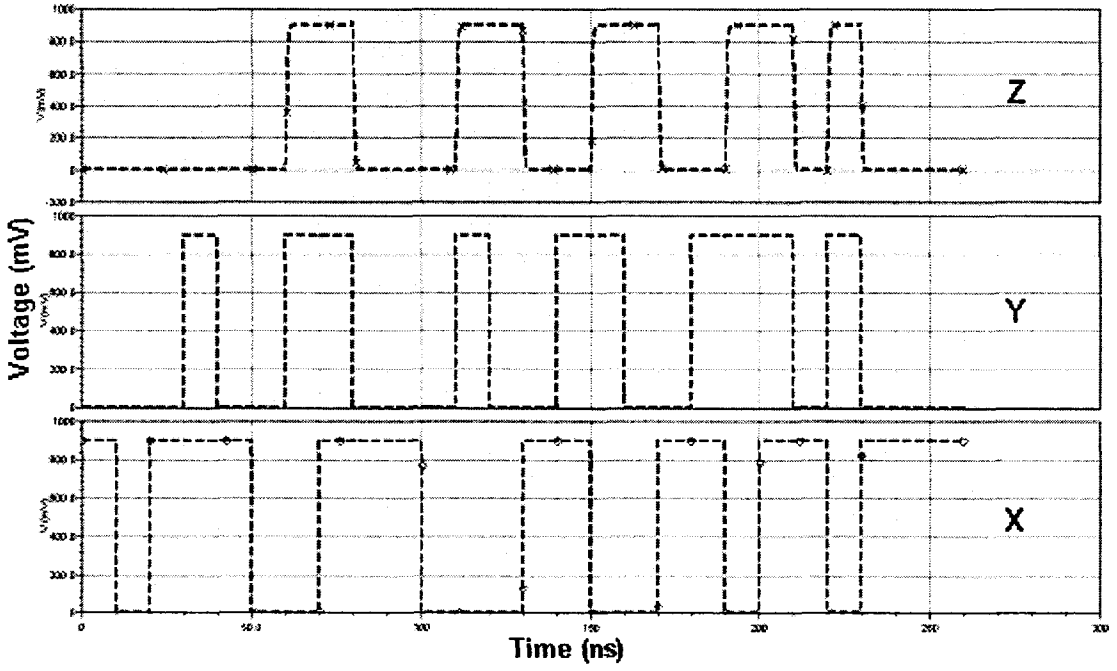

Figure 4.2: Test Waveforms Used to Simulate C-Element Circuits

4.2.2 Conventional C-Element

The circuit diagram of the conventional C-Element is shown in Figure 4.3. Several variations of the conventional C-Element were simulated, and these are described briefly in the coming sections.

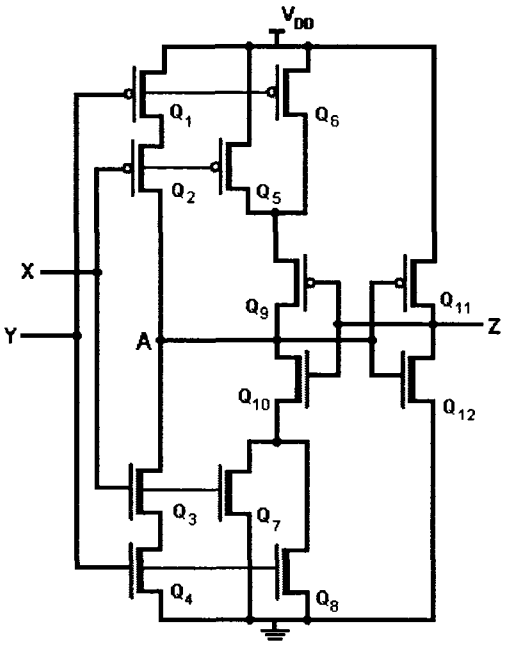

Figure 4.3: Conventional Muller C-Element

4.2.3 Conventional C-Element Transistor Sizing

By studying the conventional C-Element circuit, the relative sizing of the transistors can be determined. Transistors in the critical path need to be sized large enough to allow for sufficient current flow to switch the output quickly. The remaining transistors can be sized small as they are not required to draw much current. Transistors $Q_1 - Q_4$ and $Q_{11} - Q_{12}$ are the critical path of the circuit. Transistors Q_9 and Q_{10} provide an output latch, and $Q_5 - Q_8$ provide them with connections to the power supply

and to ground. Based on this analysis, it is clear that $Q_1 - Q_4$ and $Q_{11} - Q_{12}$ should be sized reasonably, while $Q_5 - Q_{10}$ can have minimum size. It is also required to have roughly equal rise and fall times at the output of the C-Element. This means that PMOS transistors should be sized roughly twice as large as NMOS transistors to account for the hole mobility being lower than electron mobility.

Based on this analysis, the relative sizing of the transistors was done. To optimize the transistor sizing a parametric simulation was performed across a range of reasonable transistor sizes, and the energy-delay product of the C-Element was plotted. This simulation was run on a basic conventional C-Element circuit, and the sizing of all conventional C-Elements was done to minimize the energy-delay product according to the results shown in Figure 4.4.

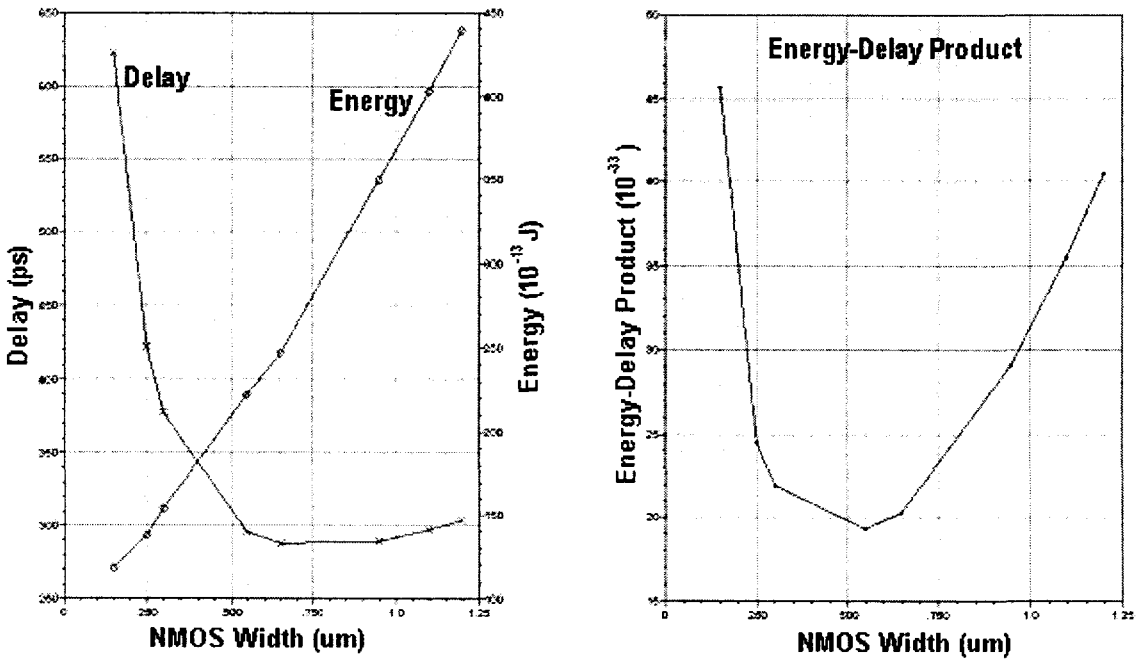

Figure 4.4: Transistor Sizing to Minimize Energy-Delay Product

4.2.4 High Threshold Voltage Transistor Conventional C-Element

This C-Element was built using high threshold transistors, and is meant to be used as a benchmark to compare with the other variations of the conventional C-Element. Since all transistors are high threshold, this implementation is expected to be the slowest but to have the lowest leakage current and hence the lowest power consumption. Collecting simulation results from this circuit and comparing them with the C-Elements where the low power techniques were applied will clearly demonstrate the benefits of the low power techniques.

4.2.5 Low Threshold Voltage Transistor Conventional C-Element

This is the other benchmark circuit needed to evaluate the proposed C-Elements with the low power techniques applied. Since this circuit consists of low threshold transistors, it is expected to show the best performance in terms of speed, and to have the highest static leakage current. This circuit should represent the worst case in terms of static leakage current since it is implemented in low threshold transistors and none of the techniques to counter static leakage current are applied to it. It should also represent the best case in terms of high speed for the same reasons.

4.2.6 Mixed Threshold Voltage Transistor Conventional C-Elements

As discussed earlier, one of the techniques to reduce static leakage current in a circuit while maintaining performance is to intelligently select the threshold voltage of each transistor in the circuit. Those transistors that are responsible for switching the output are the transistors responsible for the delay characteristics of the circuit. Therefore, in order to maintain high performance when using a low power supply voltage, these

transistors need to have a low threshold voltage. Other transistors in the circuit are not in the critical path and do not directly switch the output. They may be used as keeper transistors or to latch the output, but they are not responsible for charging and discharging the output. These transistors have little or no effect on the propagation delay from input to output. Therefore, these transistors can be implemented as high threshold voltage transistors, even when a low power supply voltage is used to lower power consumption. This technique of mixing high and low threshold voltage transistors in one circuit to obtain the best results was applied to the conventional C-Element.

Transistors Q_1 – Q_4 and Q_{11} – Q_{12} were implemented using low threshold voltage transistors, and Q_5 – Q_{10} using high threshold transistors. This lowers the static leakage current flowing in the non-critical path parts of the circuit, while maintaining the performance of the circuit by not affecting the critical path. This C-Element implementation should therefore give a good compromise between high speed and lower power consumption.

4.2.7 MTCMOS Conventional C-Element

The MTCMOS technique, discussed in detail in a previous chapter, was applied to the conventional C-Element circuit. High threshold voltage PMOS and NMOS transistors are added in order to isolate the circuit from the power supply and ground during sleep mode, while the rest of the circuit is implemented in low threshold voltage transistors to maintain high speed. The addition of series transistors will degrade performance if the sleep control transistors are not sized large enough. However, the MTCMOS technique is expected to achieve a sharp reduction in static leakage current.

The sleep control transistors were sized by running a parametric simulation and sweeping the transistor width of the sleep control transistor. At a certain point, a significant degradation in performance occurs, as can be seen from the simulation waveform shown in Figure 4.5. This point is where the transistor starts to become too small to provide adequate current for the circuit. The sleep control transistors were sized larger than this value. Figure 4.6 shows the circuit structure of the MTCMOS conventional C-Element.

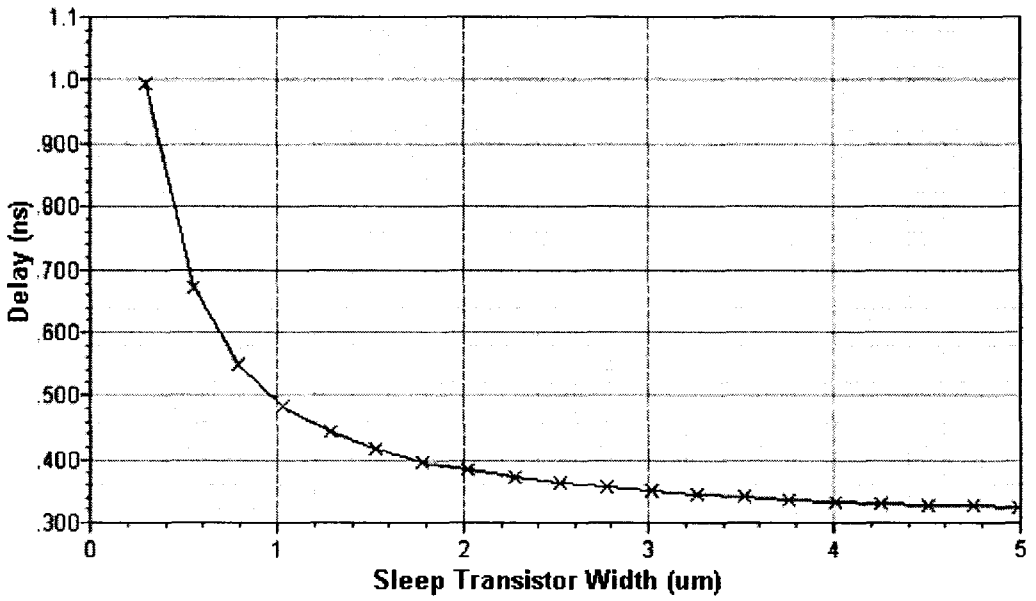

Figure 4.5: MTCMOS Sleep Control Transistor Sizing

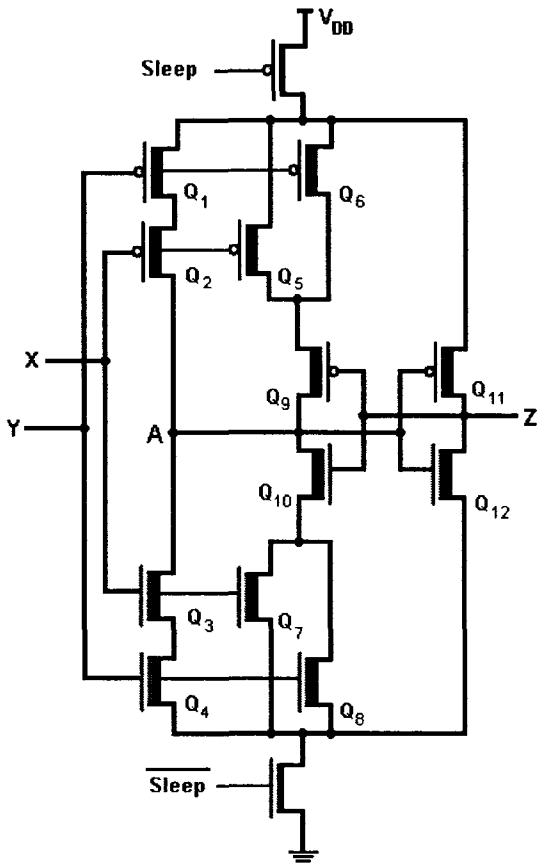

Figure 4.6: MTCMOS Conventional C-Element

4.2.8 SCCMOS Conventional C-Element

The SCCMOS technique is applied to this C-Element, as shown in figure 4.7. All transistors are implemented as low threshold transistors. A sleep control PMOS transistor is inserted between the circuit and the power supply. This PMOS transistor is over-driven during sleep mode in order to minimize leakage. The addition of a series transistor will degrade performance but not as much as the MTCMOS case since it is only one transistor and it is low threshold, therefore it can provide more current. Again, appropriate sizing of the sleep control transistor is essential. The sizing was done using a similar method to that of the MTCMOS C-Element, as shown in figure 4.8. The results expected from the

SCCMOS technique are similar to that of MTCMOS, except that slightly better performance may be achieved.

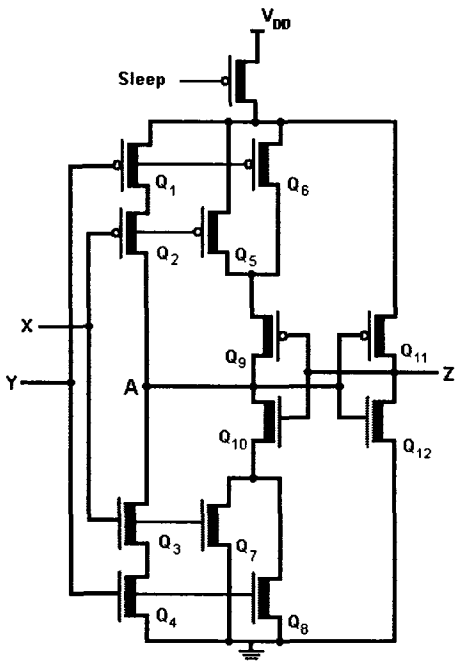

Figure 4.7: SCCMOS Conventional C-Element

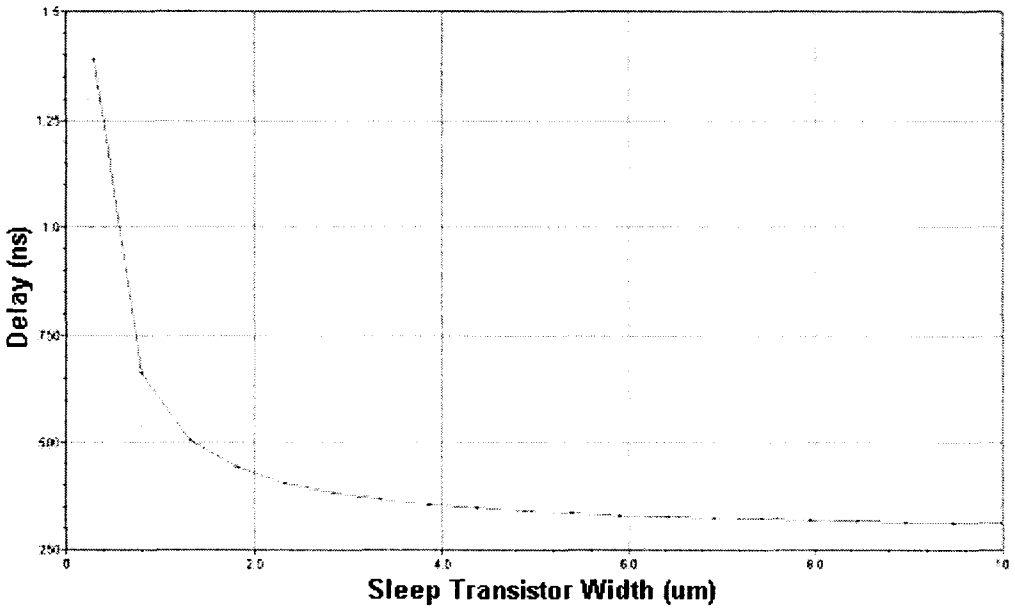

Figure 4.8: SCCMOS Sleep Control Transistor Sizing

4.2.9 VTCMOS Conventional C-Element

The Variable Threshold CMOS (VTCMOS) technique is used in this circuit. The bulk terminal (also known as the body or substrate) of the transistors is switched between two values, one lowers the threshold of all transistors and the other increases the threshold of all transistors. This way the circuit can have two modes, a high speed, high power consumption mode and a lower speed, lower power consumption mode. The VTCMOS technique theoretically should work very well. If it is possible to really control the threshold voltage of the transistors across a wide range of values, the VTCMOS technique should achieve the same high speed as low threshold voltage circuits, and the same low static leakage current as high threshold voltage circuits. When high speed is required, the circuit can be placed into the high speed mode, and when low static leakage current is required, it can be placed in the low leakage state.

VTCMOS is very dependent on the process technology and how well the threshold voltage can be controlled by applying a substrate voltage. In the case of the $0.13\mu m$ process available for this research, it was found that varying the substrate voltage did not provide with a wide enough range of threshold voltage on the transistors.

The VTCMOS Conventional C-Element was implemented using low threshold transistors, as shown in Figure 4.9. Two multiplexers are used to put the circuit into the two different modes of operation. The output of one of these multiplexers is connected to the substrate terminal of each NMOS transistor, while the other one connects to the PMOS transistors. The PMOS transistors substrate voltage, V_{BBp}, is switched by the multiplexer either to V_{DD} or a lower value $V_{DD} - x$. When the substrate is connected to

V_{DD}, the PMOS transistors work as a conventional C-Element circuit implemented using low threshold transistors. But when it is connected to $V_{DD} - x$, the threshold voltage of the transistors is increased. This reduces the static leakage current flowing through the transistors, but also affects the performance of the circuit.

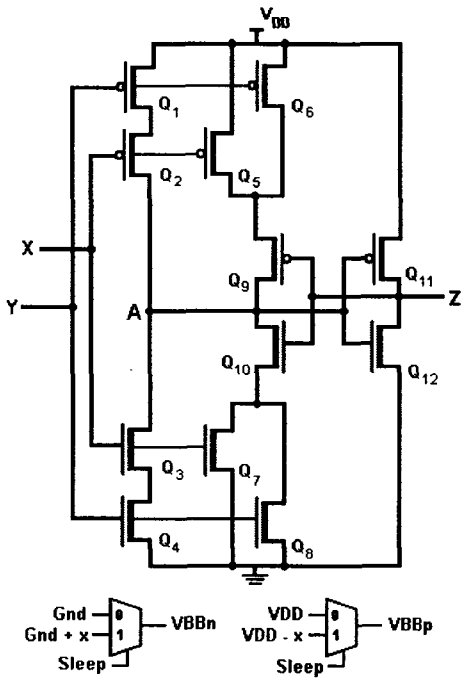

Figure 4.9: VTCMOS Conventional C-Element

The VTCMOS technique requires an additional overhead of two multiplexers, but does not require any sleep control transistors. It also requires a sleep control signal that must be somehow controlled at the system level to place the circuit into the sleep or active modes. Finally, the extra voltage levels of $V_{DD} - x$ and $V_{SS} + x$ must be somehow provided. These voltages are within the available range of voltages in the circuit, so no special charge pump circuit is required as in SCCMOS. These voltages can be generated

by using transistors in active mode and using the voltage drop across them to generate these voltage values. This may increase the static current drawn by the circuit. For the purposes of this thesis however, it was assumed that these voltage levels are provided at the system level. Hence, they were treated as inputs to the VTCMOS conventional C-Element, and were not generated within the circuit.

4.3 Symmetric C-Element

The circuit diagram of the symmetric C-Element is shown in Figure 4.10. Several variations of the symmetric C-Element were simulated, and these are described briefly in the coming sections. The same low power techniques were used as with the conventional C-Element discussed earlier. The techniques were applied in the same way as they were for the conventional C-Element. Additional discussion is provided whenever the application of the technique differed from the conventional C-Element case.

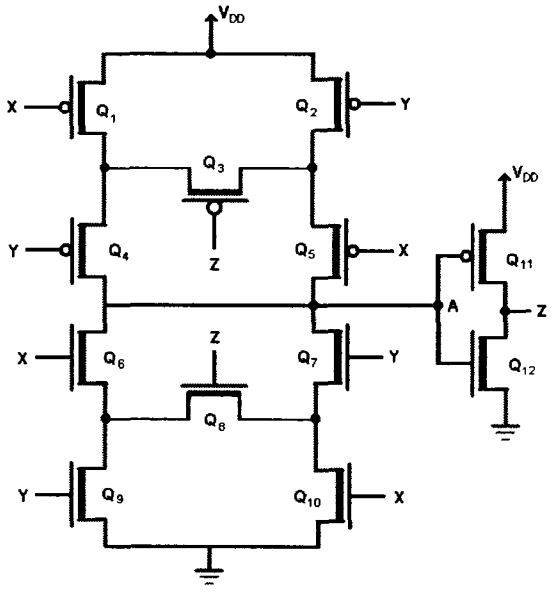

Figure 4.10: Symmetric C-Element Implementation

4.3.1 Symmetric C-Element Transistor Sizing

The same method was used to optimize the transistor sizing of the symmetric C-Element as was used for the conventional C-Element. The operation of the symmetric C-Element was explained in an earlier chapter. Transistors Q_3 and Q_8 are used as latching transistors, and can therefore be minimally sized. The remaining transistors are all part of the critical path of the circuit and must be sized appropriately. In order to maintain equal rise and fall times PMOS transistors were sized roughly twice as large as NMOS transistors to account for the hole mobility being lower than electron mobility.

This analysis gives a basis for relative sizing of the transistors in the symmetric C-Element. In order to optimize the sizing, a parametric simulation was run that swept a range of values while satisfying the relative sizing. The energy-delay product of the C-Element was plotted, as shown in Figure 4.11, and the minimum point of that plot was used to achieve an optimum transistor sizing.

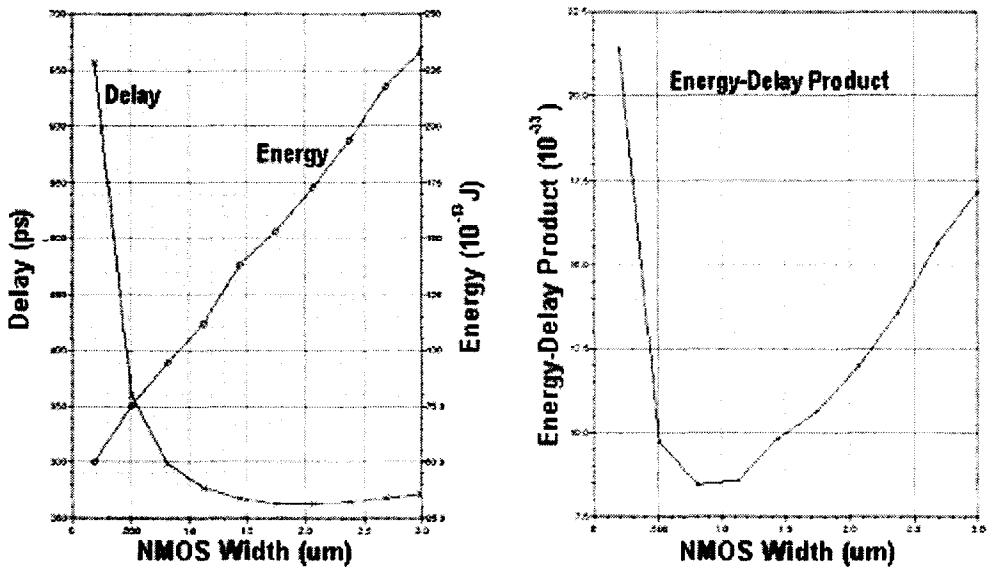

Figure 4.11: Transistor Sizing to Minimize Energy-Delay Product

4.3.2 High Threshold Voltage Symmetric C-Element

This circuit was built using high threshold voltage transistors, and will be used to provide the best case scenario for static leakage current, in a similar manner to that discussed for the High Threshold Voltage Conventional C-Element.

4.3.3 Low Threshold Voltage Symmetric C-Element

Similarly, this circuit provides the best case in terms of propagation delay through the circuit since it was implemented using low threshold voltage transistors. It also provides the worst case scenario in terms of static leakage current since all transistors are low threshold voltage and there are no precautions taken to limit static leakage current.

4.3.4 Mixed Threshold Voltage Symmetric C-Element

As was done for the Mixed Threshold Voltage Conventional C-Element, the first step in implementing this circuit was to analyze the role of each transistor in the

symmetric C-Element. Then it can be determined whether to implement each transistor as a low or high threshold voltage transistor. The same analysis also helps in transistor sizing as discussed earlier.

In the case of the symmetric C-Element it was determined that transistors Q_1, Q_2, Q_4, Q_5, Q_6, Q_7, Q_9, Q_{10}, Q_{11} and Q_{12} have a significant impact on the propagation delay of the circuit. Therefore, these transistors are implemented using low threshold voltage transistors, while transistors Q_3 and Q_8 are implemented as high threshold voltage transistors. Therefore, the output inverter will always have static leakage current flowing through it. Also, the two main branches of the circuit will have static leakage current.

The only leakage path that was disrupted is the one through the latching transistors Q_3 and Q_8. When one input is low and the other is high, either Q_3 or Q_8 is on depending on the level of the output, which creates a path through one of these two transistors. Since they are implemented as high threshold voltage transistors, this should help to minimize the static leakage currents through them. If minimizing static leakage currents just through these two paths is not found to be sufficient, then other techniques need to be implemented.

4.3.5 MTCMOS Symmetric C-Element

The MTCMOS technique was applied to the Symmetric C-Element circuit, and the same methods were used for transistor sizing as was done for the conventional C-Element. Figure 4.12 shows the circuit schematic of the MTCMOS symmetric C-

Element. In contrast to the mixed threshold voltage symmetric C-Element, the MTCMOS technique cuts all static leakage current paths in the circuit when it is placed in sleep mode. However, this comes at the expense of adding two series sleep transistors.

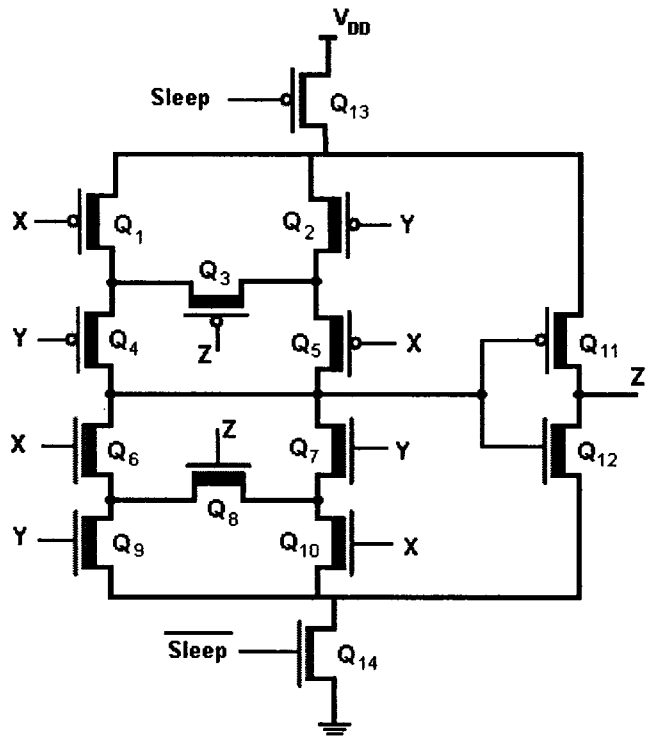

Figure 4.12: MTCMOS Symmetric C-Element

4.3.6 SCCMOS Symmetric C-Element

The SCCMOS technique was applied to the Symmetric C-Element circuit, as shown in Figure 4.13. All transistors were implemented as low threshold transistors. Like the MTCMOS technique, this circuit minimizes static leakage current through all branches when it is in sleep mode. It only requires one sleep control series transistor, and all transistors can have the same threshold voltage. However, an extra charge pump circuit needs to be available on the chip to provide the overdrive voltage.

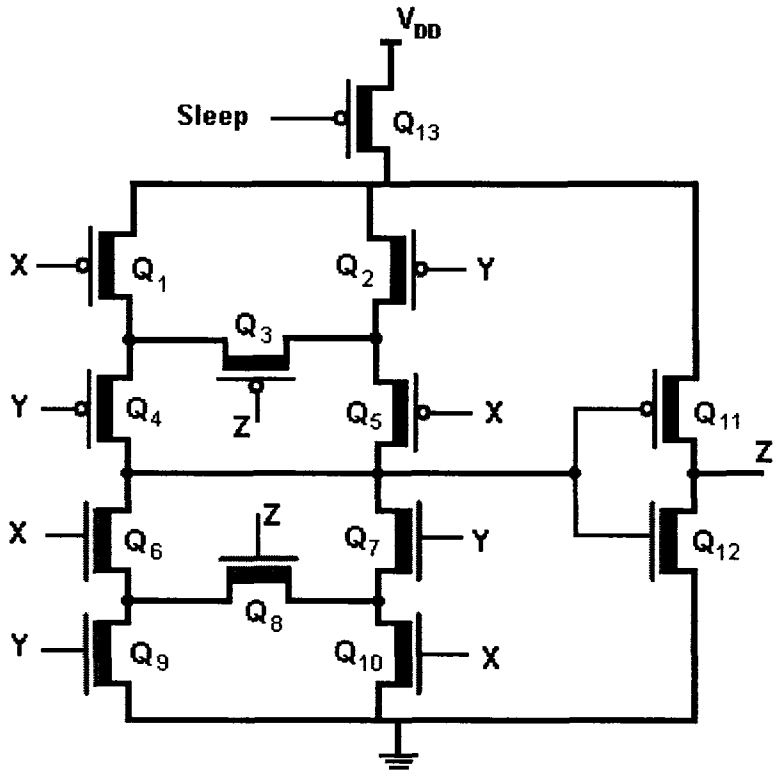

Figure 4.13: SCCMOS Symmetric C-Element

4.3.7 VTCMOS Symmetric C-Element

The VTCMOS technique was applied to the symmetric C-Element in the same way it was applied to the conventional one. Two multiplexers are added that connect two different bulk voltage values to the bulk terminals of the NMOS and PMOS transistors. By switching between these two values, the transistor threshold voltages can be adjusted to either achieve high speed at the expense of high power dissipation, or low power dissipation at the expense of degraded performance. The circuit diagram is shown in Figure 4.14.

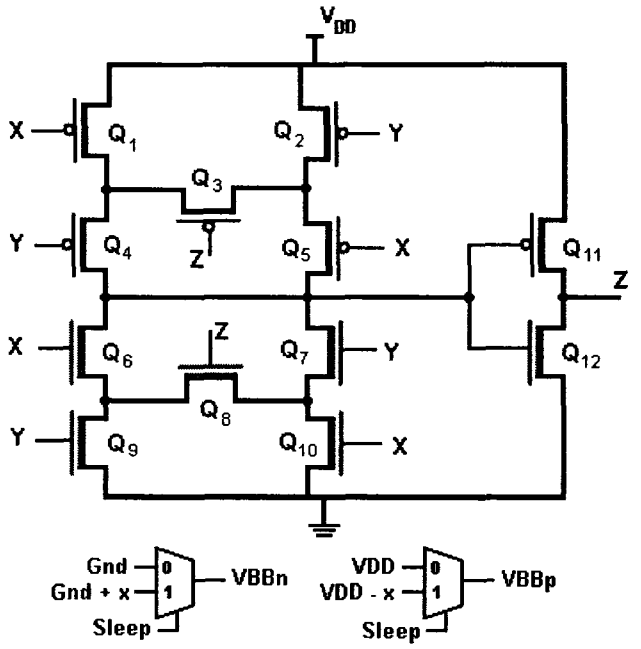

Figure 4.14: VTCMOS Symmetric C-Element

4.4 Proposed C-Element Circuits

This section discusses some improved and newly proposed C-Element circuits. Several ideas were pursued, with varying degrees of success. The circuits are introduced in this section, and the results will be presented in a later section in comparison to all the other C-Element variations attempted.

4.4.1 MISCCMOS Conventional C-Element

The MISCCMOS technique was applied to this C-Element. It is based on the SCCMOS technique. However, instead of adding a series sleep transistor that degrades performance during normal mode, the over-drive voltage is applied to existing PMOS transistors in the C-Element circuit. A multiplexer is used to select between the input and

the over-drive voltage to be applied to the PMOS transistors, and the control voltage of the multiplexer is simply the sleep control signal.

Figure 4.15 shows the circuit structure of the MISCCMOS Conventional C-Element. This C-Element is expected to produce the same static leakage current minimization as SCCMOS in sleep mode, while giving performance equal to that of the low threshold voltage transistor implementation.

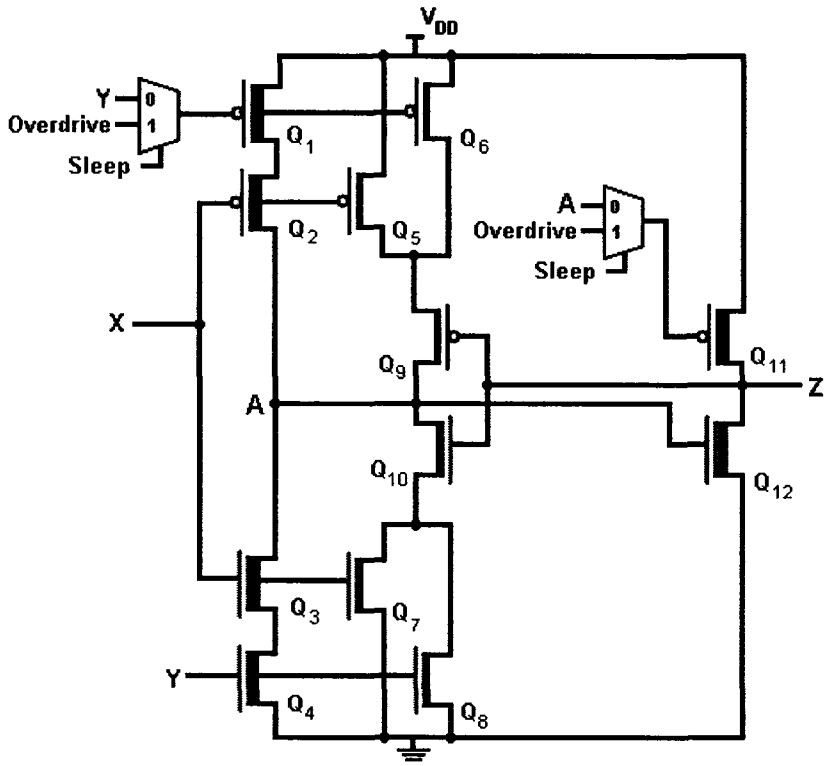

Figure 4.15: MISCCMOS Conventional C-Element

4.4.2 MISCCMOS Symmetric C-Element

The MISCCMOS technique was also applied to the symmetric C-Element, as show in Figure 4.16.

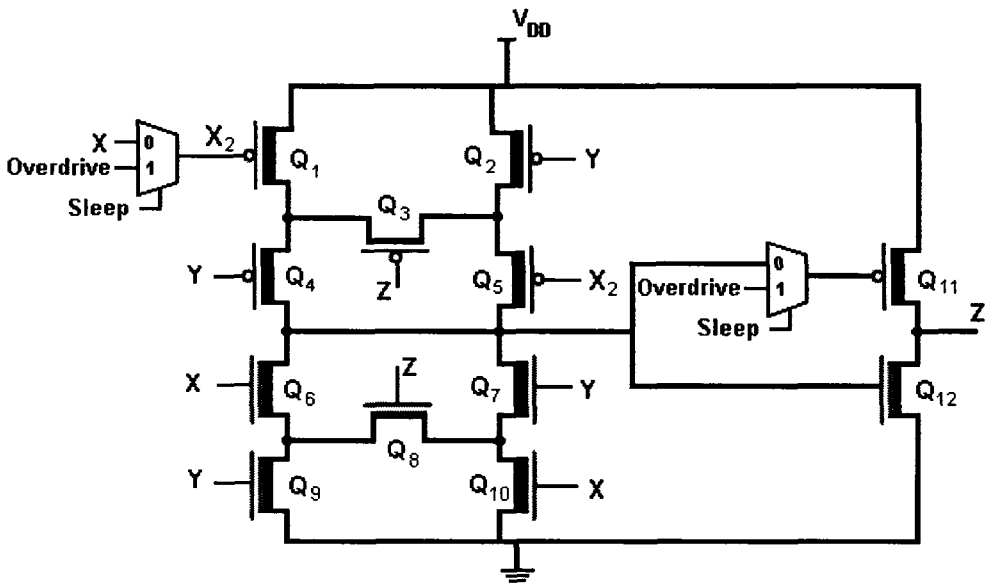

Figure 4.16: MISCCMOS Symmetric C-Element

4.4.3 Latched SCCMOS

One of the problems with MTCMOS and SCCMOS which was discussed earlier was their inability to hold their state when they are placed in sleep mode. This is because the sleep control transistor(s) cut the circuit off from the power supply, meaning that leakage currents will discharge the output capacitance if it was charged and the output state is lost. The latched SCCMOS C-Element was proposed as a way to counter this effect. It is an SCCMOS conventional C-Element to which two inverters are added at the output. The two inverters are connected such that they form a loop, creating a latch at the output of the C-Element, as shown in Figure 4.17.

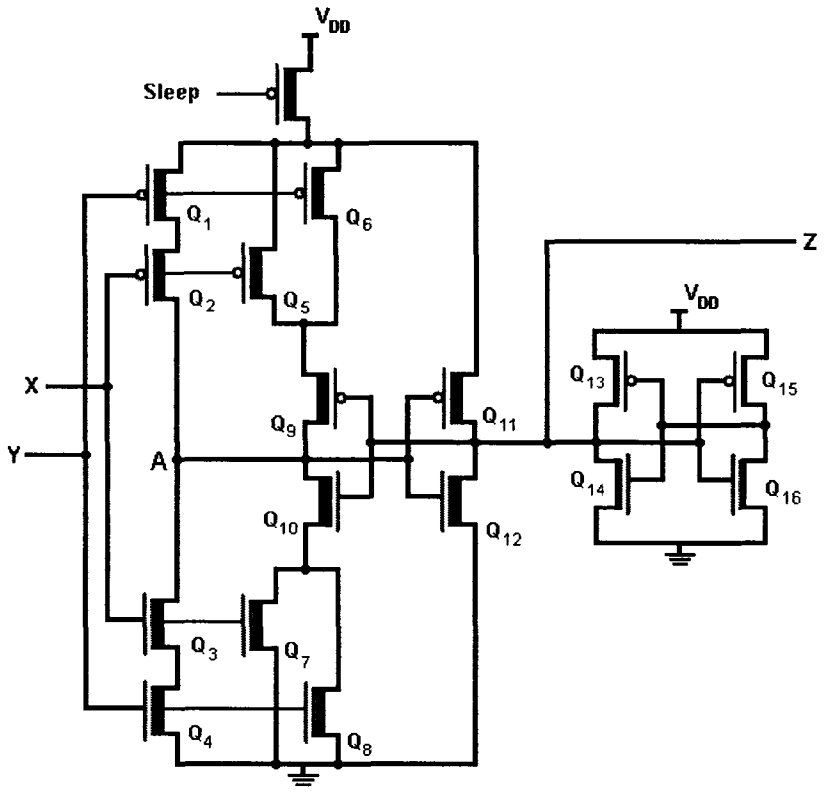

Figure 4.17: Latched SCCMOS Circuit Diagram

The output inverters are connected directly to the power supply voltage and to ground, meaning that they continue to operate even in sleep mode. However, in order to keep their static leakage current draw minimal, they are implemented as high threshold transistors and are sized minimally. This makes it easier for the C-Element to override the latched value, and minimizes resistance which leads to increased switching current. The extra switching current needed to override the output latch means that the latched SCCMOS will draw more current in active mode than the normal SCCMOS C-Element. However, in sleep mode, latched SCCMOS should provide the same reduction in static leakage current as SCCMOS, with the added benefit of not losing its state during sleep mode.

4.4.4 Dual Sleep SCCMOS

Another idea based on SCCMOS that was explored was Dual Sleep SCCMOS (DSSCCMOS). This is another attempt at holding the state of the C-Element when the circuit is placed in sleep mode. The idea was to place a high threshold minimum sized transistor in parallel with the sleep control transistor. This parallel sleep transistor would be always turned on by connecting its gate terminal directly to ground.

The idea is that in active mode, the sleep control transistor, which is low threshold and sized large enough for the C-Element, provides the current required for high speed switching of the output. This ensures that the circuit has the high performance required. On the other hand, in sleep mode, the sleep control transistor is overdriven to strongly turn it off. This limits the static leakage current through the low threshold transistors that make up the C-Element. The parallel high threshold voltage sleep transistor, however, remains turned on. Since it is a high threshold transistor and it is of minimum width, it provides just enough current to maintain the state of the circuit, without wasting power on a high static leakage current. The circuit diagram is shown in Figure 4.18.

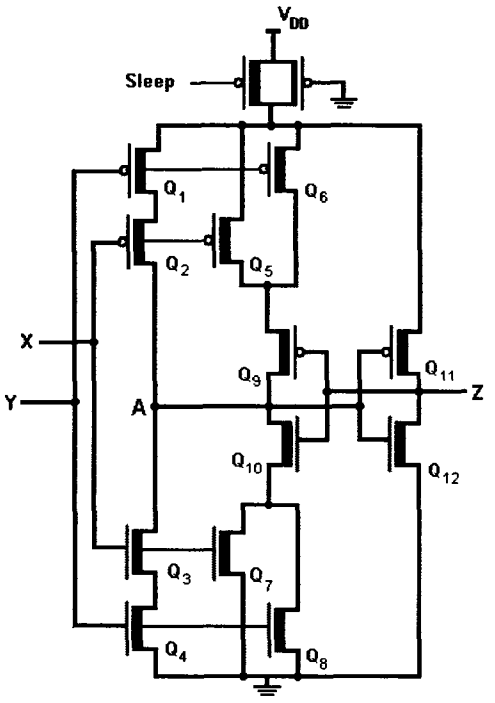

Figure 4.18: Dual Sleep SCCMOS C-Element

4.4.5 NOR Generated Sleep Control Signal

One interesting area of research involves the study of when to place C-Elements into sleep mode. In the MTCMOS and SCCMOS techniques and their derivatives, a sleep transistor is provided that allows the circuit to be placed into a sleep mode. In sleep mode, static leakage current draw, hence static power dissipation, is greatly reduced. At the same time, when not in sleep mode, the circuits have high speed performance because of the use of low threshold transistors in the circuit implementation.

An important question to consider is how to determine when it is appropriate to place the C-Element in sleep mode, and when it is not. This decision can be made at the system level, which requires an algorithm or a methodology to determine when the system will go into sleep mode, and when it will remain in active mode. This adds

complexity at the system level, and so it would be very beneficial to be able to generate a sleep signal at the circuit level. This would make the circuit have the high performance required, while consuming lower power dissipation, and without any additional interference from the system level.

During switching, the circuit needs to be in active mode in order for the switching to be fast. Once switching is complete and the output has reached its final state, the circuit can be placed into sleep mode to reduce static current draw and static power dissipation. It would also be very beneficial if the circuit can also maintain the current state during sleep mode so that no special action needs to be taken when bringing the circuit out of sleep mode.

The SCCMOS sleep transistor needs to be turned on only when the output is being switched from a low to a high state. When the circuit is static or when the output is being switched from high to low, the sleep transistor can be turned off. Therefore, the sleep signal supplied to the gate of the SCCMOS sleep transistor can be derived by applying the NOR function to the inputs A, B and the output C. Only when the output C is low and both inputs A and B are low do we need to charge the output node high. Figure 4.19 shows the circuit diagram of this circuit.

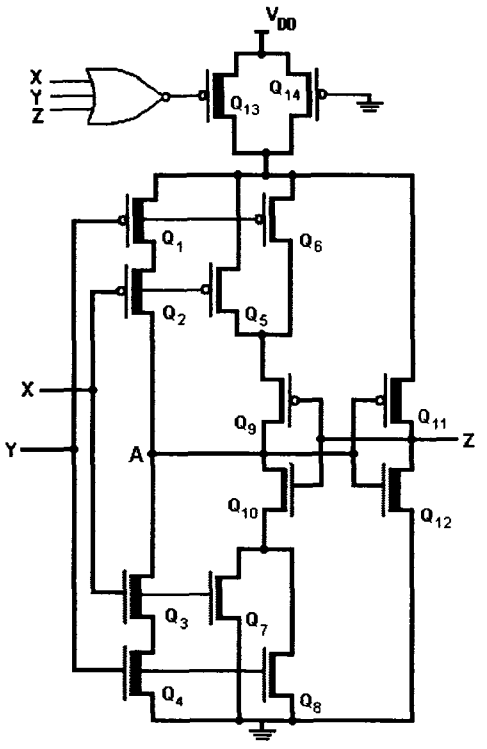

Figure 4.19: SCCMOS C-Element with NOR Generated Sleep Signal

Transistor Q_{13} is the SCCMOS sleep control transistor, and transistor Q_{14} is the parallel high threshold voltage minimum sized transistor that enables the C-Element to hold its state when it is in sleep mode. The input of the sleep control transistor Q_{13} comes from a NOR gate whose inputs are the two inputs of the C-Element and the output of the C-Element. This NOR gate is powered with the overdrive voltage, which means that its output will go up to the overdrive voltage when it is high, strongly cutting-off the sleep control transistor. The NOR gate used provides the sleep control transistor and also replaces the multiplexer that would have been required to switch between ground and the overdrive voltage. Therefore, although the conventional NOR gate consists of six transistors, the four transistors of the multiplexer are no longer needed.

Since the sleep control transistor in SCCMOS can usually be shared among several circuits, and only one multiplexer is needed per sleep control transistor, this technique still has the potential of adding a significant number of transistors over normal SCCMOS. This technique is quite elegant as the NOR gate is powered from the higher over-drive voltage, and therefore its output will swing from ground to the over-drive voltage, providing the proper input to the gate of the sleep transistor without the use of multiplexers as is the case with the conventional SCCMOS technique. The use of multiplexers adds significant leakage currents because of the path between the inputs of the multiplexer, one of which is the over-drive voltage and the other is ground.

This technique was attempted using both a conventional CMOS NOR gate and a pass transistor (PT) design, which was implemented as a PT OR gate, and an inverter powered by the over-drive voltage to produce the final sleep signal. The PT implementation was intended to be faster, but introduced a new problem. The V_{th} voltage drop across the transistors in pass transistor logic meant that the input of the inverter did not have a full swing signal at its input. Since NMOS transistors were used, the voltage at the input of this inverter ranged from 0 to $(V_{DD} - V_{th})$. The output of the inverter was able to switch high very fast. However, when the input of the inverter was $(V_{DD} - V_{th})$, the NMOS transistor of the inverter was not turned on fully, so it was slow to pull the sleep signal low. In addition, the PMOS transistor in the inverter was also not fully turned off, which meant that it was passing some current. This was causing resistance to the NMOS in pulling the output low. A large amount of current was drawn from the over-drive power supply as the inverter was pulling the sleep signal low. This also meant that

the low V_{th} sleep transistor meant to help provide current for switching of the C-Element got turned on too late, which meant that the C-Element was relying on the high V_{th} transistor, resulting in very poor performance.

An attempt to fix this was made by using complimentary pass-transistor logic instead of conventional pass-transistor logic, as it has full swing at its output. In this version, the NOR gate is implemented as a CPL OR gate followed by an inverter.

4.4.6 Embedded Sleep Control C-Element

This newly proposed C-Element circuit tries to accomplish the same goal as the one presented in the previous section. It generates the sleep control signal from within the circuit. It also minimizes the number of transistors in the circuit, providing the same solution but with fewer transistors. This simplicity is reflected in the simulation results, which show that this proposed circuit greatly outperforms the one proposed in the previous section.

The circuit is shown in Figure 4.20, and is based on the conventional C-Element circuit. However, the latching inverter and the transistors that connect it to the power supply and to ground have been moved. By moving the PMOS and NMOS transistors of this inverter to be adjacent to the power supply and ground, these transistors can be used as sleep control transistors. The latching function is still the same as that of a conventional C-Element. However, now only these two transistors need to be high threshold, and they will efficiently cut-off any static leakage currents in the circuit. Therefore, they act as sleep control transistors for their branch of the circuit.

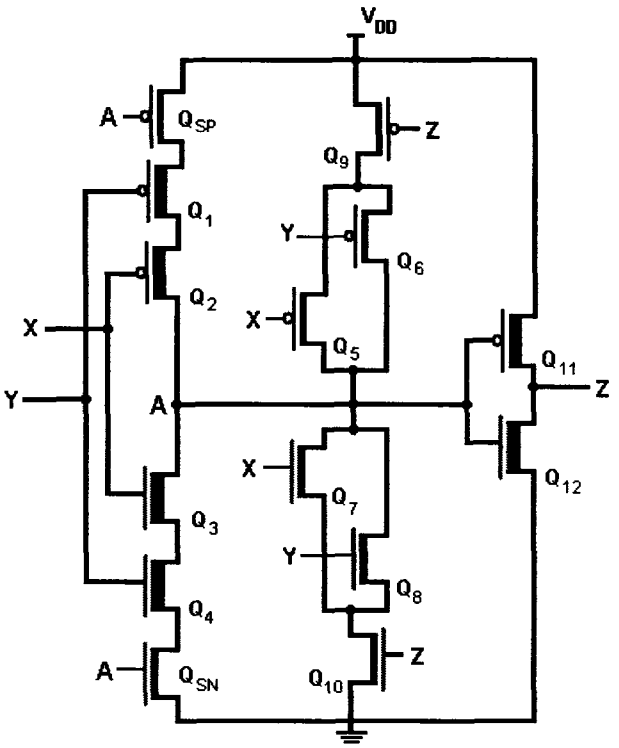

Figure 4.20: Embedded Sleep Control C-Element

The main innovation of this C-Element, however, is the addition of a third transistor, Q_{SP}, in series with the two transistors of the pull-up network in the first branch of the circuit. A similar NMOS transistor, Q_{SN}, is added in the pull-down network, and both these transistors are implemented as high threshold transistors. These transistors have their gates connected to node A, as labeled in Figure 4.20. The effect of this is as follows. Assume that node A starts low, with both inputs high turning on both NMOS transistors of the pull-down network. At this point, transistor Q_{SP} will be turned on, while transistors Q_1 and Q_2 are off. Transistor Q_{SN} will eventually turn itself off once A reaches a low enough value, cutting off any static leakage current. When the inputs switch to both low, transistors Q_1 and Q_2 will turn on, while Q_3 and Q_4 will turn off.

Since transistor Q_{SP} is on, it will pull up node A, along with transistors Q_1 and Q_2. As the node A is charged, Q_{SP} will slowly turn off, eventually cutting off any further static leakage current. Node A maintains its value due to the latch within the conventional C-Element, which holds the state. Since no sleep control transistors are used, there is no sleep mode and the state is not lost.

This method cuts static leakage current in the main branch of the conventional C-Element, while maintaining high performance. The performance is not degraded too much by the addition of the high threshold series transistors because they are already on before the two other transistors in series with them turn on. For example, when A is low, Q_{SP} is already on and it provides a virtual power supply voltage to transistors Q_1 and Q_2, which are off at this point. Therefore the speed of the high threshold transistors is not an issue during switching, as they take their time turning on and off when they are not being used.

4.4.7 Simplified Conventional C-Element

While analyzing the conventional C-Element circuit, it was noted that transistors Q_5, Q_6, Q_7 and Q_8 were not essential to the operation of the C-Element. Their main purpose is that they disconnect the latching inverter from the power supply and ground when the C-Element output is switching. This is done in order to minimize resistance to the switching of the output, which would increase the propagation delay and the switching current drawn. Therefore, removing these four transistors will result in a working but inefficient C-Element circuit.

When using multi-threshold transistor techniques, the ability to select which transistors in a circuit will be implemented as low threshold voltage and which ones will be implemented as high threshold voltage transistors gives the designer extra flexibility. The proposed simplified conventional C-Element takes advantage of this flexibility, as shown in Figure 4.21. As discussed in the mixed threshold voltage implementation of the C-Element, the latching circuitry should be implemented as high threshold transistors in order to minimize static leakage current. This does not affect performance because the latching circuitry is not in the critical path and has little effect on the overall propagation delay.

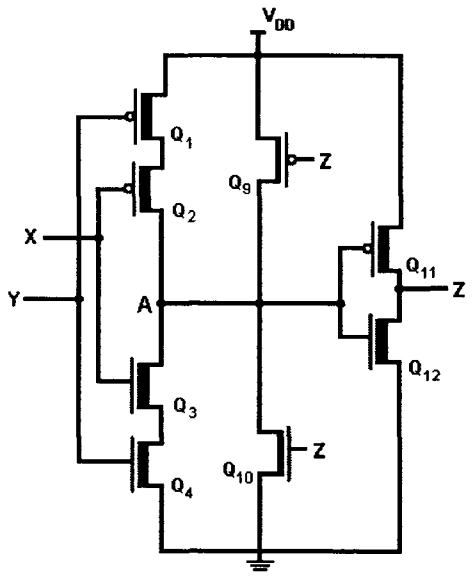

Figure 4.21: Simplified Conventional C-Element

By implementing transistors Q_9 and Q_{10} as high threshold transistors while the remaining transistors are low threshold transistors, the latching inverter presents very little resistance to switching the output. When combined with intelligent transistor sizing,

the resistance can be made small enough to allow the removal of those extra transistors with much less degradation in performance and power dissipation. The larger width low threshold transistors provide enough current to easily override the latched output value. This allows the removal of four transistors from the C-Element, simplifying the circuit and decreasing area requirements.

4.4.8 Super Cutoff C-Element

Another innovative contribution of this thesis is the Super Cutoff C-Element presented in this section. This C-Element has a similar structure to the simplified C-Element circuit presented earlier, with one important distinction, as shown in Figure 4.23. The output inverter of the C-Element is powered by the overdrive voltage instead of using the normal power supply. The effect of this is that the output of the C-Element will be overdriven when it is high. The inputs of each C-Element will in general be the outputs of the previous and the next C-Elements in the pipeline. Therefore, if a pipeline is implemented using Super Cutoff C-Elements, the inputs of each C-Element will be overdriven when they are high, which will result in a drastic reduction of the static leakage current. The transistors in the Super Cutoff C-Element are implemented as low threshold voltage transistors, which ensure the best possible performance is maintained even when the power supply voltage is reduced.

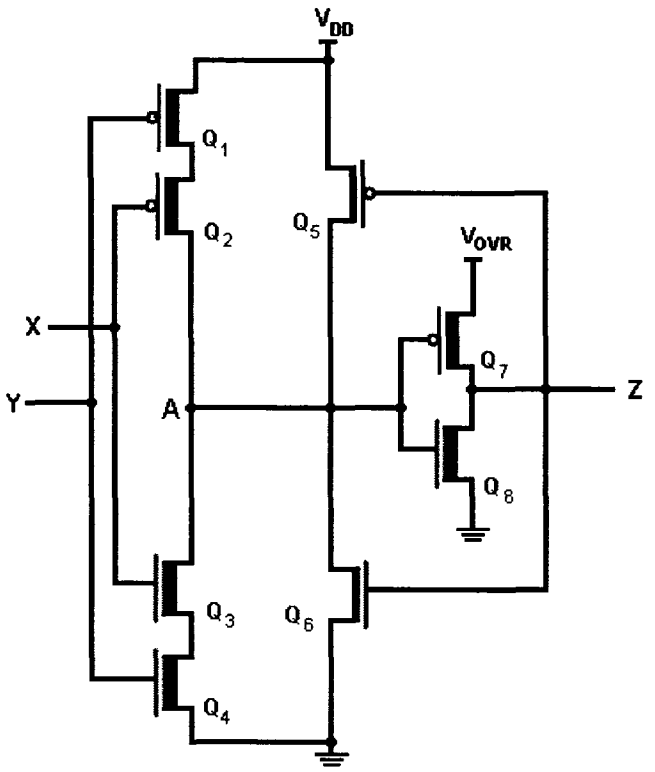

Figure 4.23: Super Cutoff C-Element

The only branch in the Super Cutoff C-Element that is not overdriven is the output inverter itself. This has to be implemented as low threshold transistors in order to maintain high performance. However, since the supply voltage of the inverter is slightly higher, this helps to reduce the static leakage current through the inverter since the difference between the threshold voltage and power supply voltage is larger.

Another limitation of this proposed new C-Element is that there is one situation where the C-Element will still have static leakage current. This occurs when both inputs of a C-Element are low, which means that neither one of the PMOS transistors is overdriven. However, for all other possible input combinations, the static leakage current will be minimized.

4.5 C-Element Simulation Results

The following sections present the simulation results obtained for all of the C-Element circuits discussed in this chapter.

4.5.1 Conventional C-Elements

This section presents the simulation results for the C-Element circuits based on the conventional C-Element. Table 4.1 below shows the results obtained for each circuit attempted. The circuit name is shown in the first column. This is the name that will be used to refer to each circuit in this thesis. The second column shows the average dynamic current drawn when the circuit is in active mode. The third column shows the static leakage current when the inputs are stable. This was averaged for all possible input combinations, and measured with the circuit at rest. The last column shows the 50% input to output propagation delay.

Table 4.1: Conventional C-Element Simulation Results

Name	Average Dynamic Current	Static Leakage Current	Input to Output Propagation Delay
High V_{TH}	3.59 μA	506 pA	499.98 ps
Low V_{TH}	1.28 μA	895 nA	313.67 ps
Mixed V_{TH}	3.25 μA	829 nA	278.59 ps
Mixed V_{TH} 2	0.67 μA	918 nA	282.75 ps
MTCMOS	2.55 μA	45.3 pA	383.37 ps
SCCMOS	2.90 μA	784 pA	228.60 ps
VTCMOS	7.55 μA	829 nA	279.59 ps

The simulations results are easier to interpret when presented as charts. Chart 4.1 shows the average current drawn in normal mode for all circuits, presented as a bar chart. Chart 4.2 shows the static leakage current of each circuit, while Chart 4.3 shows a close-up comparison of the circuits with the lowest static leakage currents. Chart 4.4 shows the propagation delay of each circuit. Finally, Chart 4.5 shows the energy-delay product of all circuits.

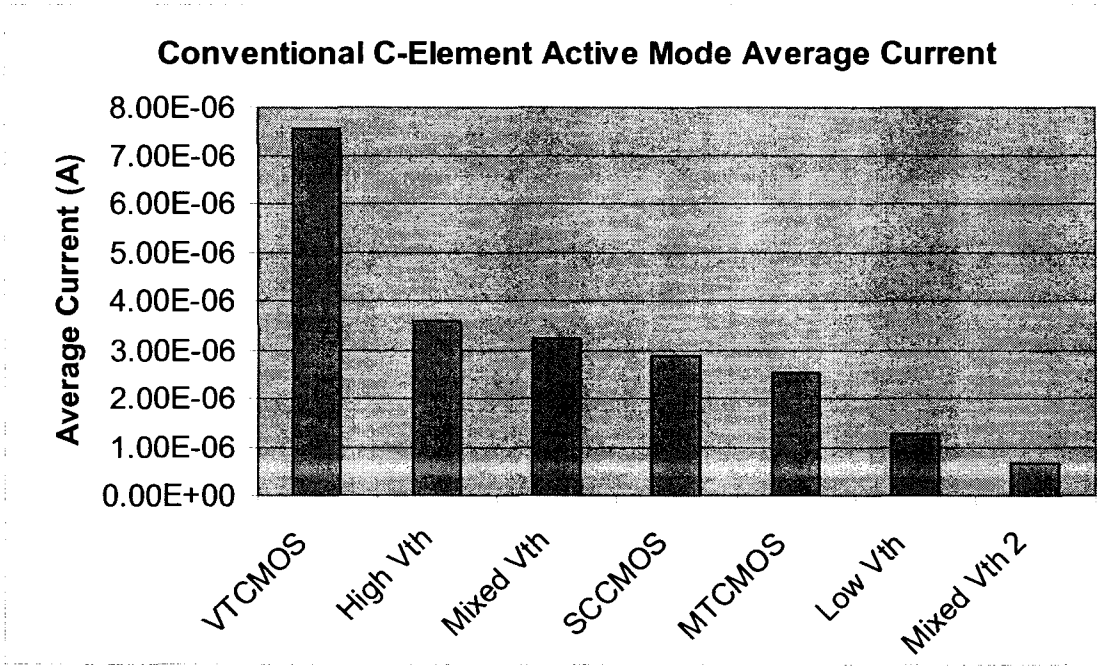

Chart 4.1: Conventional C-Element Average Current

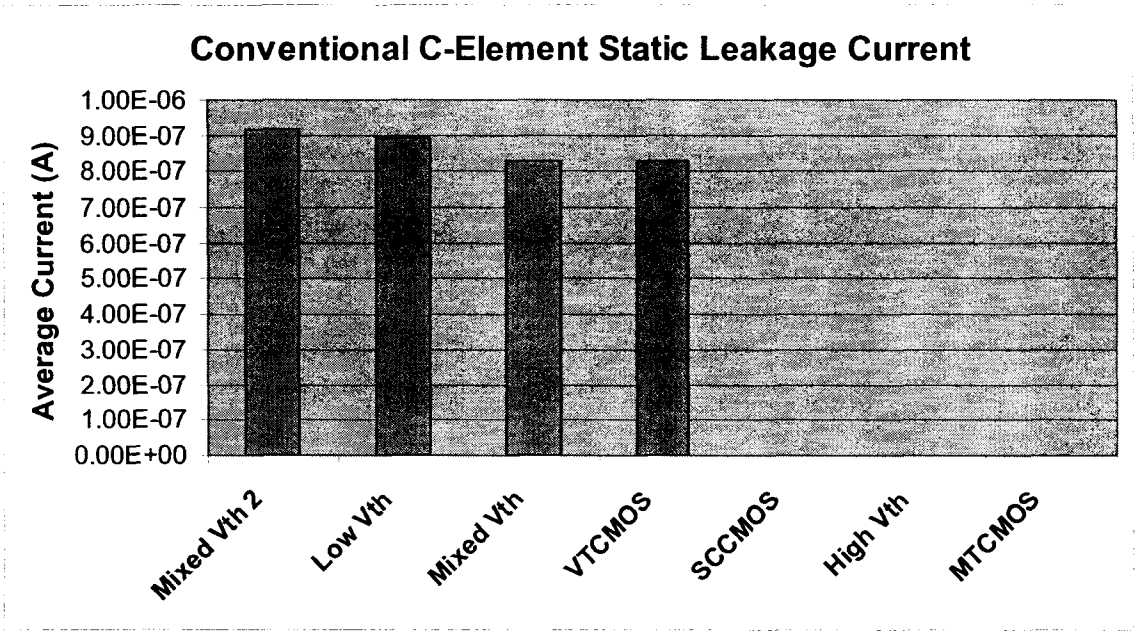

Chart 4.2: Conventional C-Element Static Leakage Current

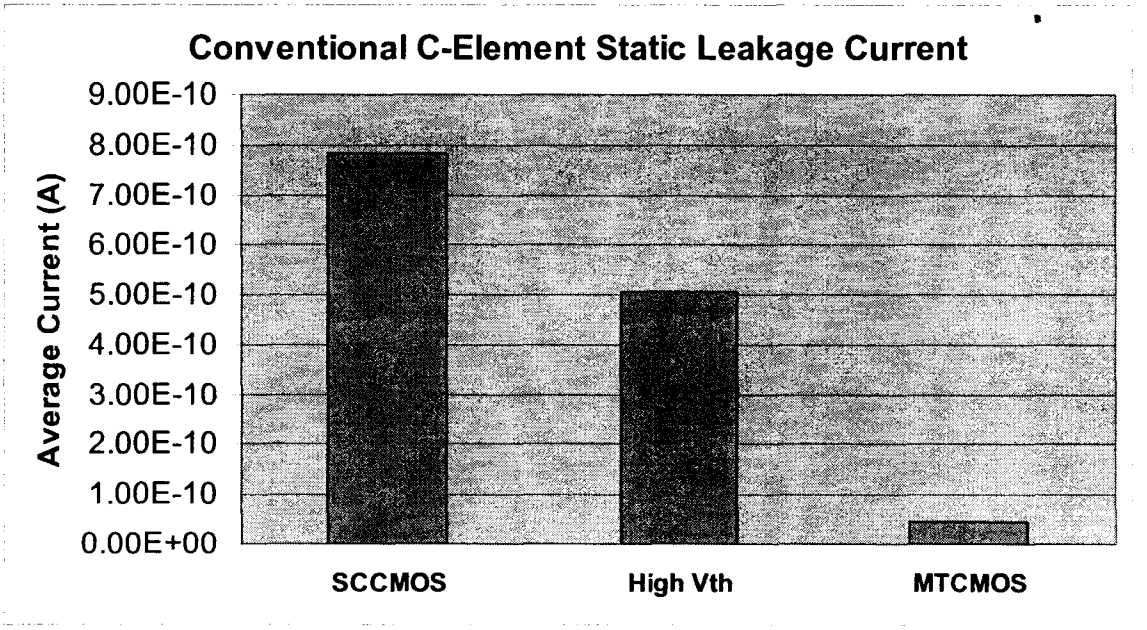

Chart 4.3: Conventional C-Element Static Leakage Current - Close-up

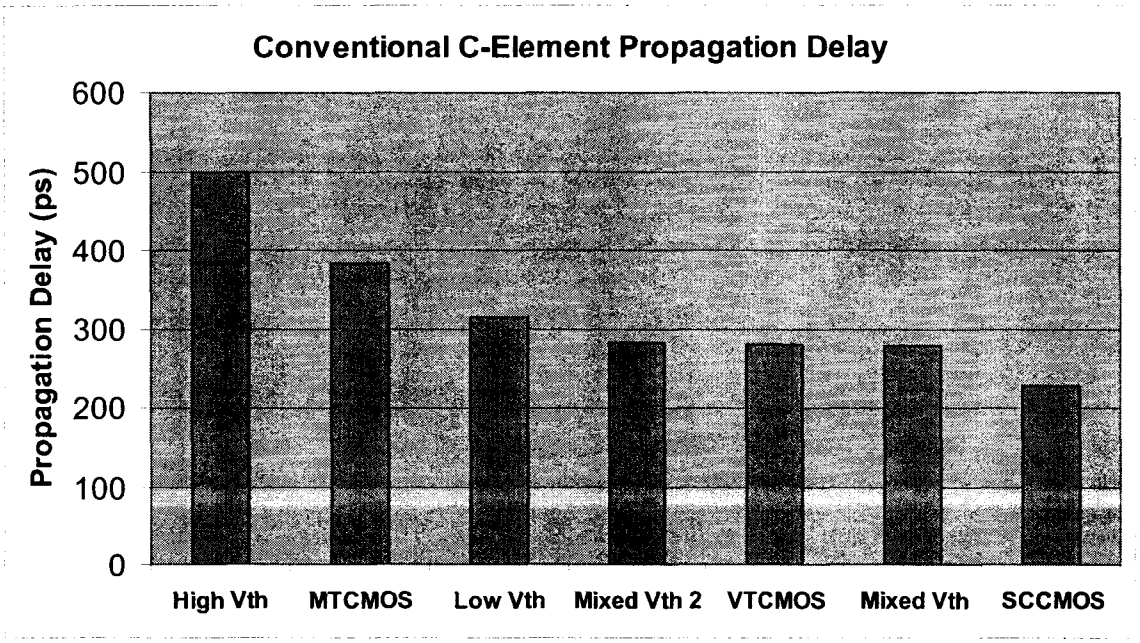

Chart 4.4: Conventional C-Element Propagation Delay

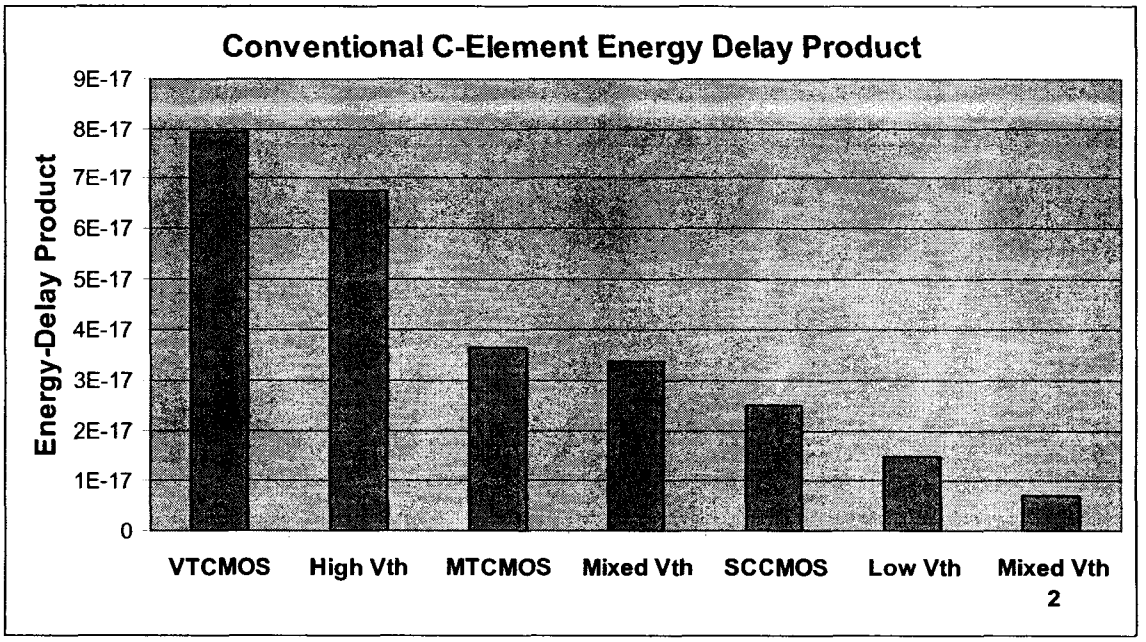

Chart 4.5: Conventional C-Element Energy-Delay Product

4.5.2 Symmetric C-Elements

Table 4.2 presents the results for the Symmetric C-Element circuits. Chart 4.6 shows the average active mode current for each circuit, while Chart 4.7 shows the static leakage current. Because of the large difference between the static leakage current of the various circuits, the high threshold voltage and MTCMOS implementation static leakage do not appear on this chart. Therefore, Chart 4.8 is used to show the static leakage current of these two circuits up close. Chart 4.9 shows the propagation delays of each symmetric C-Element circuit. Finally, Chart 4.10 shows the energy-delay product of each circuit.

Table 4.2: Symmetric C-Elements Simulation Results

Name	Average Dynamic Current	Static Leakage Current	Input to Output Propagation Delay
High V_{TH}	3.01 μA	996 pA	426.31 ps
Low V_{TH}	4.86 μA	1.81 μA	269.92 ps
Mixed V_{TH}	4.90 μA	0.84 μA	268.89 ps
MTCMOS	2.90 μA	643 pA	368.48 ps
SCCMOS	3.81 μA	560 nA	338.26 ps
VTCMOS	5.84 μA	1.38 μA	266.75 ps

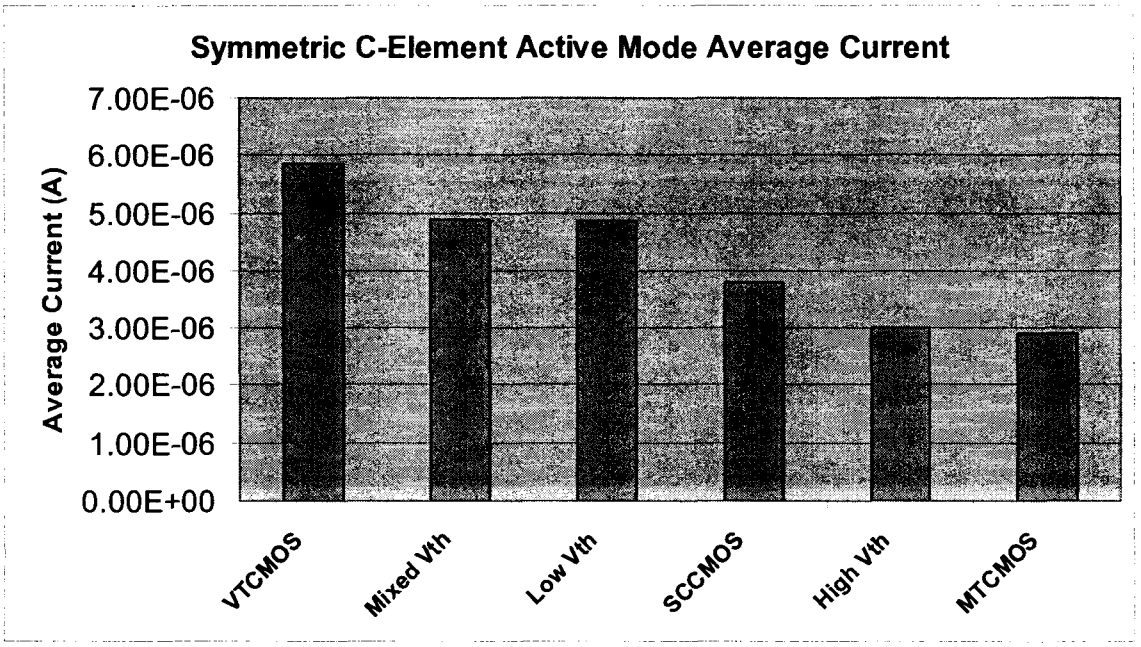

Chart 4.6: Symmetric C-Element Average Dynamic Current

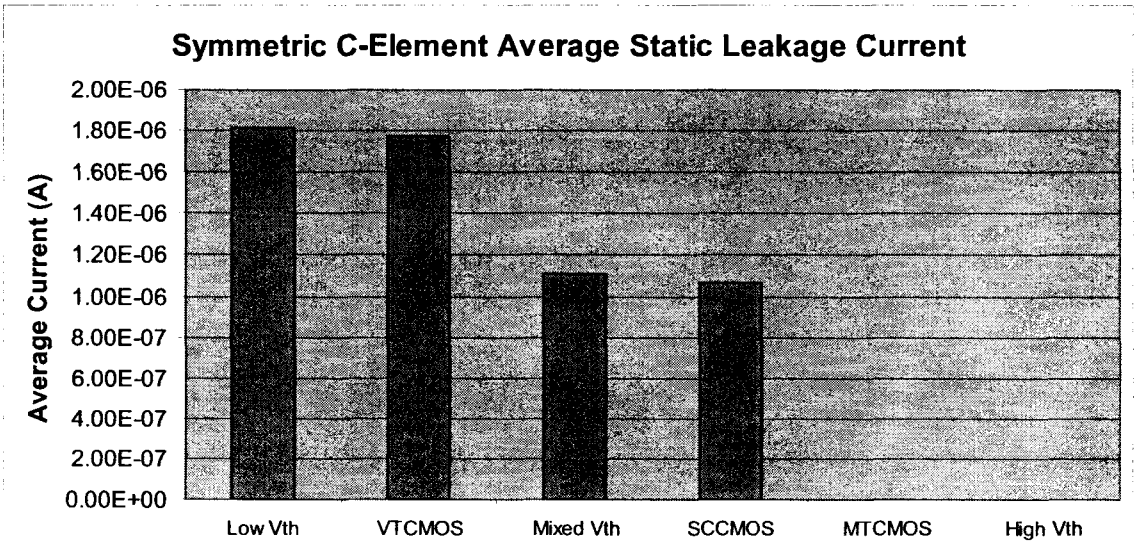

Chart 4.7: Symmetric C-Element Average Static Leakage Current

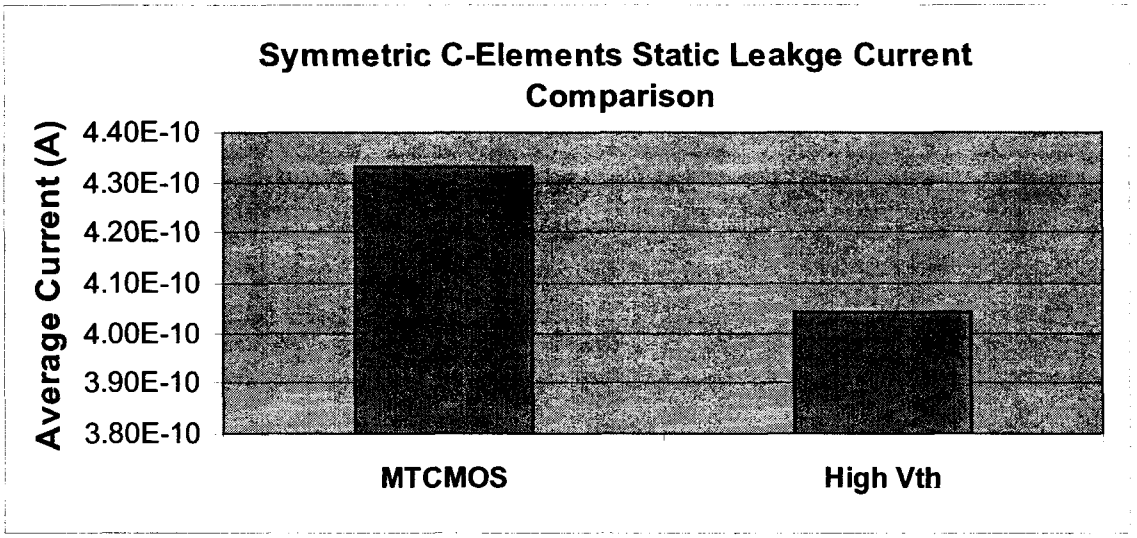

Chart 4.8: Static Leakage Current – MTCMOS vs. High Threshold Voltage Implementations

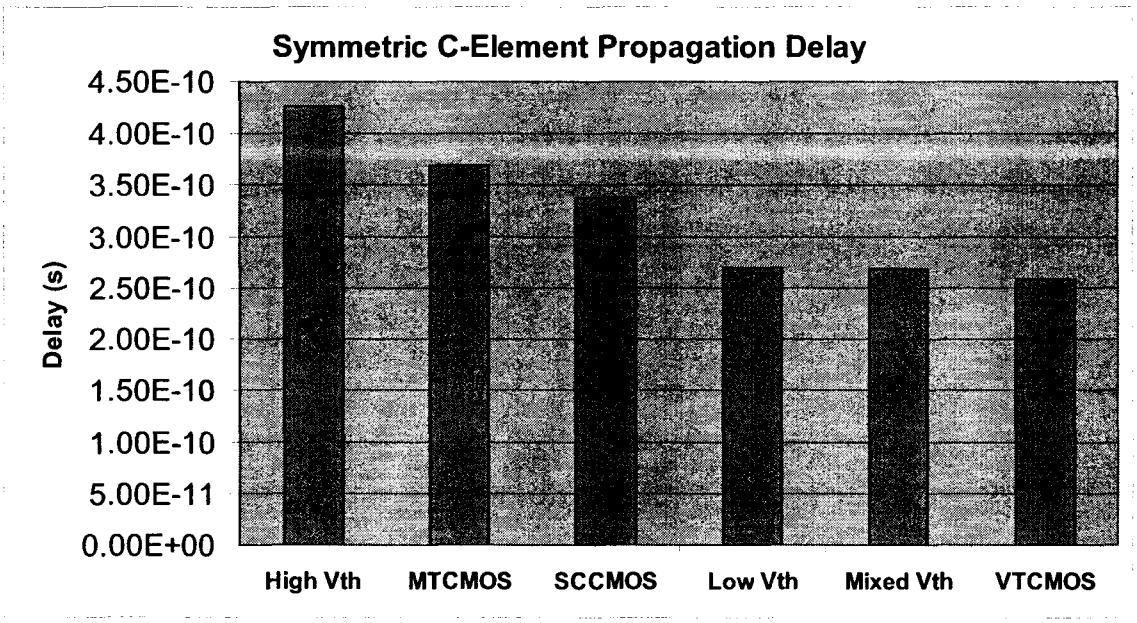

Chart 4.9: Symmetric C-Element Propagation Delay

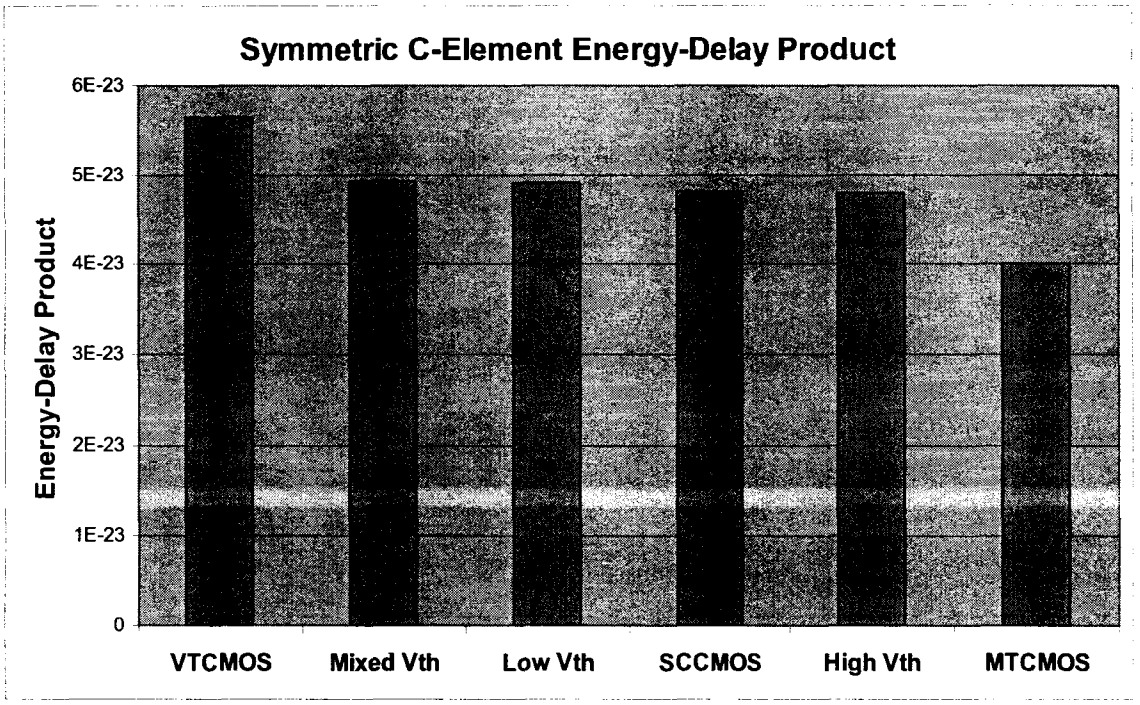

Chart 4.10: Symmetric C-Element Energy-Delay Product

4.5.3 Proposed C-Elements

This section presents the results obtained from some of the proposed improvements to the conventional and symmetric C-Elements, as well as the new C-Elements proposed in this thesis. Table 4.3 presents a summary of the results, followed by a number of charts that further illustrate the results. The first is Chart 4.11, which shows the average dynamic current drawn for each circuit. This is followed by Chart 4.12, showing the static leakage currents and Chart 4.13, which shows a close up of the two smallest static leakage current values. Chart 4.14 shows the propagation delay, while Chart 4.15 shows the energy-delay product of each C-Element.

Table 4.3: Proposed C-Elements

Name	Average Dynamic Current	Static Leakage Current	Input to Output Propagation Delay
Conv. Latched SCCMOS	3.90 *μA*	981 *pA*	317.90 *ps*
Conv. MISCCMOS	2.82 *μA*	451 *nA*	293.83 *ps*
Conv. DSSCCMOS	2.54 *μA*	659 *nA*	343.05 *ps*
Conv. SCCMOS NOR	2.24 *μA*	749 *nA*	354.17 *ps*
Conv. SCCMOS PT NOR	3.16 *μA*	1.02 *μA*	1.01 *ns*
Conv. SCCMOS CPT NOR	1.44 *μA*	1.02 *μA*	560.60 *ps*
Symm. MISCCMOS	3.27 *μA*	52.2 *nA*	299.58 *ps*
Symm. DSSCCMOS	3.77 *μA*	1.67 *μA*	332.10 *ps*
Symm. SCCMOS NOR	2.73 *μA*	1.41 *μA*	333.89 *ps*
Symm. SCCMOS PT NOR	2.94 *μA*	2.58 *μA*	312.36 *ps*
Symm. SCCMOS CPT NOR	4.17 *μA*	837 *nA*	324.81 *ps*
Embedded Sleep C-Element	3.228 *μA*	626 *nA*	423.99 *ps*
Simplified C-Element	2.41 *μA*	167 *nA*	189.08 *ps*
Simplified C-Element 2	1.89 *μA*	914 *pA*	225.27 *ps*
Super Cutoff C-Element	869.4 *nA*	149 *nA*	156.5 *ps*

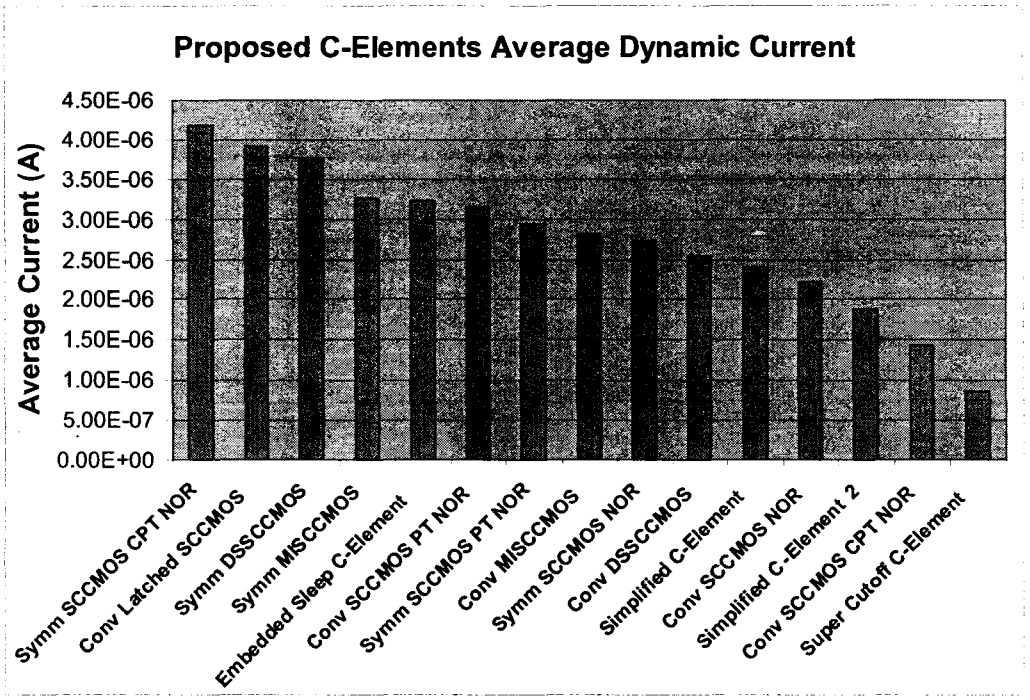

Chart 4.11: Proposed C-Element Average Dynamic Current

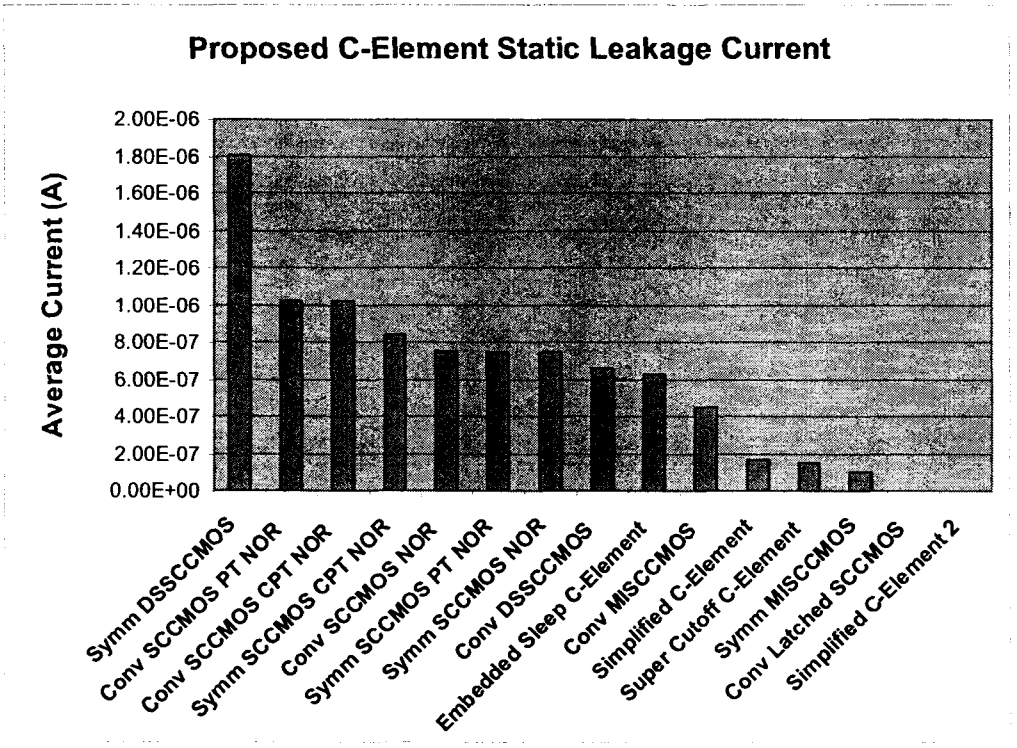

Chart 4.12: Proposed C-Element Average Static Leakage Current

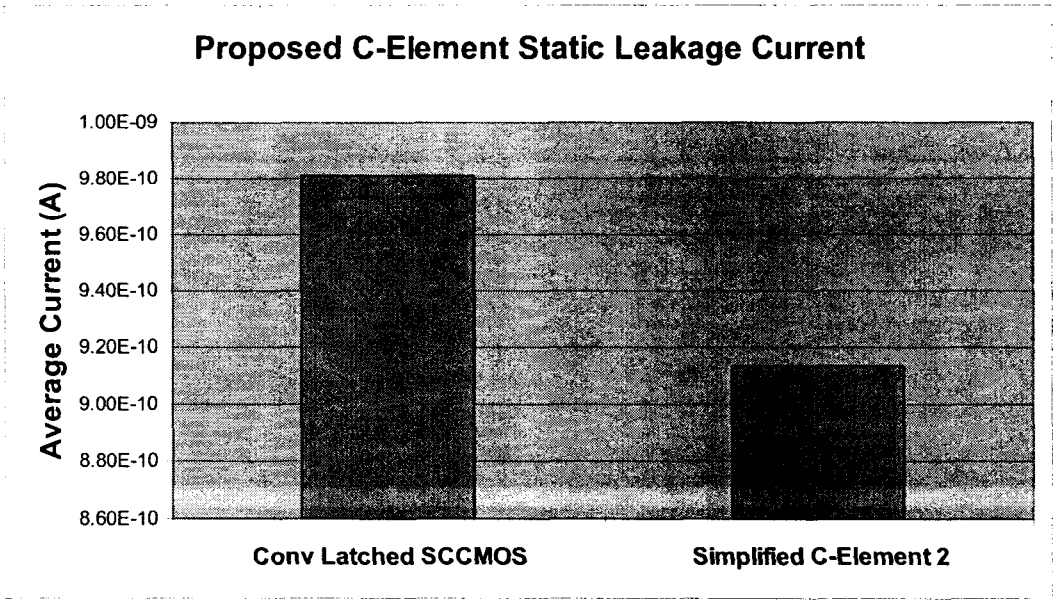

Chart 4.13: Proposed C-Element Average Static Leakage Current Close Up

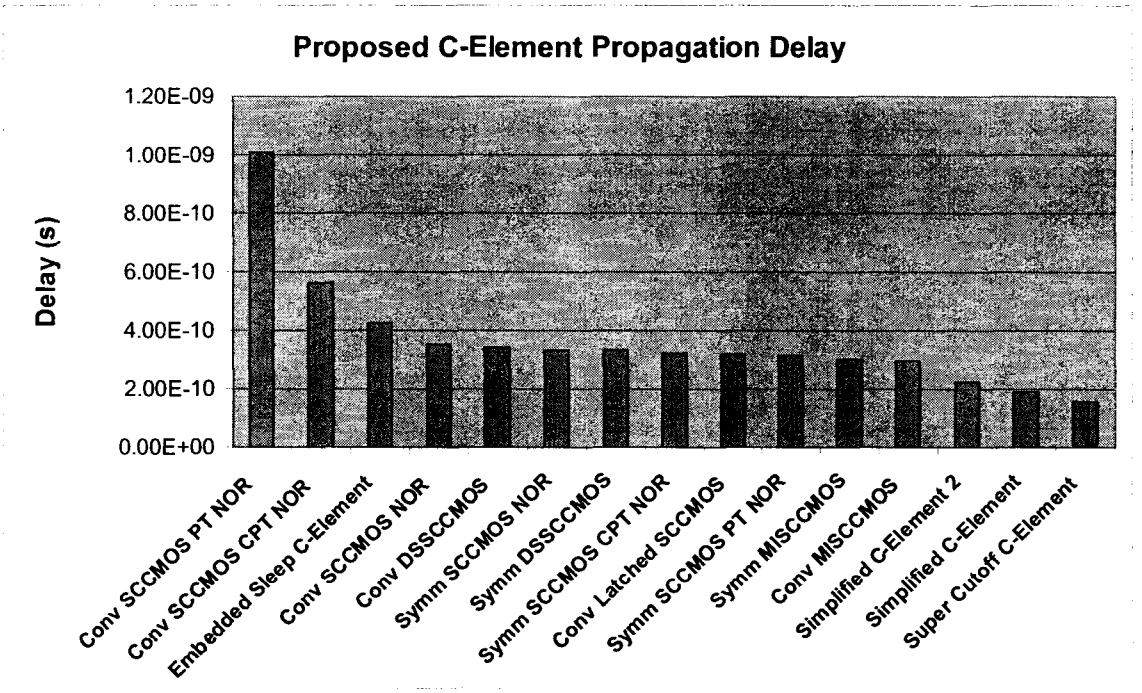

Chart 4.14: Proposed C-Element Propagation Delay

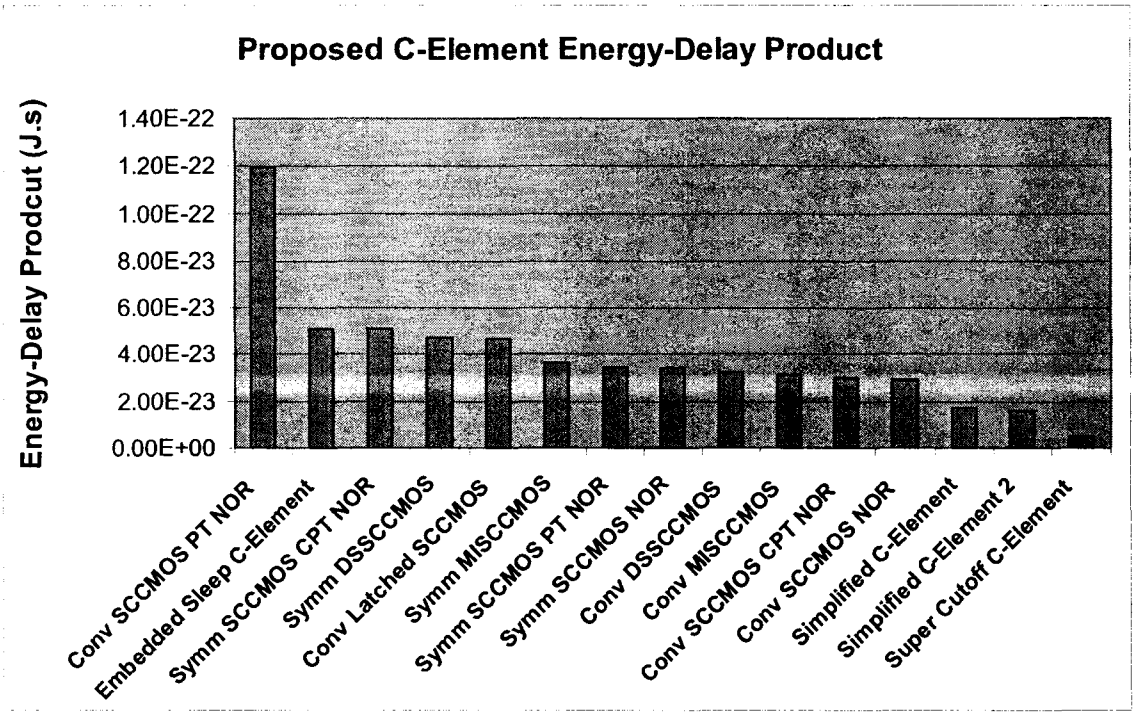

Chart 4.15: Proposed C-Element Energy-Delay Product

4.5.4 Discussion of Results

As a summary of the overall results from the C-Element circuits, Chart 4.16 shows the ten best performing C-Element circuits in terms of active mode energy-delay product. This takes into account the propagation delay and the dynamic power dissipation of each circuit. Chart 4.17 shows the best performing C-Element circuits in terms of static leakage power dissipation and propagation delay. This is represented as the power-delay product (PDP) of each circuit. Figures 4.24 and 4.25 show the simulation waveforms of a rising and falling transition for a selected number of the C-Element circuits.

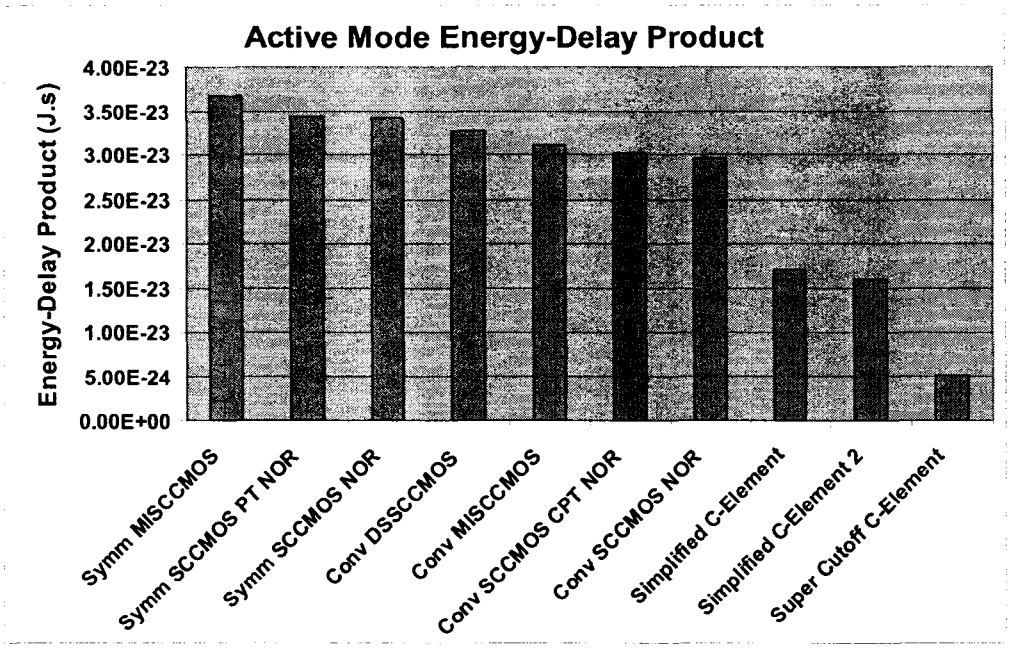

Chart 4.16: Active Mode Energy-Delay Product Comparison

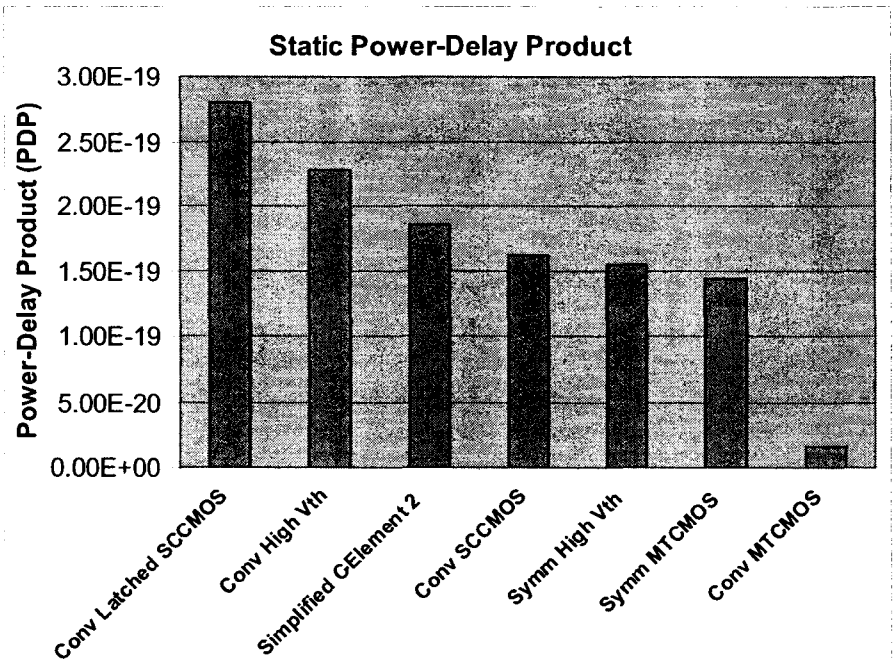

Chart 4.17: Static Power-Delay Product Comparison

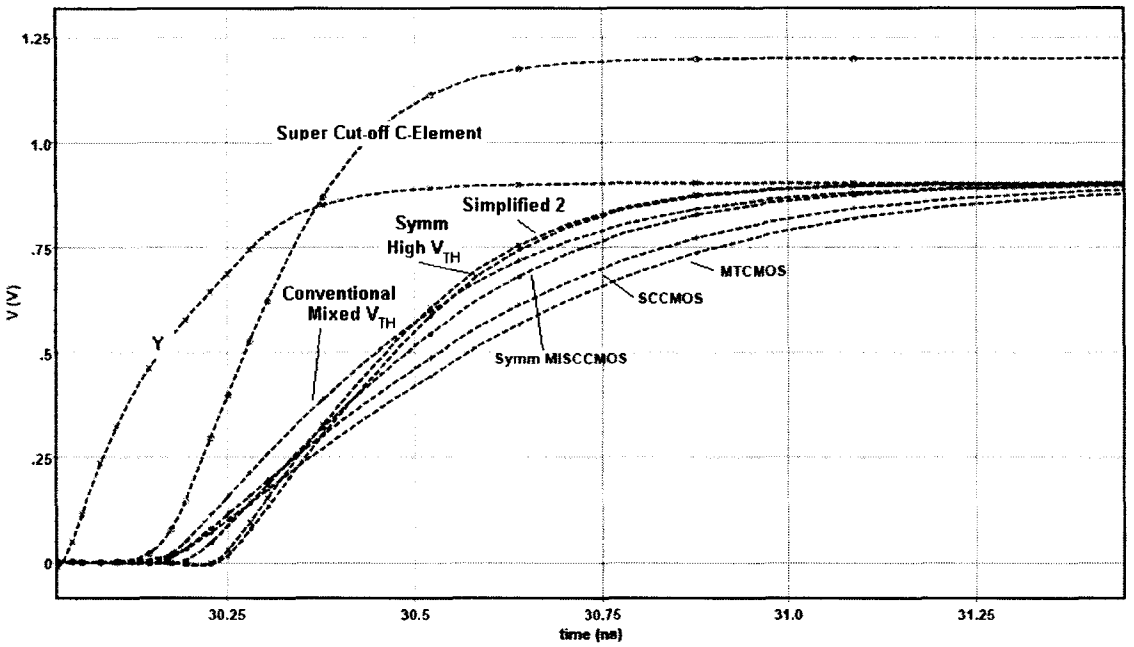

Figure 4.24: Simulation Waveforms of a Low to High Transition of Selected C-Elements

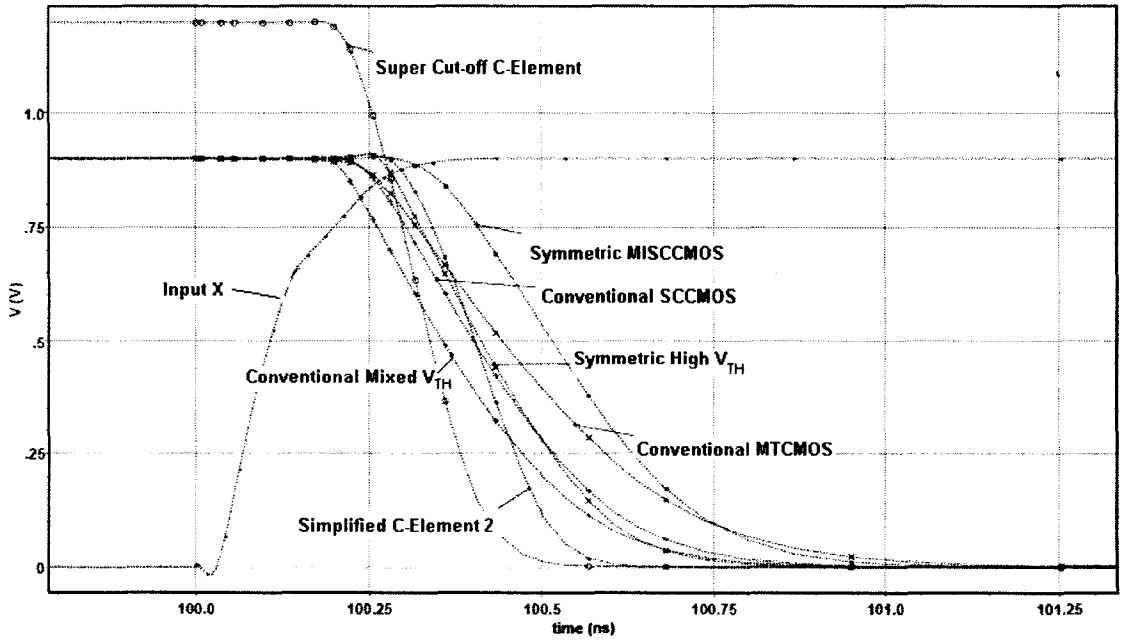

Figure 4.25: Simulation Waveform of a High to Low Transition of Selected C-Elements

The results shown in charts 4.16 and 4.17 summarize what was learned from all of the C-Element circuits attempted. Chart 4.16 ranks the C-Elements based on propagation delay and energy consumption while the circuits are active. Chart 4.17 ranks them based on propagation delay and static power dissipation. The circuits that achieve the lowest EDP and PDP are the one's that provide a good balance of high performance and low active and static power dissipation.

The results for the average switching current were very interesting. For the conventional C-Element, although the high threshold voltage circuit had the second lowest static leakage current, it actually consumed significantly more active switching current than low threshold voltage circuits. This is due to the fact that the switching process is much faster in low threshold voltage circuits, which means there is less time when both the pull up and pull down networks are on simultaneously. This is clearly reflected in the results for the conventional C-Elements.

The MTCMOS technique had an even lower static leakage current than the high threshold voltage implementation of the C-Element. This was due to the fact that the MTCMOS technique had an additional two transistors in series to the C-Element, and they were both high threshold voltage transistors. The effective channel length across the C-Element was therefore increased. Both the MTCMOS and SCCMOS achieved better performance than the high threshold voltage circuit at the same time as limiting static leakage current.

As for the symmetric C-Elements, the low threshold voltage C-Element was actually one of the highest consumers of current in both the active and static states. This

was due to the nature of the circuit, and the latching mechanism which is different than that of the conventional C-Element. The MTCMOS and SCCMOS techniques again were successful at greatly reducing the static leakage current, while providing performance that was better than that of the high threshold voltage circuit.

The proposed C-Element circuits had mixed results. For static leakage current, many of the more complex circuits attempted which added a lot of overhead to the C-Element performed poorly. The techniques that added a large number of transistors for an OR gate simply added too many additional paths for leakage currents that they performed even worse than the low threshold voltage circuit. The MISCCMOS technique performed well, especially in the symmetric C-Element. The second implementation of the Simplified C-Element performed exceptionally well, having the lowest static leakage current of all the proposed techniques.

In terms of propagation delay, the simplified C-Element implementations and the super cutoff C-Element all showed excellent performance and were the three circuits with the lowest energy-delay product of all the proposed circuits. The super cutoff C-Element performed very well, except for its static leakage current consumption. Although it was amongst the lowest consumers, it was still significantly higher than the second implementation of the Simplified C-Element. This was due to the fact that the output inverter in the super cutoff C-Element is implemented using low threshold voltage transistors, and no techniques are used to limit leakage current through it.

When looking at the overall results combining the best circuits from all three categories of C-Elements, it can be said that second implementation of the Simplified C-

Element was the best of the proposed circuits. This is because it combined a low static power dissipation level that is comparable to the high threshold, MTCMOS and SCCMOS C-Elements, while also exhibiting a dynamic energy-delay product that was among the best. When compared with the MTCMOS and SCCMOS C-Elements, this C-Element has two main advantages. The first is that it does not require a sleep control signal to be provided in order to switch into the low static leakage power dissipation mode. Also, the SCCMOS and MTCMOS techniques both have the problem of losing state during sleep mode. The Simplified C-Element does not have this problem.

The fact that the ten best performing C-Element circuits were all multi-threshold implementations clearly illustrates the advantages of using multi-threshold technology in C-Element circuits. Half of all of the transistors in the conventional C-Element are used for latching the output. Multi-threshold technology enables the designer to implement these latching transistors using high threshold transistors while implementing the others as low threshold. This fact alone produces significant improvements in both performance and power dissipation.

4.6 Low Power Supply Voltage Flip-Flops

The following sections introduce the low power supply voltage flip-flops studied in this thesis.

4.6.1 Dual-Edge Triggered Flip-Flops

A latch is a level sensitive circuit. A positive latch is "transparent" when the clock signal is high, and it holds the output value when the clock is low. A negative latch is

"transparent" when the clock signal is low, and holds the output when the clock signal is high. While latches have some applications in digital circuit design, flip-flops are much more widely used. Flip-flops are an edge-sensitive circuit, which means that they sample the input and transfer it to the output at the clock edge. Otherwise, the output is held for the rest of the clock cycle. A positive edge-triggered flip-flop samples on the rising edge of the clock signal, while a negative edge-triggered flip-flop samples on the falling edge of the clock cycle.

Another type of flip-flop, called a dual edge-triggered flip-flop, samples the input on both edges of the clock cycle. This is the type of flip-flop that was used in the asynchronous micropipelines presented in this thesis. Dual edge-triggered flip-flops greatly simplify the design of asynchronous circuits such as the micropipeline discussed in this thesis. This is because the C-Element output is a toggle signal, meaning that it signals an event by changing the output level. The event is signaled by change on the output, which could be a rising or a falling edge. This behavior lends itself very nicely to be used in conjunction with dual edge-triggered flip-flops.

Dual edge-triggered flip flops are also useful in synchronous designs, although they are not as popular as single edge-triggered flip-flops. When dual edge-triggered flip-flops are used, the clock frequency can be cut in half, while maintaining the same functional throughput. This is because although the frequency is cut in half, the sampling is occurring at both edges. A single edge triggered flip-flop that samples on rising edges only would have to be running at double the clock frequency in order to maintain the same number of samples per unit of time. This allows circuits implemented using dual

edge-triggered flip-flops to have twice the throughput if they are run at the same clock frequency, or to maintain the same throughput while halving the clock frequency, which results in large savings in power dissipation in the clock distribution network.

Figures 4.26 and 4.27 show how both single and dual edge-triggered flip-flops are implemented using latches. Single edge triggered flip flops can be implemented using two latches placed in series, while dual edge triggered flip flops can be implemented using two latches placed in parallel, with a multiplexer selecting the output from one of the latches.

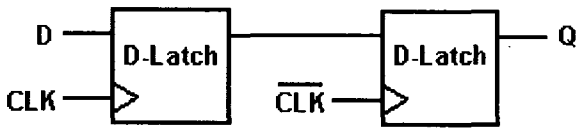

Figure 4.26: Single Edge-Triggered Flip-Flop

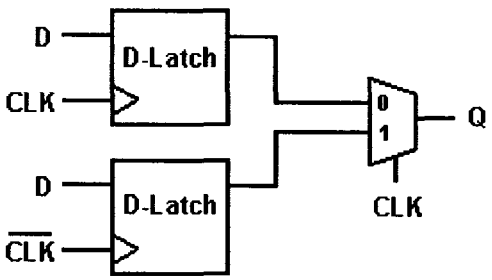

Figure 4.27: Dual Edge-Triggered Flip-Flop

Dual edge-triggered flip-flops were implemented using a number of the circuit techniques discussed previously in this thesis. The circuits are presented in the following sections.

4.6.2 High Threshold Voltage Dual Edge-Triggered Flip-Flop

This is a normal Dual Edge-Triggered (DET) Flip-Flop, with all transistors implemented as high threshold voltage transistors. This circuit will be used as a benchmark to compare with the flip-flops that employed the leakage reduction techniques. Since this circuit was implemented using high threshold voltage transistors, it is expected to exhibit the slowest performance and the lowest levels of static leakage current.

4.6.3 Low Threshold Voltage Dual Edge-Triggered Flip-Flop

This is again a regular DET flip-flop circuit to be used as a benchmark, and it was implemented using low threshold voltage transistors. It is expected to have the highest static leakage current, and the highest performance due to the use of low threshold voltage transistors.

4.6.4 MTCMOS Threshold Voltage Dual Edge-Triggered Flip-Flop

The low power supply voltage techniques discussed in earlier chapters can also be applied to latches and flip-flops. One such latch was proposed in [8], where the MTCMOS technique was applied to the conventional static latch circuit. The operation of this latch was explained in chapter 2. MTCMOS latches were used to implement an MTCMOS dual-edge triggered flip-flop. This circuit is expected to have performance somewhere in between the high and low threshold voltage circuits. It is also expected to have static leakage current almost equal to that of the high threshold voltage circuit.

4.6.5 SCCMOS Threshold Voltage Dual Edge-Triggered Flip-Flop

An SCCMOS latch was developed using the same idea as the MTCMOS latch but using the SCCMOS technique instead. Only two sleep control transistors were used, and the transistors are all low threshold, including the sleep control transistors. An overdrive voltage is used to cut-off leakage currents in sleep mode.

4.6.6 MISCCMOS Threshold Voltage Dual Edge-Triggered Flip-Flop

This DET flip-flop was implemented using an MISCCMOS latch. As discussed earlier, the idea is to eliminate the series transistors added in SCCMOS. Therefore, performance is expected to be slightly better than SCCMOS. However, due to the additional overhead circuitry, static leakage current is expected to be the same or slightly higher.

4.6.7 Simulation Environment

The dual edge-triggered flip-flops were simulated under the same test bench circuit, shown in Figure 4.28. The flip-flop inputs were passed through inverters and the flip-flop Q output was loaded with an inverter as well in order to reproduce a realistic operating environment. The flip-flops were simulated using a 0.9 V power supply voltage. The same clock and data inputs were applied to all circuits for consistency. Figure 4.29 shows a sample simulation waveform including the clock and data inputs and the data output.

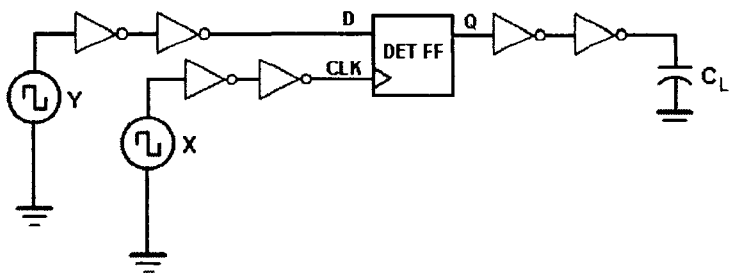

Figure 4.28: MTCMOS DET Flip-Flop Test Bench Circuit

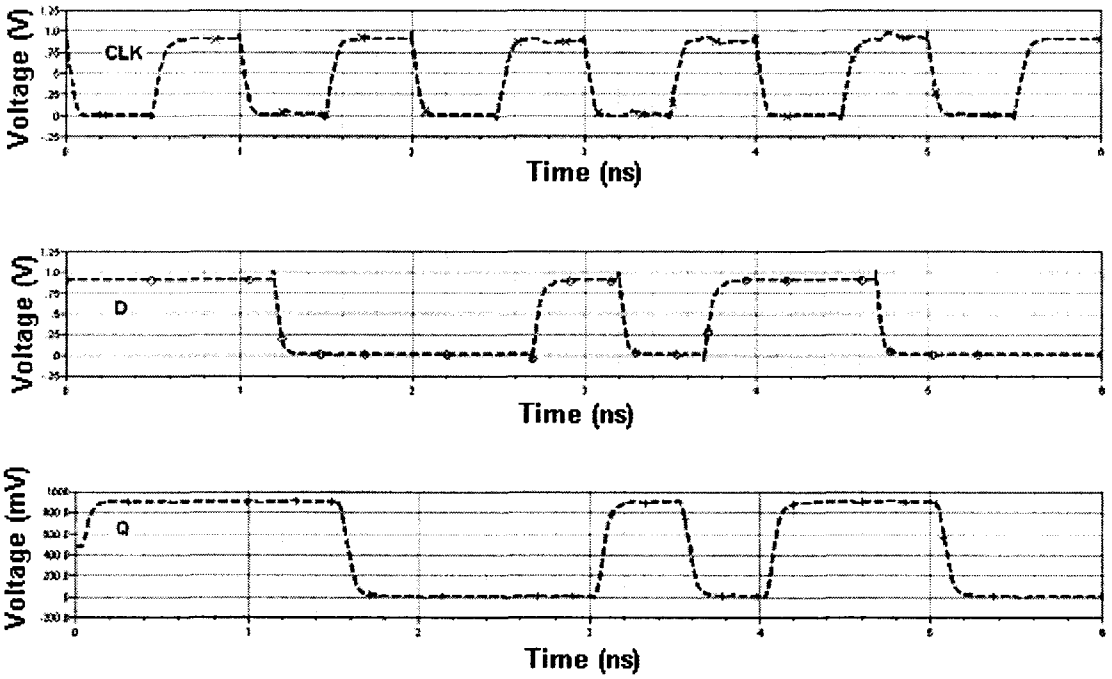

Figure 4.29: MTCMOS DET Flip-Flop Simulation Waveforms

4.6.8 Simulation Results

Table 4.4 shows the simulation results obtained from the dual-edge triggered flip-flops. The charts 4.18, 4.19, 4.20 and 4.21 illustrate the results more clearly. Chart 4.18 shows a comparison of the average dynamic current of each implementation. Chart 4.19 shows the static leakage current, while Chart 4.20 shows a close up comparison between

the MTCMOS and high threshold circuits. Chart 4.21 shows the clk-to-Q delay. Finally, Chart 4.22 shows the average rise/fall time of each implementation.

Table 4.4: Dual Edge-Triggered Flip-Flop Simulation Results

Circuit	Average Current	Static Leakage Current	Clk-to-Q delay	Avg. Rise/Fall time
High V_{TH}	62.2 μA	1.96 *nA*	140.3 *ps*	367.1 *ps*
Low V_{TH}	75.3 μA	4.40 *nA*	71.6 *ps*	220.0 *ps*
MTCMOS	65.6 μA	116.2 *nA*	122.8 *ps*	417.3 *ps*
SCCMOS	57.6 μA	2.52 *nA*	120.8 *ps*	365.7 *ps*
MISCCMOS	80.1 μA	2.42 *nA*	115.9 *ps*	302.7 *ps*

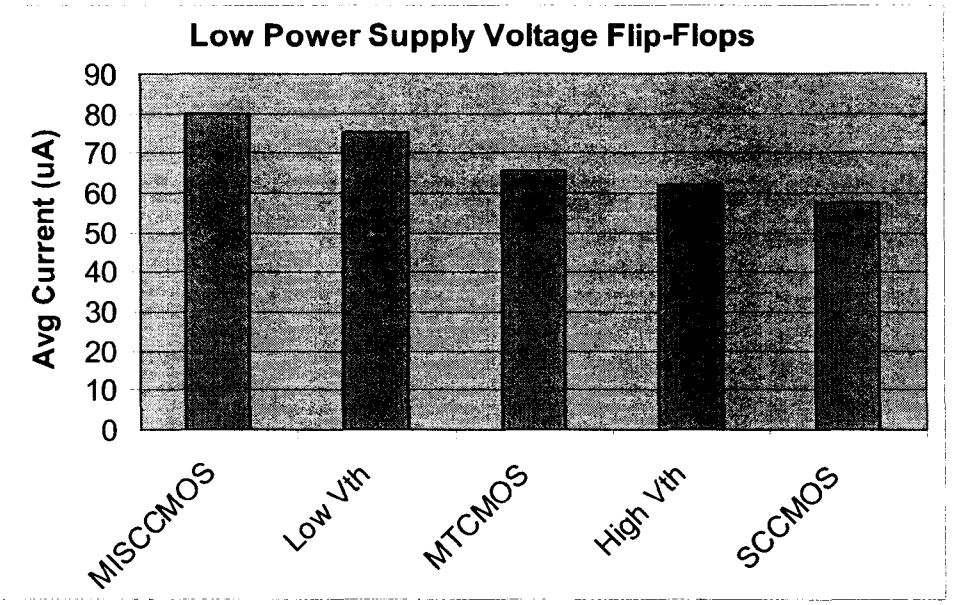

Chart 4.18: Dual Edge-Triggered Flip-Flop Average Dynamic Current

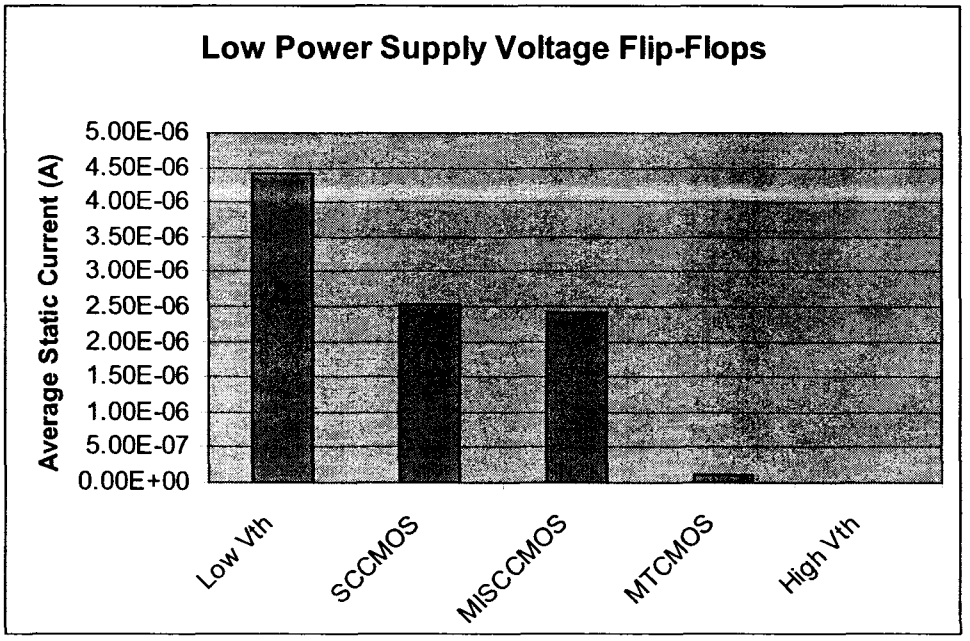

Chart 4.19: Dual Edge-Triggered Flip-Flop Static Leakage Current

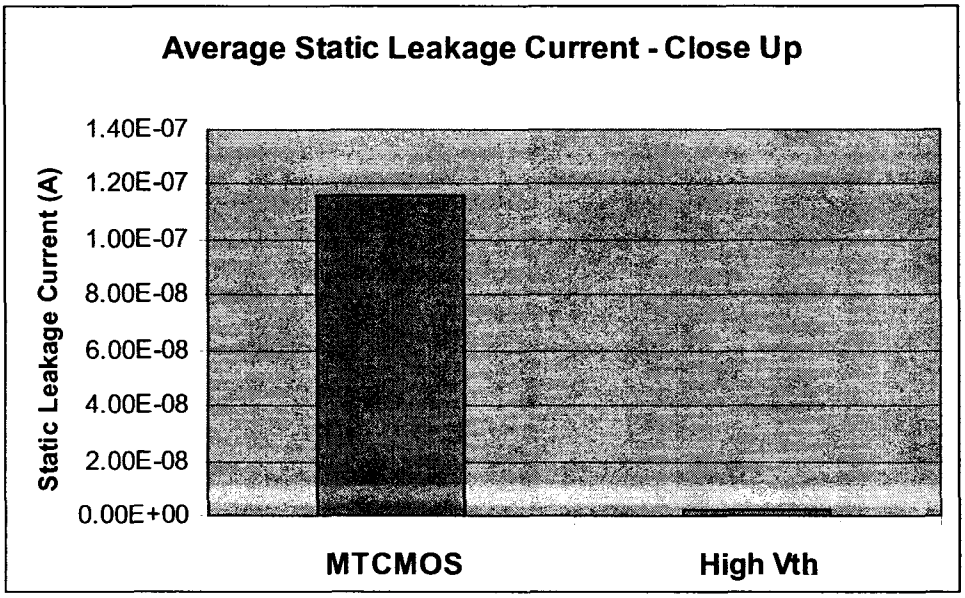

Chart 4.20: Dual Edge-Triggered Flip-Flop Static Leakage Current

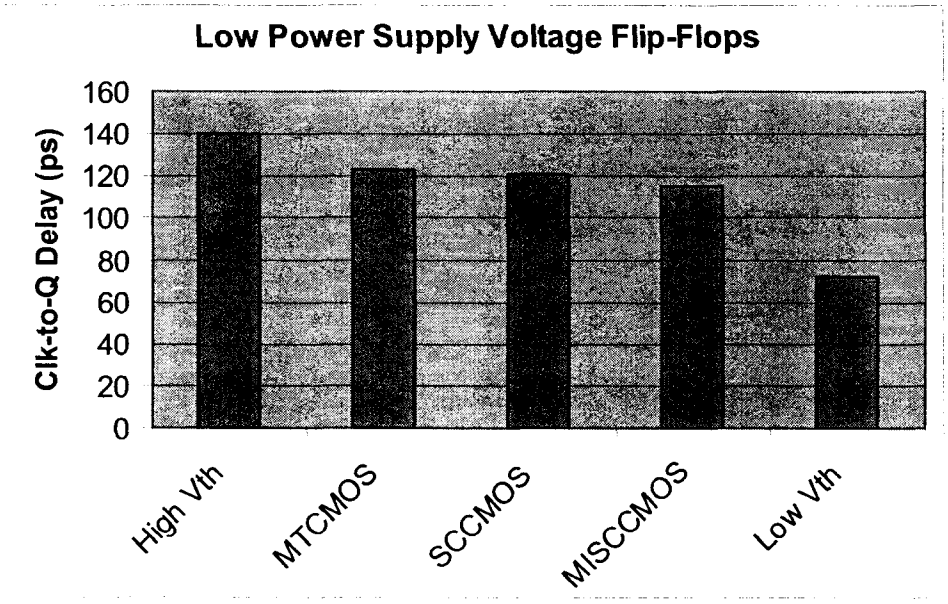

Chart 4.21: Dual Edge-Triggered Flip-Flops Clk-to-Q Delay

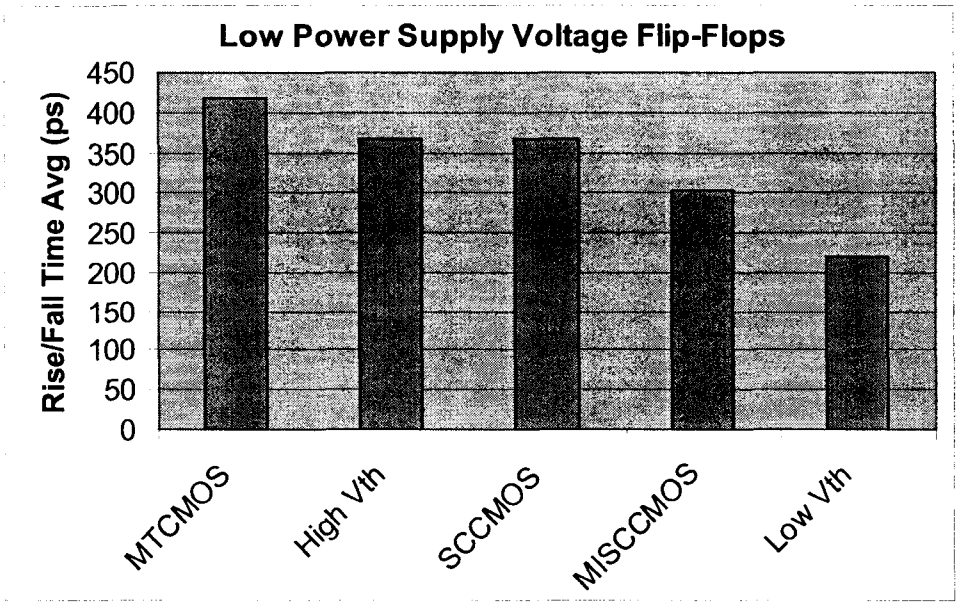

Chart 4.22: Dual Edge-Triggered Flip-Flops Average Rise/Fall Time

4.7 Delay Elements

The following sections introduce the need for delay elements in asynchronous pipelines and present some multi-threshold delay element circuits.

4.7.1 Introduction

The delay element can be implemented in any way, as long as the input presented to it appears at the output after a delay. When the asynchronous micropipeline is implemented with no combinational logic between each stage, it behaves as a simple FIFO. The data input is transferred from one stage to the next, controlled by the C-Elements which determine whether a new data value is available in the previous stage and whether the next stage is ready to accept a new data value. Adding combinational

logic between the stages allows a computation to be performed on the data before it appears at the output.

In synchronous pipelines, the clock period of a pipeline must be made larger than the propagation delay of the slowest combinational logic between any two stages of the pipeline. This ensures that the outputs of the combinational logic at all stages are ready and valid before the clock edge arrives capturing the outputs into the flip-flops. In asynchronous pipelines, however, there is no clock signal, and this is one of the advantages of asynchronous circuits for reasons discussed previously in this thesis. But for asynchronous micropipelines to work, the control signals traveling between the C-Elements must be delayed to allow the combinational logic sufficient time to complete the computation.

One of the advantages of asynchronous circuits is that the delay of each stage can be customized to fit the combinational logic of that stage. There is no need to use a delay equal to that of the slowest combinational logic stage, as there was in synchronous circuits. Faster stages in the micropipeline can have small delay elements, while slower stages can have larger delays. This achieves average overall delay in the pipeline, as opposed to worst case delay in a synchronous pipeline.

The delay elements are usually implemented as a series of inverters, the number of which must be even in order to maintain the signal polarity. The transistor sizing and the number of the inverters in the delay element determine the amount of delay it produces. The ability to use both high and low threshold transistors in the same circuit gives the designer extra flexibility in the design of the delay elements. By selecting which

transistors are high or low threshold voltage transistors, the amount of delay can be controlled as well. High threshold voltage transistors will be slower than low threshold voltage transistors of the same size.

Since this thesis explores multi-threshold circuits, it was found to be useful to examine the various possible multi-threshold voltage delay element circuits and to characterize their delay performance. This will aid multi-threshold asynchronous pipeline designers in selecting the right delay element for each stage in their pipeline designs. The combination of transistor sizing and the use of multi-threshold transistors allow a wide range of options for delay elements. The delay element circuits and the results obtained from simulating them are explored in the following sections.

4.7.2 Multi-Threshold Delay Element Circuits

Four different implementations of the delay element were attempted. The delay elements that were studied consisted of two inverters. These delay elements can then be cascaded for longer delays, according to the requirements of each stage in the micropipeline. The more complex the combinational logic in a stage, the longer delay will be required. For each implementation of the delay element, the delay was also simulated across a range of transistor widths in order to give a wide range of options for delay elements.

4.7.3 High Threshold Voltage Delay Element

This is a simple circuit consisting of two inverters, with all transistors implemented as high threshold voltage transistors. This implementation should give the largest possible delay along with the smallest possible static leakage current.

4.7.4 Low Threshold Voltage Delay Element

All transistors are implemented as low threshold voltage transistors. This circuit is expected to have the smallest possible delay, and high static leakage current. This circuit was simulated mainly for illustration purposes, but also possible for micropipeline stages that are very fast and require a very small delay element.

4.7.5 Mixed Threshold Voltage Delay Element

This implementation of the delay element employed a low threshold voltage PMOS transistor and a high threshold voltage NMOS transistor in the first inverter. The second inverter employed a high threshold voltage PMOS and a low threshold NMOS transistor. This implementation was expected to be a balance between the all high threshold and all low threshold implementations.

This arrangement of transistors introduces a problem by making the overall rise and fall times of the delay element unequal. When the input of the first inverter goes low, the low threshold PMOS in the first inverter will give it a fast rise time, and the low threshold NMOS of the second inverter will give it a fast fall time. So the output of the delay element will fall quite fast. However, when the input of the overall delay element rises, the two transistors that will pull the output are the high threshold NMOS of the first

inverter, and the high threshold PMOS of the second one. This will result in a slow rise time, making it unbalanced compared to the fast fall time.

4.7.6 High Threshold Voltage NMOS transistor Delay Elements

This implementation was an attempt to fix the rise/fall delay imbalance while still having a delay that is somewhere in between the all high threshold and all low threshold implementations. This implementation employed high threshold transistors for the NMOS of both inverters, and low threshold transistors for the PMOS of both inverters. This arrangement ensures that there is no path from the power supply rail to ground through low threshold voltage transistors. It also balances the rise and fall times, since each change of output of the delay element will require action by one high threshold transistor and one low threshold transistor. With proper transistor sizing, the rise and fall times can easily be made equal.

4.7.7 Multi-Threshold Delay Element Simulation Results

Figure 4.30 shows the test bench circuit used to run simulations on the delay element circuits. A parametric transient simulation was run that swept the transistor widths of the delay element. Chart 4.23 shows the average dynamic current of each implementation across a range of transistor sizes. The transistor widths shown are those of the NMOS. The PMOS was always sized twice the width of the NMOS for all inverters in order to maintain equal rise and fall times. The delay obtained from each implementation is shown in Chart 4.24 across a range of transistor sizes. Finally, Chart 4.25 shows the static leakage current of each implementation.

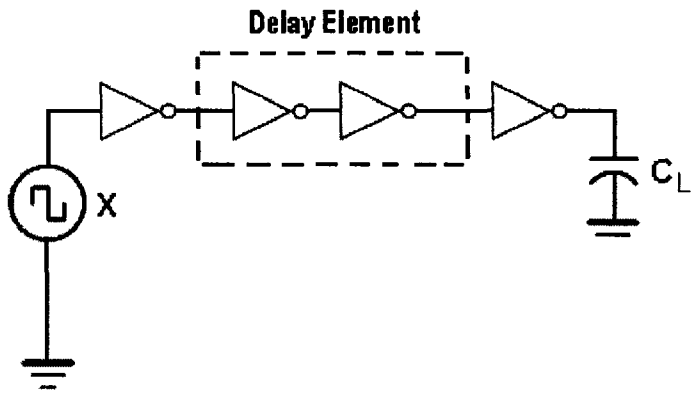

Figure 4.30: Delay Element Test Bench Circuit

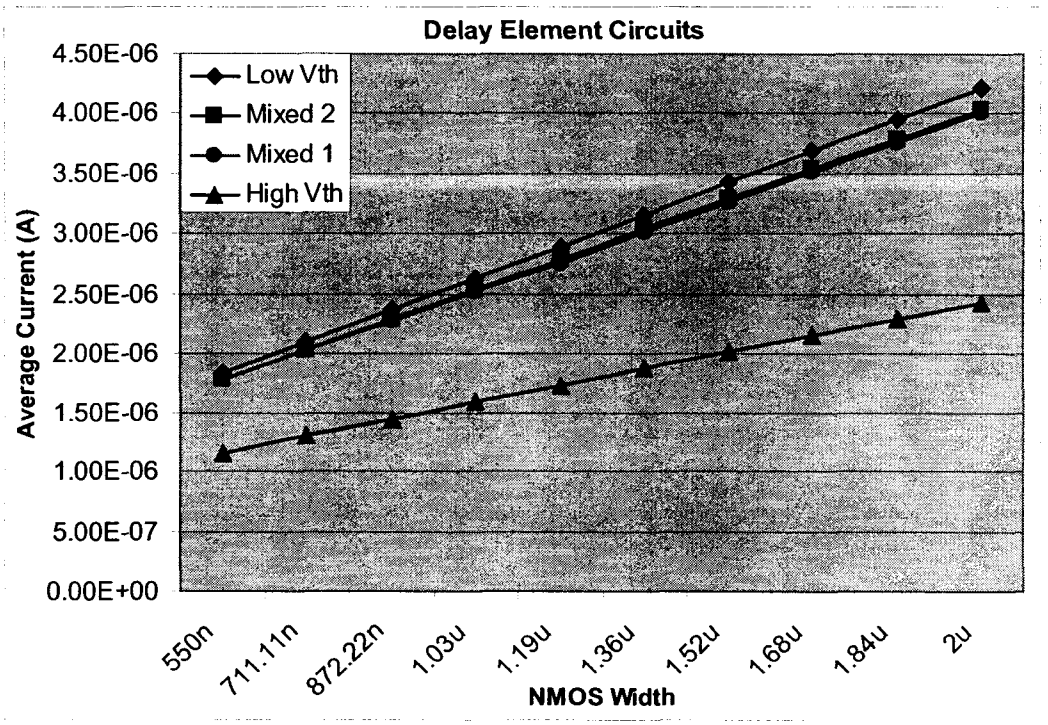

Chart 4.23: Average Dynamic Current of Delay Element Circuits

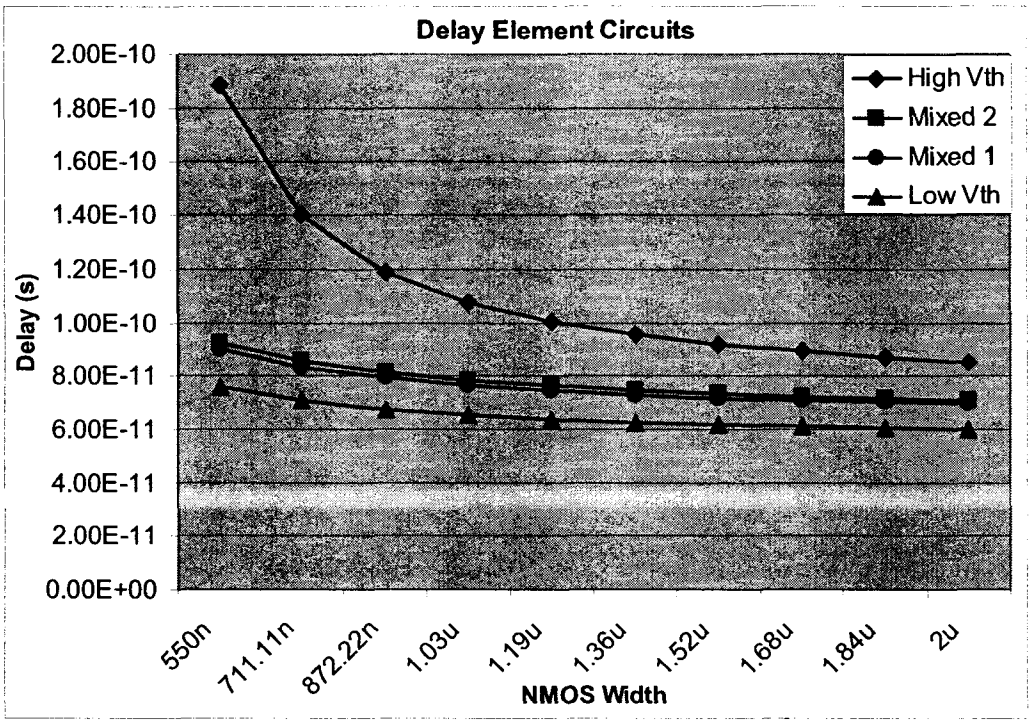

Chart 4.24: Average Delay of Delay Element Circuits

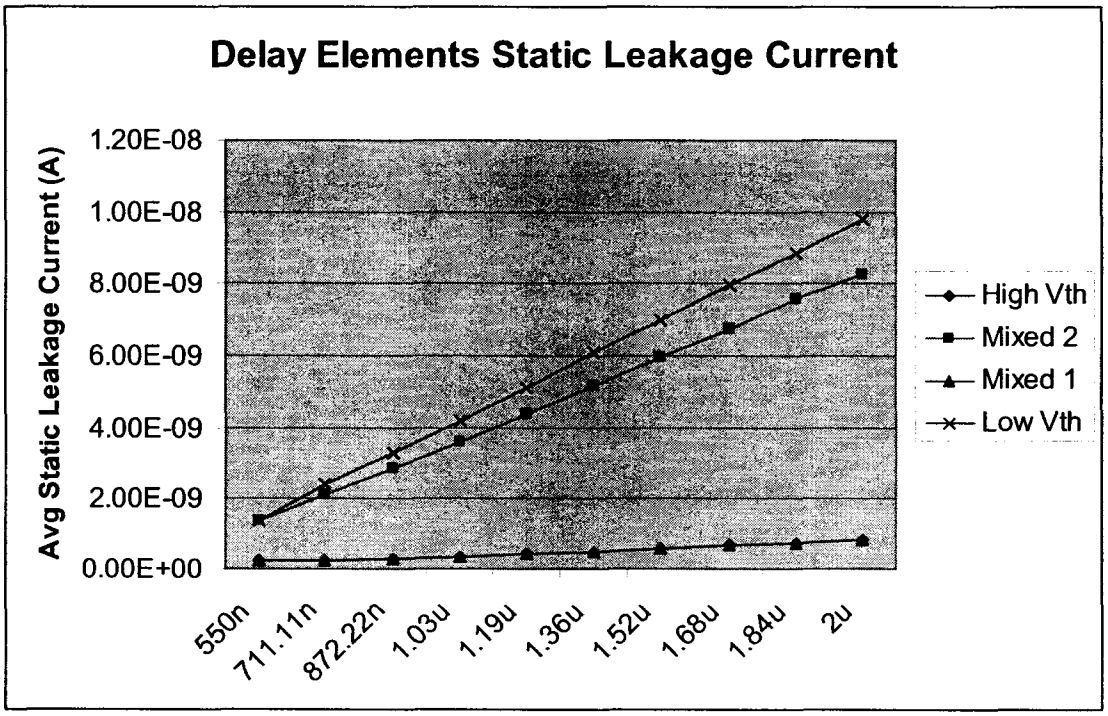

Chart 4.25: Static Leakage Current of Delay Element Circuits

4.7.8 Discussion of Results

The results obtained from the simulations run on the delay elements were as expected. Chart 4.23 shows that as the width is increased, the average current increases as well. It also shows that the low threshold voltage delay element had the highest current draw, followed by the mixed threshold voltage delay elements. The high threshold voltage delay element had the least amount of current draw. The plots also show the linear relationship between current and transistor width, which is expected.

Chart 4.24 also shows the expected results. The low threshold delay element had the smallest delay, followed by the mixed threshold delay elements. The high threshold voltage delay element had the largest delay. The delay was also shown to increase as the transistor widths are decreased. This is expected because the smaller the transistor width and lower the current through the transistors, and the longer it takes to charge and discharge the output capacitance.

The delay chart presents the micropipeline designer with an illustration of the transistor widths required for a particular delay. It also illustrates the effect of using transistors with different threshold voltages on the delay and power dissipation of the delay elements. These results were very valuable in gaining insight into delay elements, and the effect of using multi-threshold voltage transistor techniques. Multi-threshold transistors along with transistor sizing give the designer maximum flexibility in designing delay elements with the required delay. If the process being used has the capability to

have transistors of multiple thresholds in the same integrated circuit, then using multi-threshold techniques on delay elements becomes very beneficial.

4.8 GasP Pipelines

The GasP pipeline mentioned in a previous chapter was also studied during this research. The nature of the GasP circuit is different than that of C-Elements and other conventional CMOS circuits. The GasP pipeline is a hand-crafted custom circuit that is not a simple conventional CMOS circuit. Therefore, it was more difficult to apply a lot of the low power techniques that were applied to the C-Element. However, several different types of GasP elements were simulated in order to gain more insight into this type of asynchronous pipeline.

4.8.1 High Threshold Voltage GasP Element

This variation of the GasP Element was implemented using only high threshold voltage transistors. The aim of this circuit was to provide a benchmark circuit to compare the remaining techniques with. This circuit should provide the worst case delay and best case static leakage current due to the use of high threshold voltage transistors. The circuit diagram is shown in Figure 4.31.

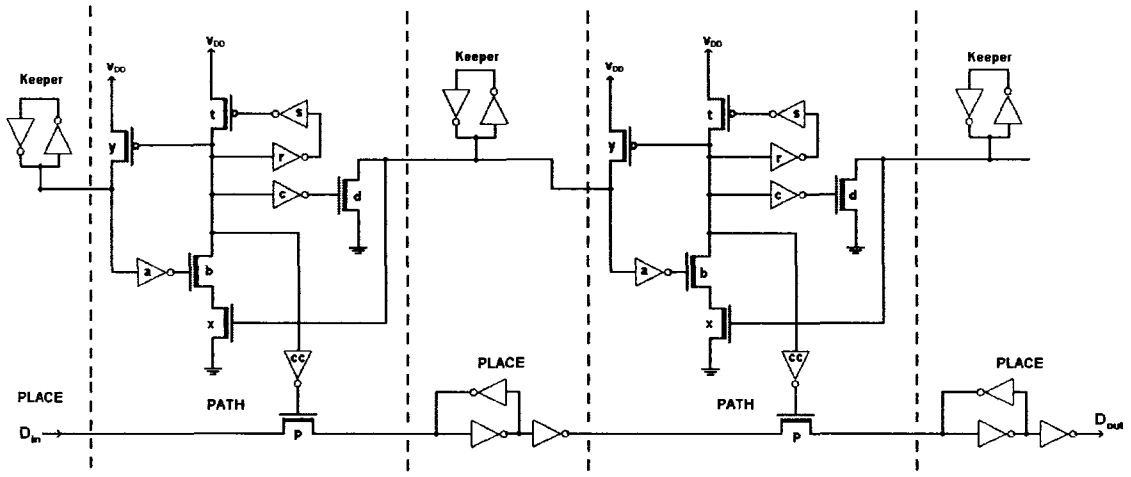

Figure 4.31: GasP Circuit Diagram

4.8.2 Low Threshold Voltage GasP Element

The low threshold voltage implementation of the GasP element was also simulated to be used as a benchmark circuit. This circuit should have the worst case static leakage current and best case delay performance. This is due to the circuit structure, which provides for many leakage paths through the circuit.

4.8.3 Mixed Threshold Voltage GasP Element

For this implementation, the GasP control circuitry was studied very closely and understood very well. The role of each transistor was identified, and the transistors were classified as those that affect the delay through the control circuitry and those that are there for other purposes. The transistors in the critical path of the circuit were then implemented as low threshold voltage transistors, while the others were implemented as high threshold transistors. The same analysis was also very helpful in transistor sizing.

The operation of the GasP control circuitry and a discussion of each transistor was the subject of a previous chapter in this thesis, and will not be repeated here. This implementation used the same circuit structure, but with careful selection of which transistors are high threshold voltage and which ones are low threshold transistors. From the circuit diagram it is clear to see that transistors y, b, x, d and inverters a and c are all involved in the transfer of the state from the previous stage to the next stage. Therefore, in order for the GasP element to have a small delay across the control circuitry, these transistors and inverters need to be implemented using low threshold voltage transistors.

Transistor p is in the data path and it transfers the data between stages. Therefore, it must also be a low threshold voltage transistor to minimize the delay across the pipeline. Inverter cc turns transistor p on and off, and it must also be fast so it should be implemented as a low threshold voltage circuit. On the other hand, inverters s and r and transistor t are only used to provide a delay before resetting the circuit, and so they do not need to be fast. In fact, using high threshold voltage transistors may actually help to achieve the desired delay.

Therefore, the mixed threshold voltage GasP element and pipeline was implemented according to this analysis. In terms of leakage current paths, the inverters a, c and cc will all have high static leakage current flowing through them since they are low threshold voltage circuits. There is a potential leakage path through transistor p, unless the data latch between GasP stages is implemented using high threshold voltage transistors. This is in fact desirable to minimize static leakage current through the data

latch itself. This has some effect on the delay through the data path, however this should be minimal if the data latch transistors are sized small such that they do not present much resistance when transistor p is on and trying to overwrite the data latch.

Since transistor t is a high threshold voltage transistor, it helps to minimize static leakage current through the transistors b and x. There is also a leakage path between transistor y of a stage and transistor d of the previous stage. This is unfortunately unavoidable without major changes to the circuit structure.

4.8.4 MTCMOS GasP Element

The MTCMOS technique was applied to the GasP circuit to cut off static leakage current through the entire circuit when the circuit is in sleep mode. This was done by connecting the power supply and ground to the GasP circuit through the sleep control transistors. In sleep mode, the circuit is completely isolated from the powers supply and ground rails, which helps to minimize static leakage current. MTCMOS was applicable to the GasP pipeline without any major changes to the circuit structure, which makes it a good candidate for a low power circuit technique to be applied to GasP.

4.8.5 SCCMOS GasP Element

This circuit is similar to the MTCMOS GasP circuit, with the exception that it uses the SCCMOS technique to cut off the static leakage current. The power supply rail is connected to the circuit through a sleep control transistor, which is implemented as a low threshold voltage transistor. The sleep control signal is multiplexed between ground in normal operation and an overdrive voltage in sleep mode.

4.8.6 Transmission Gate GasP Element

During the study of the GasP circuit, it was noticed that the pass transistor p can cause an issue for the data latch. Since the pass transistor is an NMOS transistor, it can pull down the output well. However, if the input data is high and the pass transistor is turned on, the output of the pass transistor will reach a maximum of $V_{in} - V_{th}$. This is due to the threshold voltage drop across the NMOS transistor. Therefore, the output swing of the pass transistor is 0 to $(V_{in} - V_{th})$, and this is the input range of the first inverter of the data latch. Since the input of the first inverter only goes up to $V_{DD} - V_{th}$, the PMOS transistor of the first inverter in the data latch will not be fully turned off, which causes an increase in the static leakage current flowing through that inverter.

This can be easily fixed by using a transmission gate instead of a simple pass transistor. This adds an overhead of one transistor per stage, but it eliminates the problem since the output swing of the transmission gate is 0 to V_{DD}. This technique was found to slightly reduce the static leakage current of the overall GasP circuit.

4.8.7 Simulation Results

Figure 4.32 shows the test bench circuit used to run the GasP element simulations. GasP elements were used at the inputs and outputs of the GasP element under test in order to make the simulation realistic. The output state of the final GasP was tied high to simulate the next stage always being empty, while an input waveform simulated incoming state transitions at the state input. An input waveform was also connected to the

data input. Figure 4.33 is a simulation waveform that shows the input and output states of the GasP element under test.

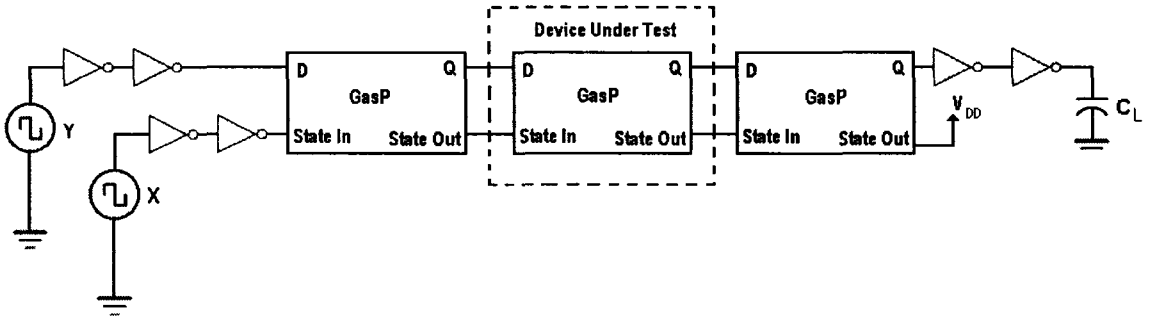

Figure 4.32: GasP Test Bench Circuit

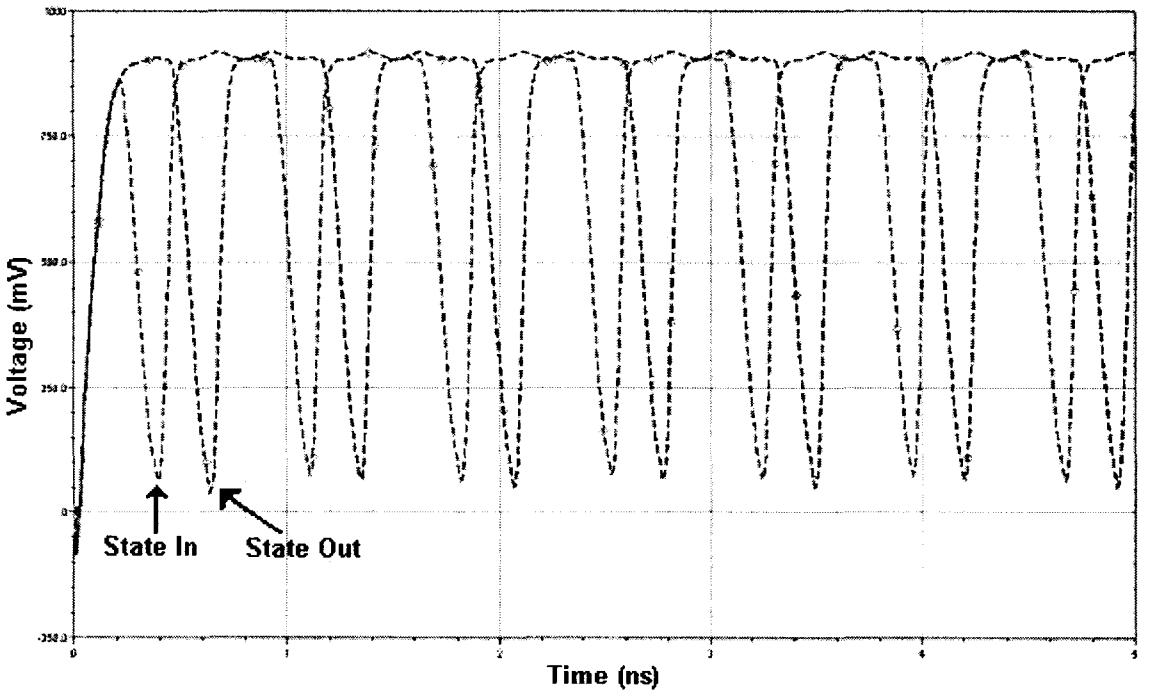

Figure 4.33: GasP State Simulation Waveform

Table 4.5 shows the simulation results obtained from the GasP element simulations. Charts 4.24, 4.25 and 4.26 illustrate the results more clearly. The results closely match what was expected, which is that the high threshold voltage transistor implementation had lower average current than the low threshold voltage implementation, while the mixed threshold voltage implementation was somewhere in the middle. The low threshold voltage implementation was the faster than the high threshold voltage implementation, while the mixed threshold voltage implementation again came in the middle.

The more interesting results came from the MTCMOS and SCCMOS circuits. These showed even lower average current drawn than the high threshold voltage implementation. This can be explained by the addition of the sleep control transistors, which even when they are turned on limit the amount of current that can pass through. This is also reflected in the delay performance of these implementations, which performed not much better than the high threshold voltage implementation.

Table 4.5: GasP Simulation Results

Circuit	Average Current	Input to Output Propagation Delay (State_in to D_{out})
High V_{TH}	142.6 μA	327.49 *ps*
High V_{TH} *(TG)*	142 μA	273.4 *ps*
Low V_{TH} *(TG)*	235 μA	159.67 *ps*
Mixed V_{TH}	197.4 μA	201.9 *ps*
MTCMOS	111.6 μA	245 *ps*
SCCMOS	108.6 μA	240.37 *ps*

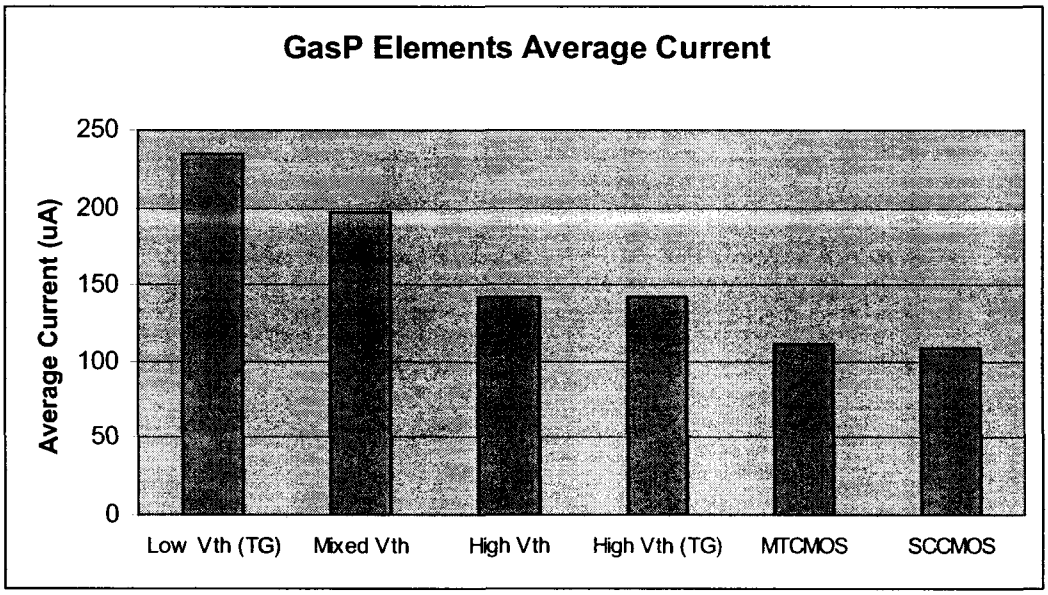

Chart 4.26: GasP Elements Average Dynamic Current

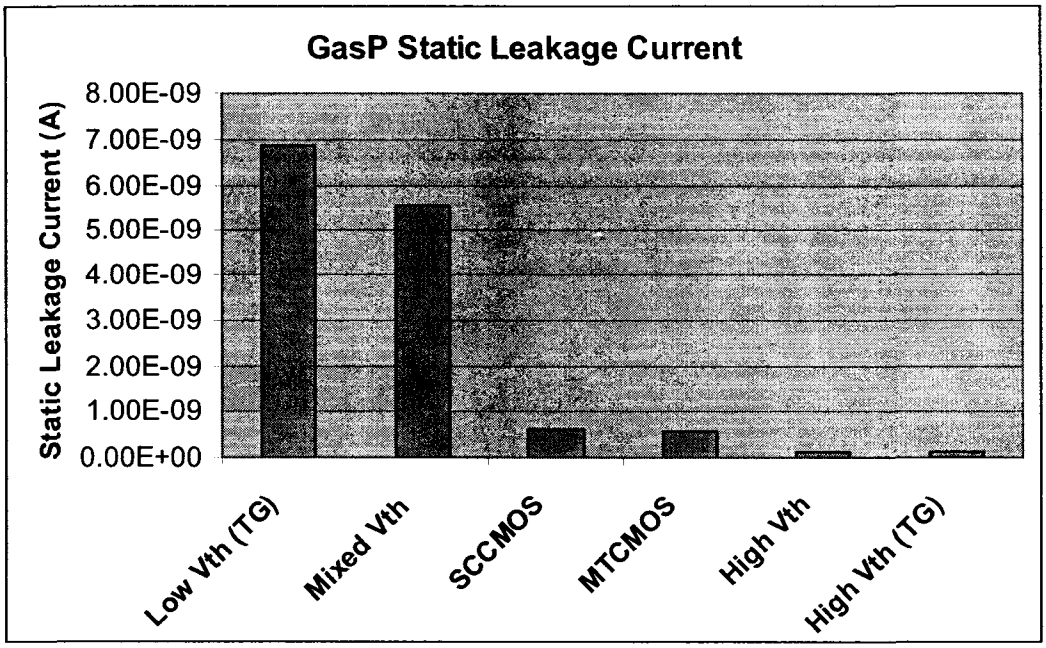

Chart 4.27: GasP Elements Average Static Leakage Current

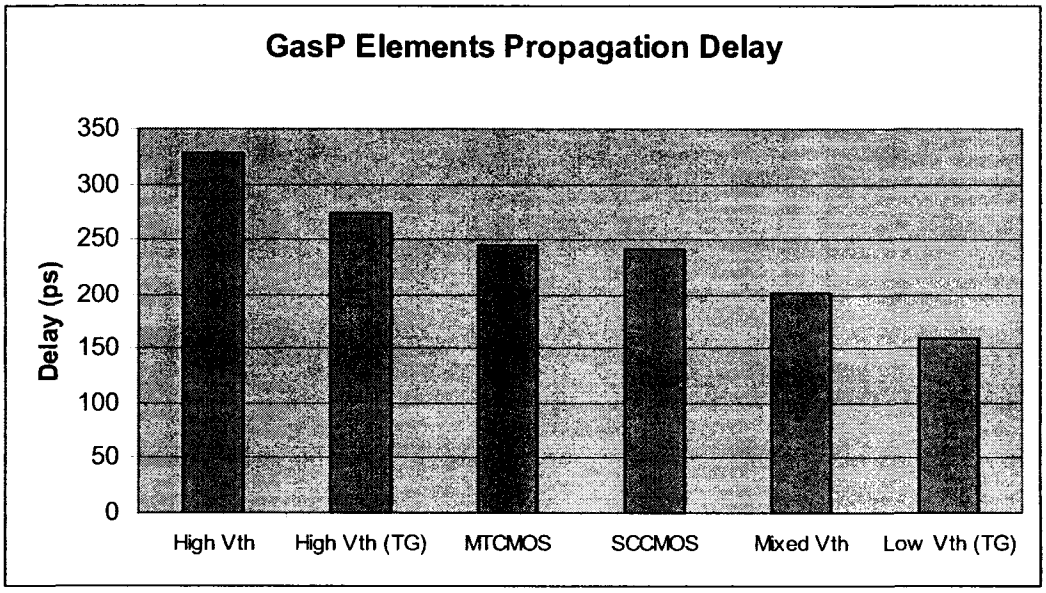

Chart 4.28: GasP Elements Propagation Delay

CHAPTER 5

Multi-Threshold Asynchronous Pipelines

5.1 Introduction

The previous chapter consisted of a detailed investigation of the performance of individual asynchronous pipeline primitives. In this chapter, these primitives are used together to construct full asynchronous pipelines and examine their behavior. The pipelines presented in this chapter are FIFO circuits which contain no combinational logic between pipeline stages.

5.2 Simulation Environment

Each pipeline simulated consisted of five stages internally. All simulations used the same Data input and Request input waveforms, and were all run at 0.9 V power supply voltage. Those micropipelines that needed an overdrive voltage were supplied with a second 1.2 V power supply. The test bench circuit used is shown in Figure 5.1.

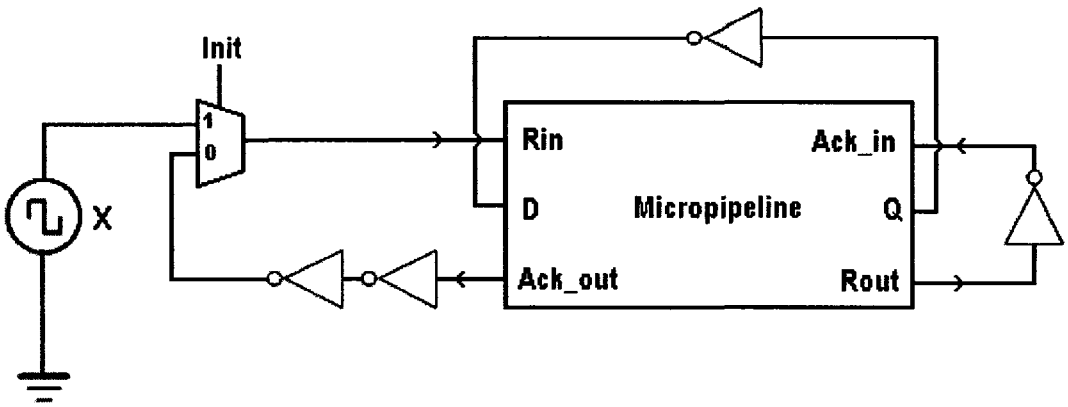

Figure 5.1: Micropipeline Test Bench Circuit

5.3 Symmetric C-Element Micropipelines

Table 5.1 shows the simulation results obtained from the micropipelines designed using the symmetric C-Elements discussed in the previous chapter. Charts 5.1, 5.2 and 5.3 illustrate the results further. Figure 5.2 shows sample clock and data output simulation waveforms for the third stage of the pipeline.

Table 5.1: Symmetric C-Element Asynchronous Micropipeline

Circuit	Average Dynamic Current	Clk-to-Q Delay / Stage	Rin-to-Q Delay
High V_{TH}	114.4 μA	177.8 *ps*	310.3 *ps*
Low V_{TH}	136.7 μA	78.32 *ps*	192.8 *ps*
Mixed V_{TH}	96.68 μA	171.5 *ps*	279.7 *ps*
MTCMOS	174.7 μA	204 *ps*	357.9 *ps*
SCCMOS	165.3 μA	133.2 *ps*	245.6 *ps*
MISCCMOS	136.1 μA	80.8 *ps*	225.1 *ps*
Conv. NOR SCCMOS	48.8 μA	93.73 *ps*	175.46 *ps*
CPT NOR SCCMOS	55.1 μA	97.9 *ps*	190.89 *ps*

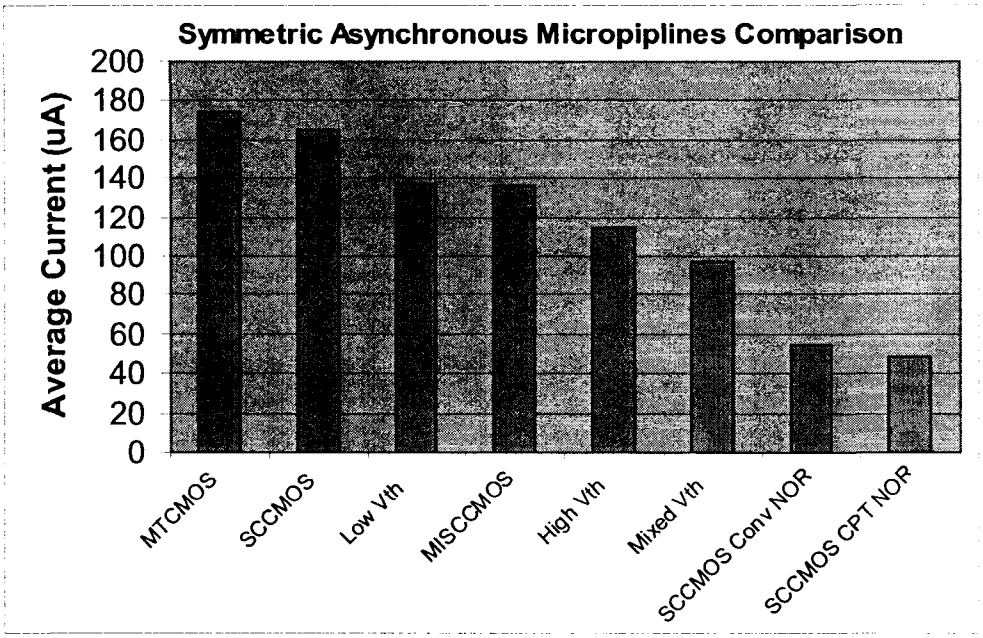

Chart 5.1: Symmetric C-Element Micropipeline Average Current Simulation Results

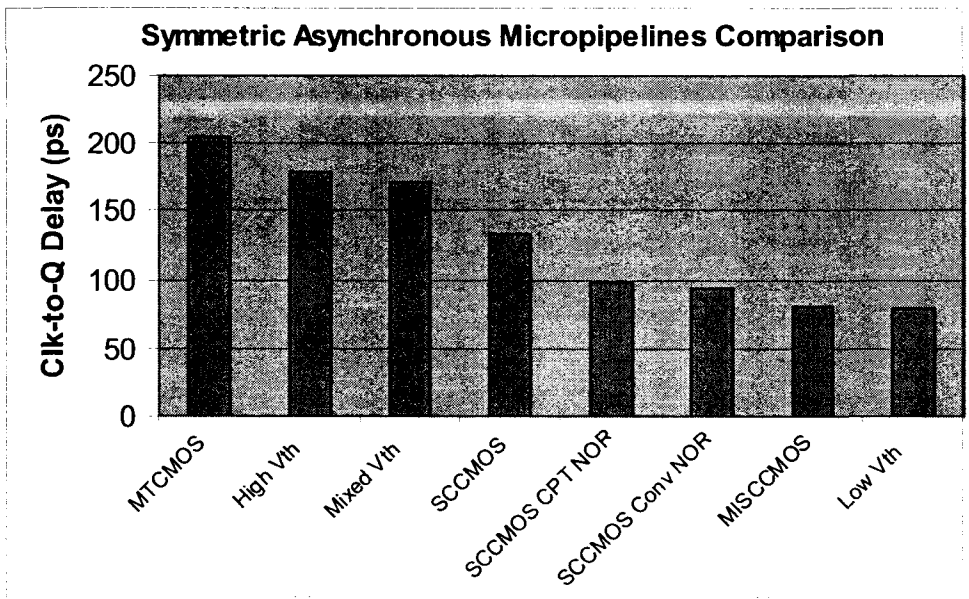

Chart 5.2: Symmetric C-Element Micropipelines Clk-to-Q Delay Simulation Results

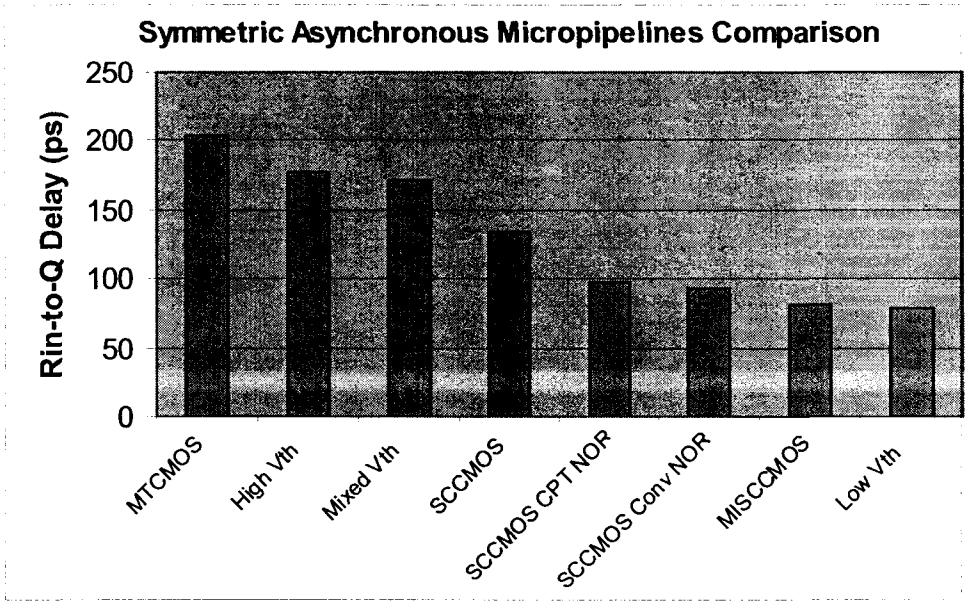

Chart 5.3: Symmetric C-Element Micropipelines Rin-to-Q Simulation Results

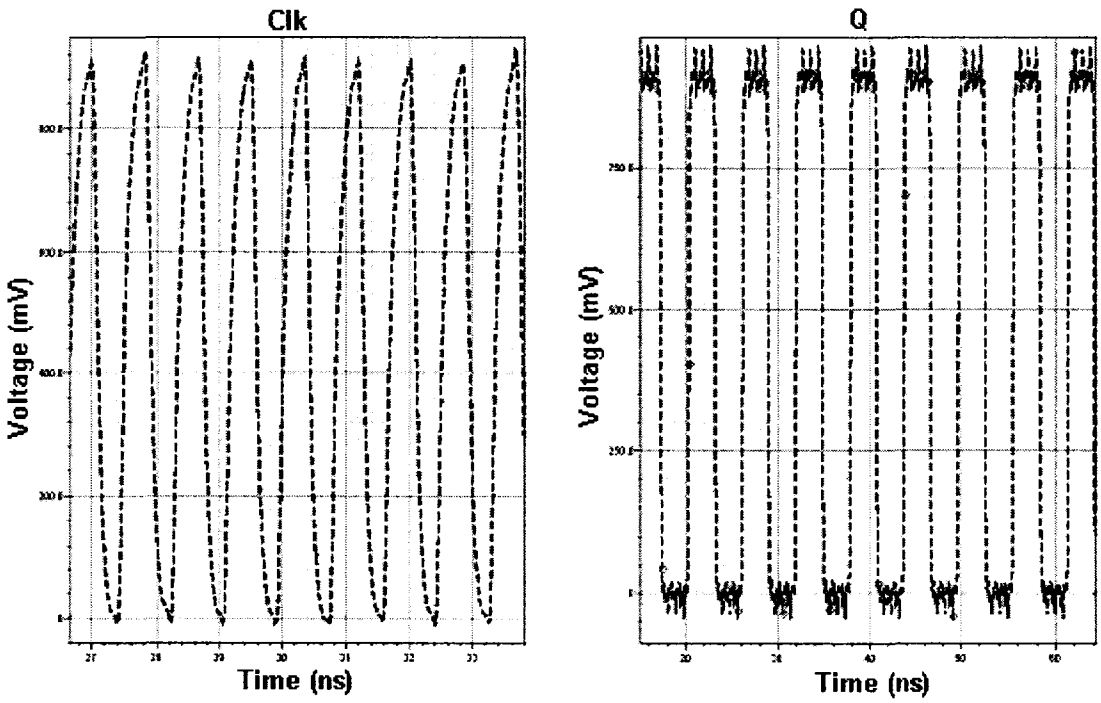

Figure 5.2: Symmetric C-Element Simulation Waveforms

The results illustrate the expected behavior of the high threshold voltage and low threshold voltage implementations. The high threshold voltage implementation is slower but uses less power, while the low threshold voltage one is much faster but uses much more current and therefore power. The MISCCMOS technique showed very fast results, which was expected since it uses low threshold transistors and does not add a series transistor which could cause a loss of performance, as seen with SCCMOS and MTCMOS. However, due to the multiplexers added, which can have significant leakage currents, the MISCCMOS micropipeline was not the best in terms of power dissipation. The MTCMOS and SCCMOS techniques also did not perform well in terms of power dissipation. This is mainly due to the fact that the simulation focused on the active mode of the micropipeline and not on the sleep mode, which was simulated earlier when simulating the C-Elements individually.

5.4 Conventional C-Element Micropipelines

The following Table 5.2 shows the results obtained from the conventional C-Element micropipeline simulations. The proposed C-Elements were also included here for comparison purposes. The simulations were run under the same test conditions as the symmetric C-Element micropipelines. Figure 5.3 shows sample simulation waveforms obtained from the third stage of the micropipeline.

Table 5.2: Conventional C-Element Asynchronous Micropipelines

Circuit	Average Dynamic Current	Clk-to-Q Delay / Stage	Rin-to-Q Delay
High V_{TH}	111.6 μA	123.6 *ps*	378.6 *ps*
Low V_{TH}	145.5 μA	77.74 *ps*	278.5 *ps*
Mixed V_{TH}	129.4 μA	70.41 *ps*	200.8 *ps*
MTCMOS	153.1 μA	100.1 *ps*	291.6 *ps*
SCCMOS	121.2 μA	88.27 *ps*	239.5 *ps*
Simplified C – Element 3	104.8 μA	80.72 *ps*	260.2 *ps*
Simplified C – Element 1	105.3 μA	120.5 *ps*	240.4 *ps*
Simplified C – Element 2	156.3 μA	76.49 *ps*	174.8 *ps*
Super Cutoff	95.6 μA	77.16 *ps*	203.4 *ps*

The following charts illustrate the results more clearly. The overall best performing techniques are the new proposed simplified C-Element micropipeline and the super-cutoff C-Element micropipeline. They exhibited good results both in terms of performance and power dissipation.

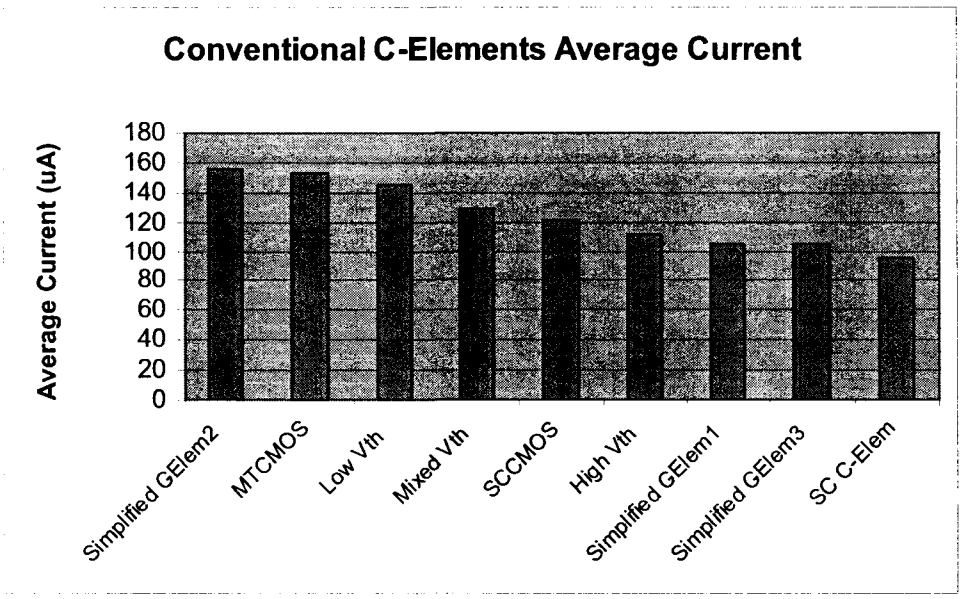

Chart 5.4: Conventional C-Element Micropipeline Average Current Simulation Results

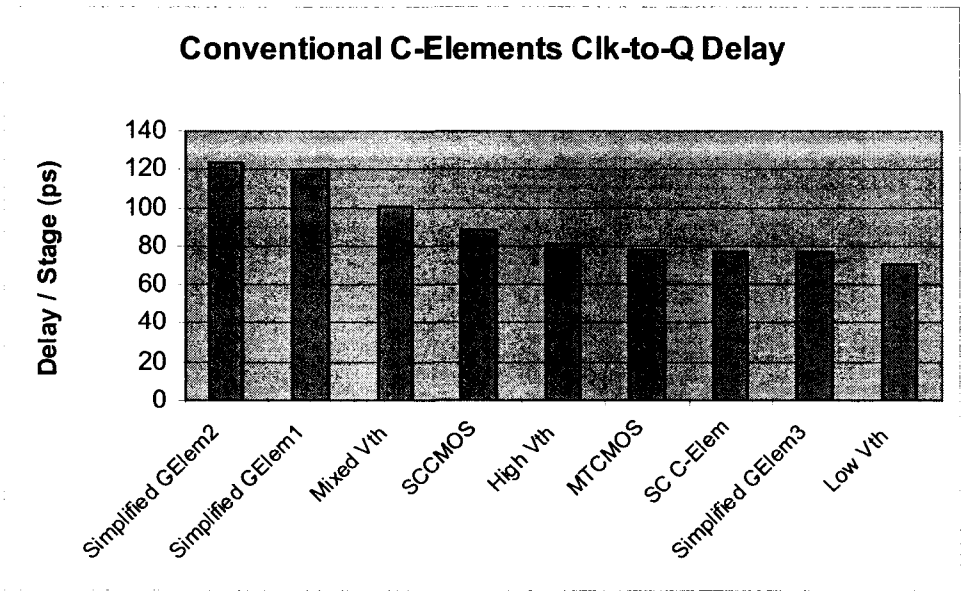

Chart 5.5: Conventional C-Element Micropipeline Clk-to-Q Simulation Results

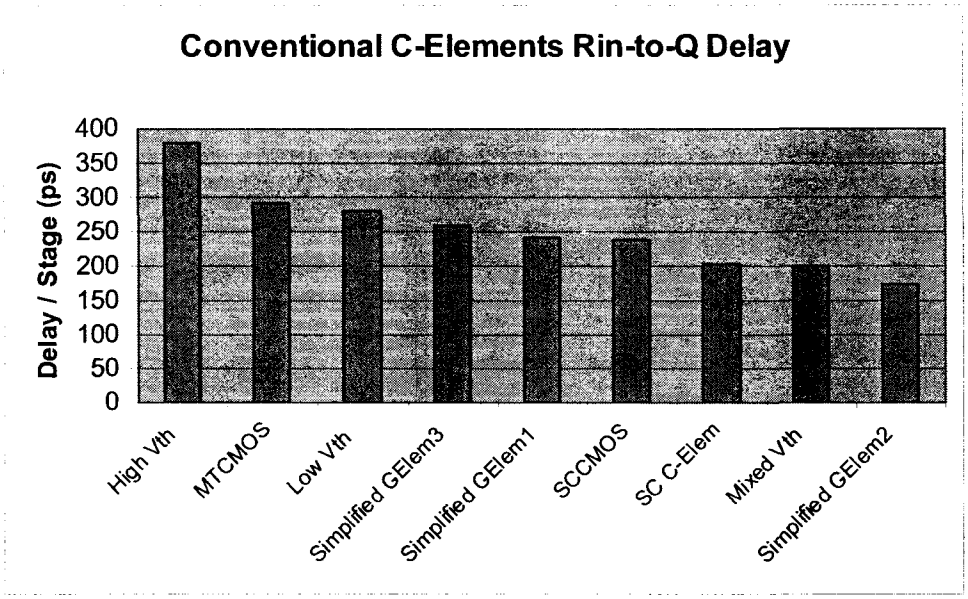

Chart 5.6: Conventional C-Element Micropipeline Rin-to-Q Simulation Results

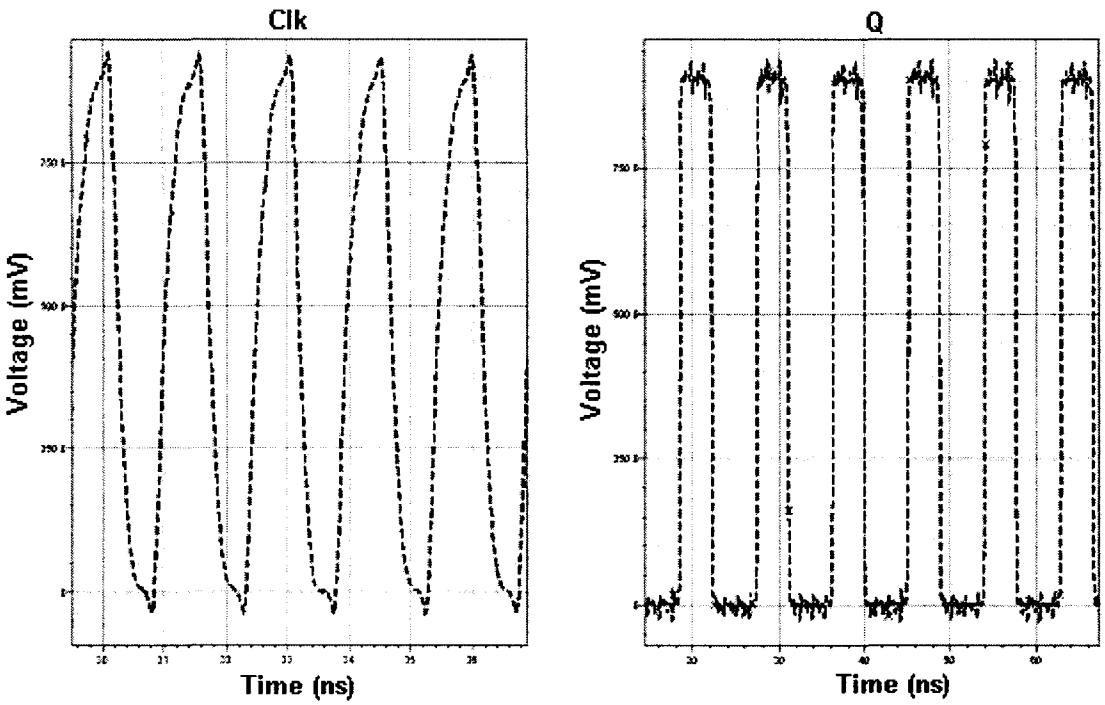

Figure 5.3: Conventional Micropipeline Simulation Waveforms

5.5 GasP Pipelines

The mixed threshold GasP circuit was selected to design a full GasP pipeline. This version of the GasP circuit was selected because of the suitability of using multi-threshold transistors in GasP circuits. After optimization of the circuit, the following results were obtained, as shown in Table 5.3. Figure 5.4 shows the test bench circuit used for simulation. Figure 5.5 shows a sample simulation waveform from one of the stages of the pipeline.

Table 5.3: Mixed Threshold Voltage GasP Pipeline Simulation Results

Frequency of Oscillation	*1.43 GHz*
Propagation Delay (State-In to Data Out)	*101.4 ps*
Energy Consumed per Cycle	*29.99 fJ*

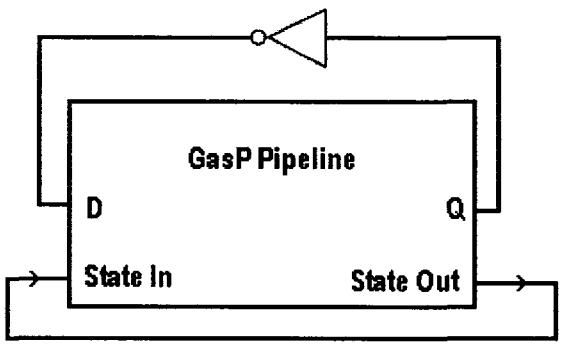

Figure 5.4: Mixed Threshold Voltage GasP Pipeline Test Bench Circuit

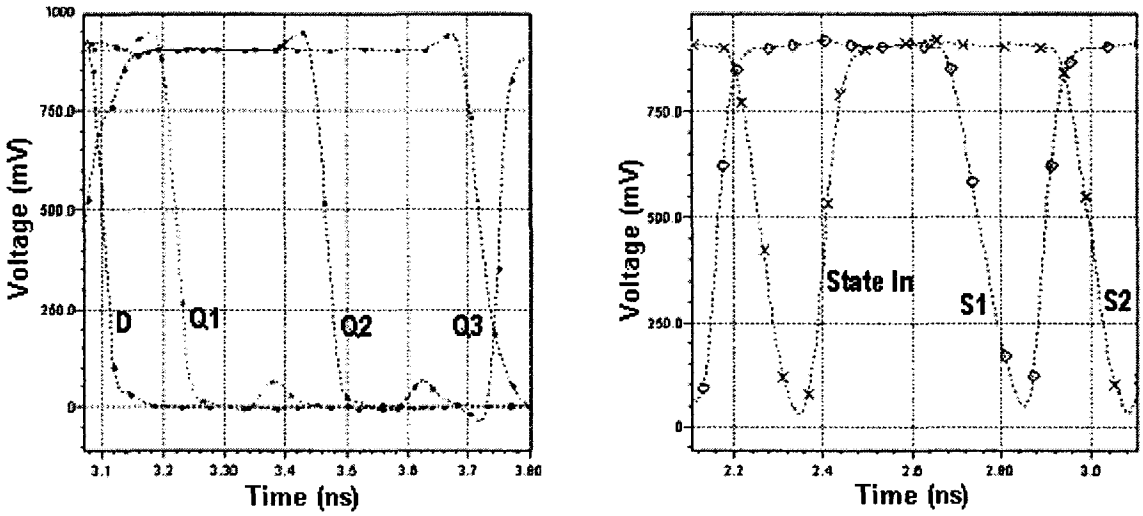

Figure 5.5: Mixed Threshold Voltage GasP Pipeline Simulation Waveforms

5.6 Discussion of Results

The simulation results presented give a clear view of which multi-threshold and low power techniques work best in conjunction with asynchronous micropipeline circuits. The conventional C-Elements in general performed better than the symmetric C-Elements for the purposes of the research leading to this thesis.

Of particular note are the Super Cutoff C-Element micropipeline and the Simplified C-Element micropipeline. These two micropipelines performed very well in all performance criteria. In terms of average current consumption, both these techniques exhibited the lowest levels of current during active operation of the circuit. They also performed very well in terms of delay compared to the other techniques. While the mixed threshold and SCCMOS implementations had better R_{in}-to-Q delay than the simplified C-Element implementation, they were much worse in terms of average current. The simplified C-Element micropipeline overall was more efficient.

5.6.1 Comparison between GasP and C-Element Based Asynchronous Pipelines

In order to compare the GasP and C-Element based asynchronous pipelines, we must use the same measurement criteria for the results to be valid. For propagation delay, the Rin-to-Q delay in C-Element based micropipelines is equivalent to the GasP delay measured from state-in to Q. For power dissipation, the best way to compare the two types of circuits is to use the energy-per-cycle measurement. This is because each type of pipeline runs at a different rate and therefore an average current or power measurement would not be accurate. Chart 5.7 below illustrates this comparison between the two asynchronous pipeline styles by using the energy-delay product for each pipeline circuit.

The chart clearly shows that GasP can achieve superior results compared to C-Element based micropipelines. This can be attributed to the less complex nature of the GasP control and data latching circuitry, which allows GasP to achieve better performance while using less energy per transition.

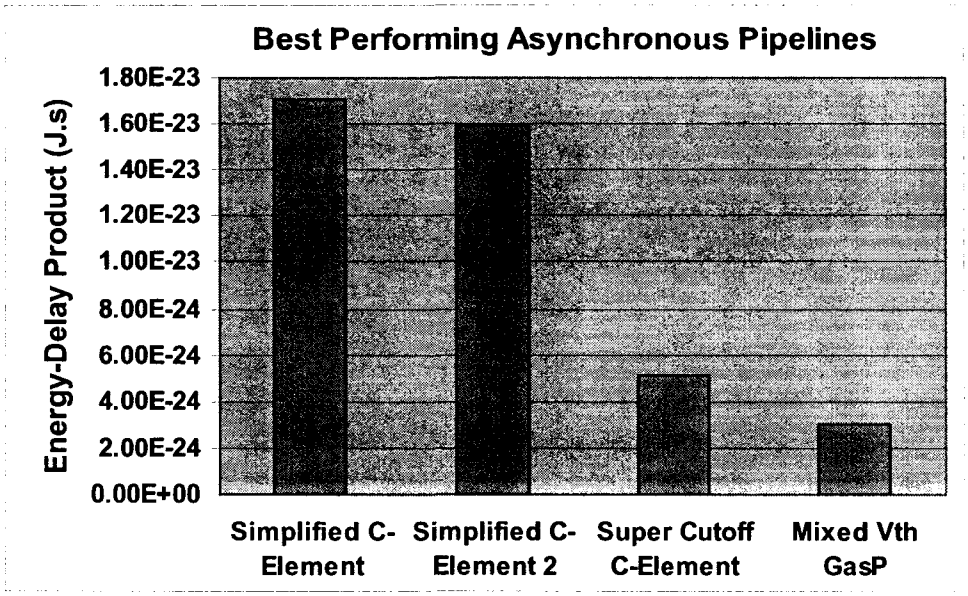

Chart 5.7: Asynchronous Micropipeline vs. GasP Pipelines

CHAPTER 6

Application Circuits

6.1 Introduction

This chapter presents two application circuits that were implemented based on techniques discussed in previous chapters. The application circuits are more complex pipelines than the circuits studied so far, containing combinational logic between stages. Layout and post-layout simulations were performed on these circuits to give a higher level of confidence that the results are achievable in a real design.

The post-layout simulation takes into account parasitic capacitances and more realistic delay models based on the layout of the circuit. This makes the results of the post-layout simulation more realistic than those obtained from schematic simulation. Post-layout simulation results usually closely mirror the behavior and performance of a circuit after fabrication, so performing this type of simulation on the application circuits gives great confidence that these circuits would actually work as expected if fabricated.

6.2 Multi-threshold Simplified C-Element Micropipeline Based 8-bit FIFO

The application circuit selected was an 8-bit five stage FIFO circuit. The circuit diagram of the application circuit is shown in Figure 6.1. In order to simulate the circuit, the FIFO was made to continuously oscillate. This was done by looping the R_{out} signal back to A_{in}, and the A_{out} signal back to R_{in}, as shown in the figure. The FIFO was chosen to be 8-bits wide to make it more realistic than a simple 1-bit FIFO.

The C-Element chosen to be used in this application circuit was the Simplified Multi-Threshold C-Element. This C-Element was selected due to its combination of performance and low power dissipation, and because it is a new circuit that has not been previously implemented or studied. Using this C-Element in an application circuit, and running post-layout simulations would prove the functionality and efficiency of this circuit.

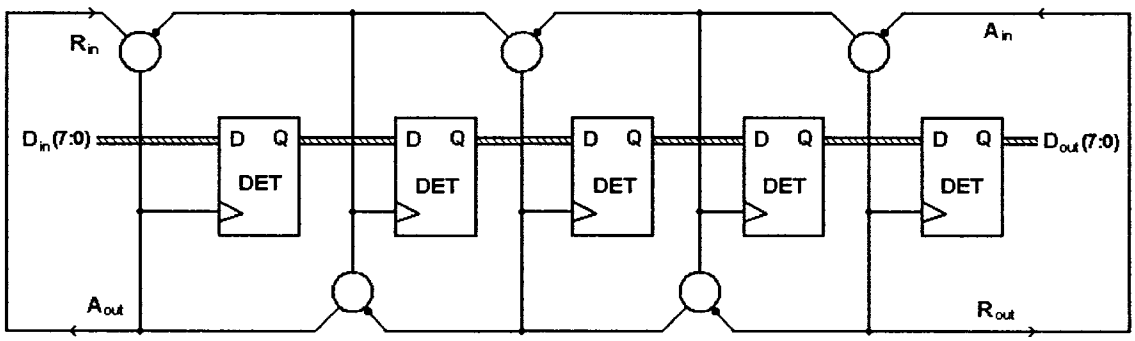

Figure 6.1: 8-Bit Micropipeline FIFO Schematic Diagram

The circuit layout was also performed, and the parasitic capacitances were extracted as well in order to be able to run post-layout simulations on the circuit. Figure 6.2 shows the layout of the FIFO.

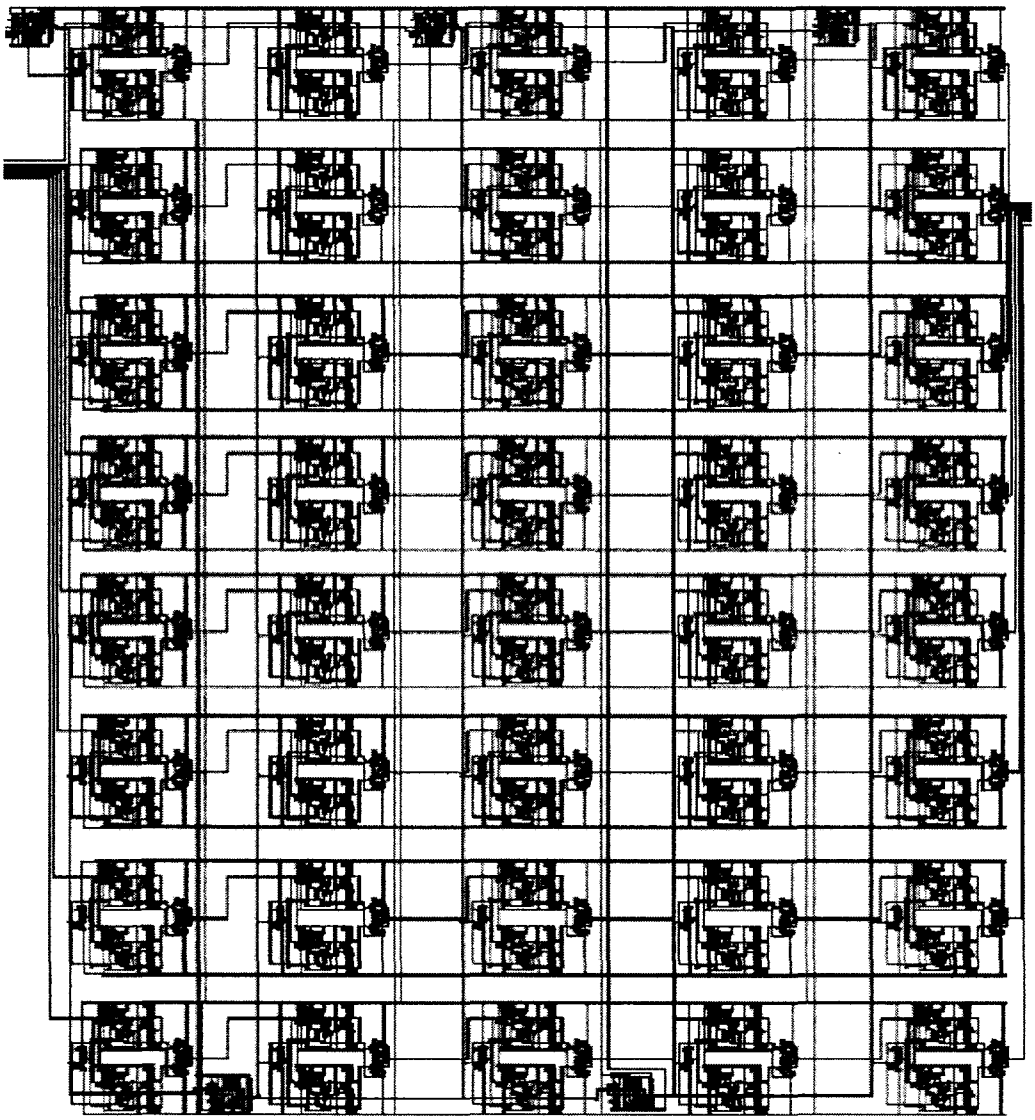

Figure 6.2: 8-Bit Micropipeline FIFO Circuit Layout

6.3 Multi-threshold Simplified C-Element Micropipeline Based 8-bit FIFO Simulation Results

After layout and extraction, post-layout simulations were carried out. A suitable test bench circuit was first developed. The test bench circuit was used to apply the data inputs. It also provided a pulse on R_{in} to start the FIFO. The signal R_{out} was looped back to A_{in}, and A_{out} was looped back to R_{in}. This allowed the FIFO to continue running at maximum speed. The following Table 6.1 presents the results obtained from the simulation. Figure 6.3 shows a sample simulation waveform of the outputs of the C-Elements in the FIFO.

Table 6.1: 8-bit FIFO Simulation Results

Forward Latency	*195.3 ps*
Backward Latency	*203.4 ps*
Throughput	*2.56 GB per second*
Current (at Max. Frequency)	*914 μA*
Current (Steady-State)	*114.6 nA*
Power Dissipation (at Max. Frequency)	*822.6 μW*
Power Dissipation (Steady-State)	*103.2 nW*

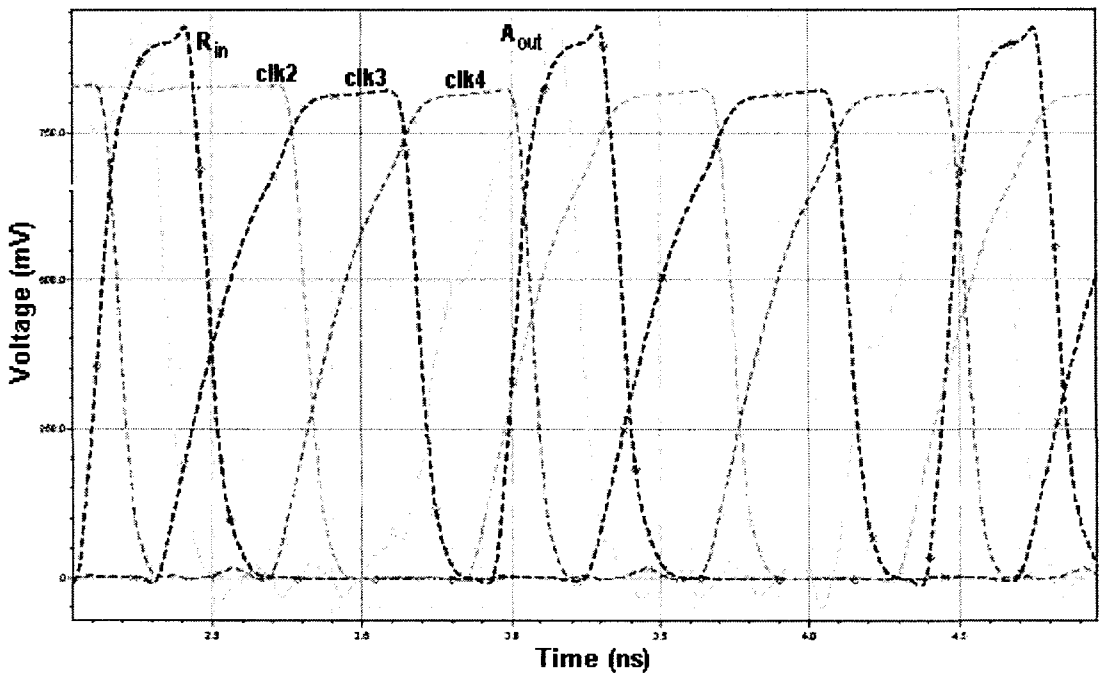

Figure 6.3: 8-Bit FIFO Simulation Waveform

The C-Element of each stage of the FIFO has two inputs, one from the preceding stage and one from the next stage. The forward latency was defined as the input to output 50% propagation delay, measured from the input originating in the previous stage. The backward latency is similar, except that it is measured from the input originating from the next stage. The reason for the difference is partly based on the circuit structure, but is also due to the fact that the backward path contains an inverter at the input, which adds some delay.

The throughput was measured as the maximum frequency of toggling at one of the C-Element output nodes. This is the rate at which the FIFO can move data. The current at the maximum frequency was averaged over a period of activity. The average current value changes depending on the frequency of switching in the FIFO, with the

average current at the maximum frequency given in the table. The other extreme is the average current when no switching is occurring, and this was called the steady-stage current in the table above. The power dissipation values were calculated by simply multiplying the average current with the power supply voltage of 0.9 V.

6.4 16-bit Brent Kung GasP Pipelined Adder

The second application circuit that was selected was a 16-bit Brent Kung GasP Pipelined Adder. The schematic of the adder is shown in Figure 6.4. The adder is a 16-bit Brent-Kung Adder, pipelined over four stages. GasP elements are used to control the pipeline and GasP data latches are used to hold data between the pipeline stages. The schematic is quite large and therefore not all of the internal elements are shown. The circuit layout is shown in Figure 6.5.

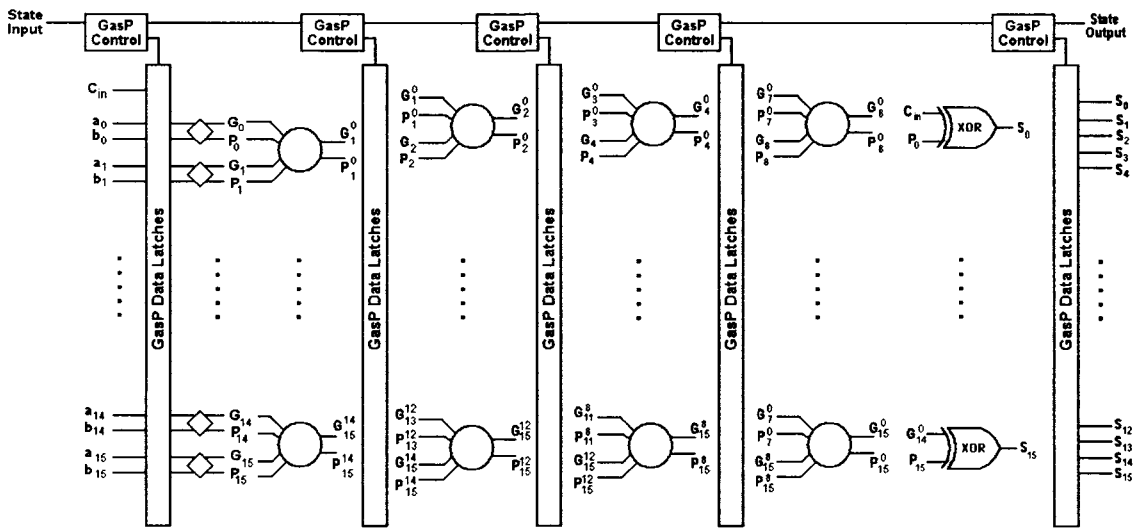

Figure 6.4: 16-bit GasP Brent Kung Pipelined Adder Schematic

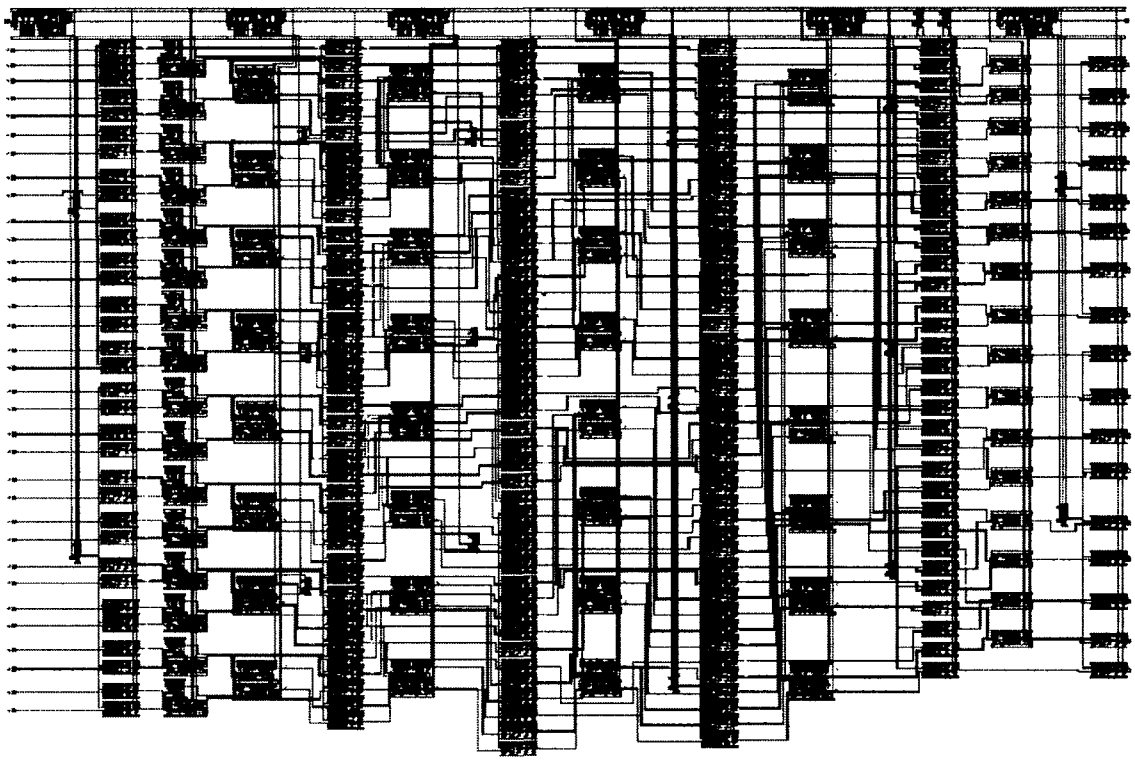

Figure 6.5: 16-bit GasP Brent Kung Pipelined Adder Layout

6.5 16-bit GasP Adder Simulation Results

Figure 6.6 shows the test bench circuit used to simulate the adder circuit. GasP elements were used to provide the appropriate interaction with the state input and output signals from the adder. The GasP element at the output had its output state held low in order to simulate it being full. This allowed the result of the addition to remain at the outputs of the adder to be verified. Several pairs of 16 bit numbers were tested to verify that the adder works properly. Figure 6.7 shows a sample extracted simulation waveform showing the state signals in the pipeline stages of the adder. Figure 6.8 shows both the schematic and extracted simulation results on the same plot for comparison purposes. The thick waveforms are the extracted ones while the thin lines represent the schematic

simulation waveforms. Figure 6.9 shows a sample simulation waveform of the output sum signals of the adder. In this case, it shows an output result of 0xF524 with a carry out of 1, which was a result of adding 0xF6A9 and 0xFE7B. The numbers were chosen intentionally to result in the carry out being high to verify the entire design. Finally, Table 6.2 shows the results obtained.

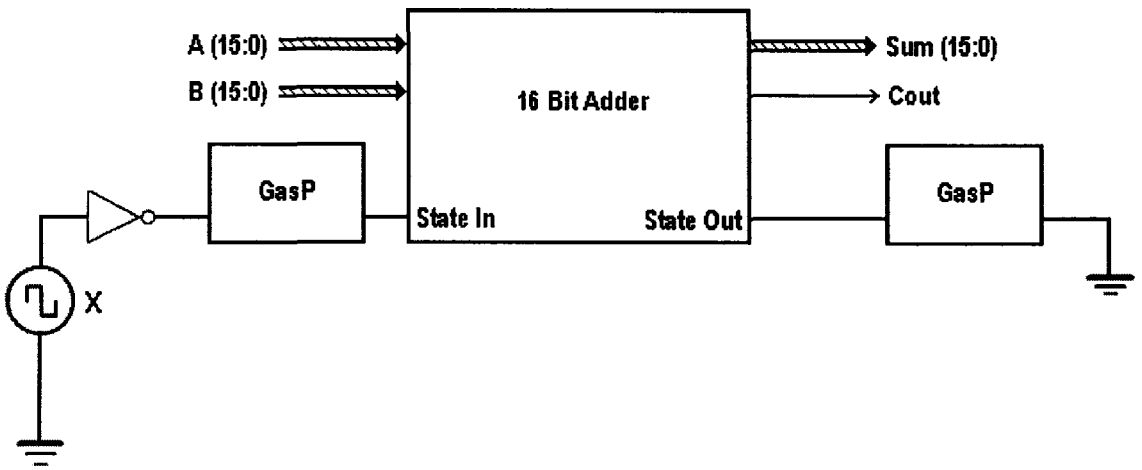

Figure 6.6: 16-bit Brent Kung GasP Pipelined Adder Test Bench Circuit

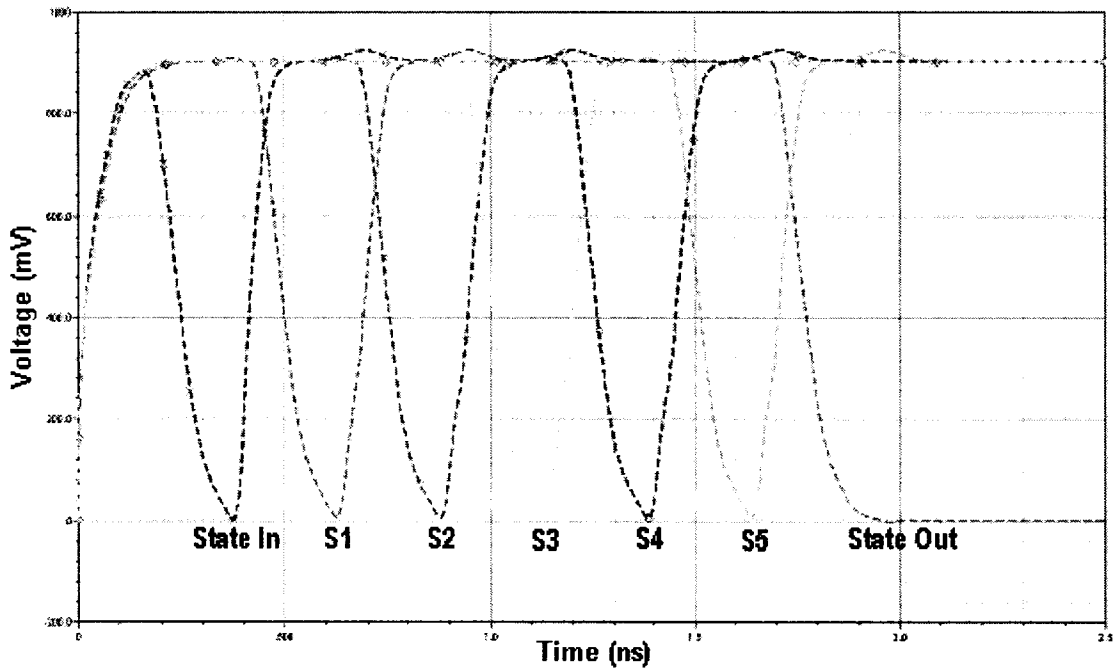

Figure 6.7: 16-bit Adder Extracted Simulation Waveforms

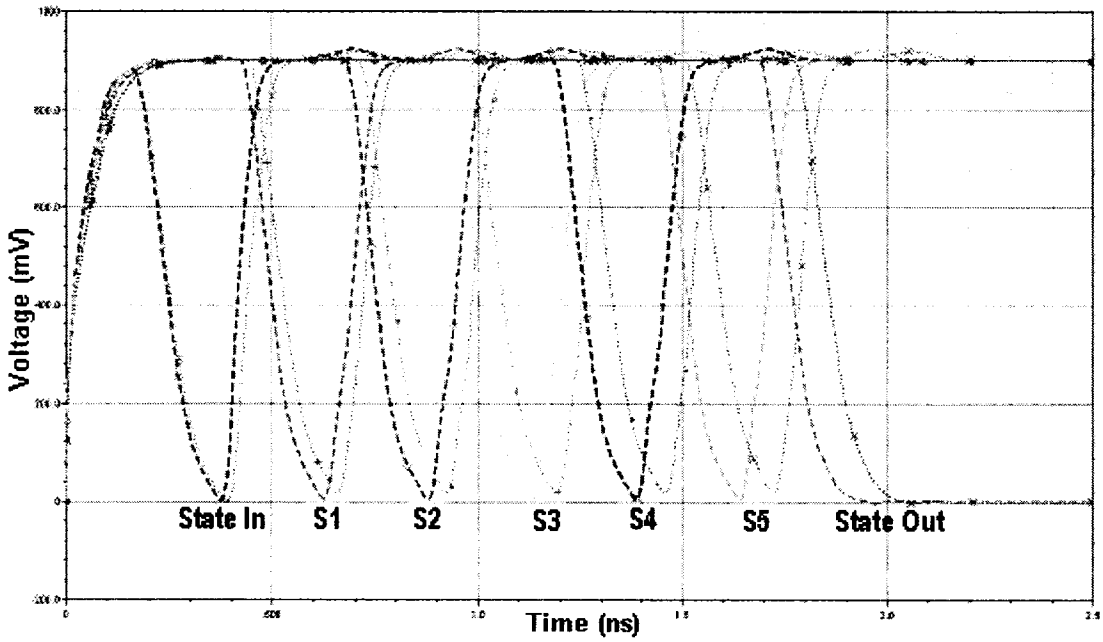

Figure 6.8: 16-bit Adder Schematic and Extracted Simulation Waveforms

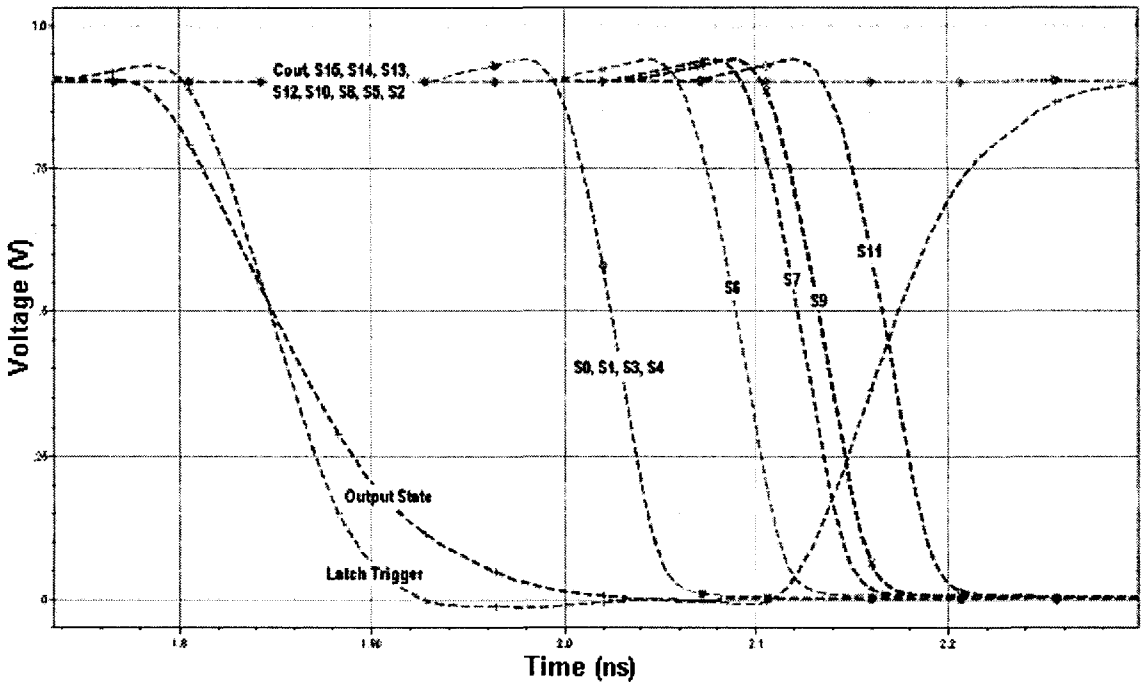

Figure 6.9: 16-bit Adder Extracted Simulation Waveforms

Table 6.2: 16-Bit Adder Simulation Results

Throughput	*3.96* Gig Items per second
Current (at Max. Frequency)	*1.69* mA
Current (Steady-State)	*503.6* nA
Power Dissipation (at Max. Frequency)	*1.52* mW
Power Dissipation (Steady-State)	*453.2* nW

CHAPTER 7

Conclusion and Future Work

7.1 Summary

This thesis explored circuit techniques using multi-threshold transistors that allow the design of high performance, low power circuits. The techniques available in existing literature on the subject were examined, studied and compared. MISSCCMOS, an improvement to the SCCMOS technique, was proposed and compared to other multi-threshold techniques. By using existing transistors in the circuit as sleep control transistors, the series sleep control transistor is eliminated, reducing the area overhead required. Simulation results show that the MISCCMOS technique provides slightly better performance than SCCMOS, while limiting sub-threshold leakage current just as well as SCCMOS.

Asynchronous pipelines were also studied, specifically the C-Element based micropipeline and GasP pipeline. For the first time as far as we know, multi-threshold techniques were applied to micropipeline and GasP pipeline circuits. A large number of multi-threshold micropipelines and GasP pipelines were designed and simulated, and the

results were compared to conventional single threshold asynchronous pipelines. It was found that the multi-threshold asynchronous pipelines outperformed the conventional pipelines while dissipating less power. In addition, the multi-threshold pipelines exhibited greatly reduced sub-threshold leakage current levels, which translated into greatly reduced static power dissipation compared to single threshold pipelines.

New and improved C-Elements and micropipelines were designed specifically to take advantage of multi-threshold techniques, such as the Simplified and Super-Cutoff C-Element based micropipelines. These C-Elements were found to outperform existing C-Element implementations, while dissipating less power and minimizing sub-threshold leakage currents.

New schemes of generating the sleep control signal for multi-threshold techniques such as MTCMOS and SCCMOS were explored. The Embedded Sleep Control Transistor C-Element was one such circuit that generated the sleep control signal internally. This is desirable because this circuit does not require an external sleep control signal which simplifies system level design. This also enables the C-Element not to lose state when it is placed in sleep mode. Simulation results show that this C-Element can compete well with other multi-threshold C-Elements.

Other schemes for saving the state of C-Elements when they are placed in sleep mode were explored. The DSSCCMOS C-Element was one such proposal based on SCCMOS. It involves placing a high threshold transistor that is permanently turned on in parallel to the low threshold sleep control transistor. When the circuit is placed in sleep mode, this transistor maintains a connection to the power supply voltage which enables

the state of the C-Element to be maintained. Since this is a high threshold voltage transistor, this does not happen at the expense of increased sub-threshold leakage current.

Multi-threshold technology was also applied to other micropipeline primitives such as delay element and dual edge-triggered flip-flops. Overall pipelines were designed using all the multi-threshold primitives. Simulation results again showed that the multi-threshold micropipeline implementations proposed in this thesis outperformed existing single threshold micropipeline implementations based on energy-delay product. At the same time, sub-threshold leakage current was minimized as much as high threshold micropipeline implementations.

Applications of multi-threshold techniques to GasP pipelines were also studied. A multi-threshold GasP pipeline was designed based on a mixed threshold voltage implementation. In this implementation, only critical path transistors were implemented using low threshold transistors while high threshold transistors were used for the remaining parts of the circuit. Simulation results show that this multi-threshold GasP pipeline achieves better performance and lower power dissipation than existing single threshold implementations. It also achieved better results than all the multi-threshold micropipelines designed in this thesis.

Two application circuits were designed and post-layout simulations were carried out. The first application circuit was a micropipeline based 8-bit five-stage FIFO. The proposed multi-threshold Simplified C-Element was used in the FIFO, and post-layout simulations were carried out in order to prove this new C-Element circuit. The second application circuit was a 16-bit Brent Kung GasP pipelined adder. The multi-threshold

GasP control element was used in this pipeline. The post-layout simulations proved that the proposed asynchronous pipelines can outperform existing techniques.

7.2 Conclusion

As the demands on integrated circuits continue to increase the need for new circuit techniques that allow designers to achieve high performance combined with low power will continue to grow. This work combines the advantages of multi-threshold techniques and asynchronous circuits to design a new class of high performance, low power circuits that can be used to satisfy these requirements. The circuits and techniques proposed in this thesis can also be used in applications that require low-power operation such as medical and portable devices. The combination of multi-threshold and asynchronous circuit techniques can provide very low static power dissipation which is a major requirement of these types of devices.

Most asynchronous pipeline primitives make use of "keepers" and latching structures. This is due to the nature of some asynchronous pipelines, which need to wait for events from the previous and next stages before any actions are taken. Multi-threshold technology was found to be very useful to implement these latching structures as it made them less resistive to change. This allowed multi-threshold asynchronous pipelines to achieve better performance and lower power dissipation as well. This thesis has demonstrated that multi-threshold technology is very well suited for high speed and low power asynchronous pipeline design.

7.3 Future Work

There is much more work to be done based upon the results obtained in this thesis. More complex application circuits should be designed and fabricated in order to prove our work in the lab. Fabrication of a complex application circuit that uses the techniques proposed in this thesis would provide more confidence that these techniques work in real circuits. Testing a fabricated circuit would also provide more accurate results and a deeper insight into how the proposed techniques would function in a real application.

The research should also be broadened to include other types of asynchronous pipelines in order to provide a more comprehensive coverage of the research area. More research should be done on single control line pipelines, like GasP, in order to find more efficient circuit structures and to find ways to take advantage of multi-threshold techniques fully. There are also other low-power techniques, such as multi-V_{DD} techniques, that should be studied and if possible combined with the work done in this thesis. One interesting idea that warrants further study is the use of multi-V_{DD} circuit design in conjunction with bulk biasing such as VTCMOS.

One of the shortcomings of this thesis is that a large number of techniques were examined and they were all applied to each type of asynchronous pipeline. This made it difficult to concentrate the research on one idea or technique. Therefore, it would be very beneficial to select some of the ideas in this thesis and focus on improving and developing these ideas further. This should be done as part of any future work that is to be done based on this thesis.

Bibliography

[1] Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic. *Digital Integrated Circuits, A Design Perspective.* New York: Prentice Hall Electronics and VLSI Series, 2003.

[2] Michael John Sebastian Smith. *Application-Specific Integrated Circuits.* New York: Addison-Wesley Professional, 1997.

[3] Neil H.E. Weste, Kamran Eshraghian. *Principals of CMOS VLSI Design, A Systems Perspective.* New York: Addison-Wesley Publishing Company, 1993.

[4] Kaushik Roy. "Leakage Power Reduction in Low-Voltage CMOS Designs." *IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, (Lisboa, Portugal), pp. 167-73, IEEE, 1998.

[5] Mohab Anis. "Subthreshold Leakage Current: Challenges and Solutions." *ICM 2003. Proceedings of the 15th International Conference on Microelectronics*, 2003.

[6] Liqiong Wei, Zhanping Chen, Kaushik Roy, Mark C. Johnson, Yibin Ye, Vivek K. De. "Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, Mar. 1999.

[7] Jonathan P. Halter and Farid N. Najm. "A Gate-Level Leakage Power Reduction Method for Ultra-Low-Power CMOS Circuits." *Proceedings, IEEE Custom Integrated Circuits Conference*, pp. 475-478, 1997.

[8] Shin'ichiro Mutoh, Takakuni Douseki, Yasuyuki Matsuya, Takahiro Aoki, Satoshi Shingematsu, Junzo Yamada. "1-V Power Supply High-Speed Digital Circuit

Technology with Multithreshold-Voltage CMOS." *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, Aug. 1995.

[9] James T. Kao and Anantha P. Chandrakasan. "Dual Threshold Voltage Techniques for Low-Power Digital Circuits." *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, Jul. 2000.

[10] Mohab Anis, Shawki Areibi, Mohamed Elmasry. "Design and Optimization of Multithreshold CMOS (MTCMOS) Circuits." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, Oct. 2003.

[11] Shin'ichiro Mutoh, Satoshi Shigematsu, Yasuyuki Matusya, Hideki Fuduka, Junzo Yamada. "A 1V Multi-Threshold Voltage CMOS DSP with an Efficient Power Management Technique for Mobile Phone Application." *IEEE Journal of Solid-State Circuits*, 1996.

[12] Mohab Anis, Shawki Areibi, Mohamed Mahmoud, Mohamed Elmasry. "Dynamic and Leakage Power Reduction in MTCMOS Circuits Using an Automated Efficient Gate Clustering Technique." *ACM/IEEE Design Automation Conference*, pp. 480-485, 2002.

[13] Mohab H. Anis, Mohamed W. Allam, Mohamed I. Elmasry. "Energy-Efficient Noise-Tolerant Dynamic Styles for Scaled-Down CMOS and MTCMOS Technologies." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 2, Apr. 2002.

[14] Mohab H. Anis and Mohamed I. Elmasry. "Power Reduction Via MTCMOS Implementation of MOS Current Mode Logic." *15th Annual IEEE International ASIC/SOC Conference*, 2002.

[15] Mohamed W. Allam, Mohab H. Anis, Mohamed I. Elmasry. "High-Speed Dynamic Logic Styles for Scaled-Down CMOS and MTCMOS Technologies." *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, 2000.

[16] Hironori Akamatsu, Toru Iwata, Hiro Yamamoto, Takashi Hirata, Hiroyuki Yamauchi, Hisakazu Kotani, Akira Matsuzawa. "A Low Power Data Holding Circuit with an Intermittent Power Supply Scheme for sub-1V MT-CMOS LSIs." *Symposium on VLSI Circuits Digest of Technical Papers*, 1996.

[17] H. Kawaguchi, K Nose, T. Sakurai. "A Super Cut-Off CMOS (SCCMOS) Scheme for 0.5V Supply Voltage with Picoampere Stand-By Current." *IEEE Journal of Solid-State Circuits*, vol. 35, no. 10, Oct. 2000.

[18] Mircea R. Stan. "CMOS Circuits with Subvolt Supply Voltages." *IEEE Design and Test of Computers*, vol. 19, no. 2, pp. 34-43, Mar. 2002.

[19] Tsuguo Kobayashi and Takayasu Sakurai. "Self-Adjusting Threshold-Voltage Scheme (SATS) For Low-Voltage High-Speed Operation." *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 271-274, May 1994.

[20] Tadahiro Kuroda, Tetsuya Fujita, Tetsu Nagamatu, Sinichi Yoshioka, Toshikazu Sei, Kenji Matsuo, Yoichiro Hamura, Toshiaki Mori, Masayuki Murota, Masakazu Kakumu and Takayasu Sakurai. "A High-Speed Low-Power 0.3um CMOS Gate Array With Variable Threshold Voltage (VT) Scheme." *IEEE Custom Integrated Circuits Conference*, pp.53-56, May 1996.

[21] Tadahiro Kuroda. "Low Power CMOS Digital Design for Multimedia Processors." *6th International Conference on VLSI and CAD*, 1999.

[22] Takayasu Sakurai. "Reducing Power Consumption of CMOS VLSI's through VDD and VTH Control." *Proceedings of the 1st International Symposium on Quality of Electronic Design*, 2000.

[23] Takayasu Sakurai, Hiroshi Kawaguchi and Tadahiro Kuroda. "Low-Power CMOS Design through VTH Control and Low-Swing Circuits." *International Symposium on Low-Power Electronics and Design*, pp.1-6, Sep. 1997.

[24] Tadahiro Kuroda, Tetsuya Fujita, Shinji Mita, Tetsu Nagamatu, Shinichi Yoshioka, Fumihiko Sano, Masayuki Norishima, Masayuki Murota, Makoto Kako, Masaki Kinugawa, Masakazu Kakumu and Takayasu Sakurai. "A 0.9V 150 MHz 10mW 4mm^2 2-D Discrete Cosine Transform Core Processor with Variable-Threshold-Voltage Scheme." *IEEE Journal of Solid-State Circuits*, pp. 1770-1779, Nov. 1996.

[25] Toru Tanzawa and Tomoharu Tanaka. "A Dynamic Analysis of the Dickson Charge Pump Circuit." *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, Aug. 1997.

[26] Jieh-Tsorng Wu and Kuen-Long Chang. "MOS Charge Pumps for Low-Voltage Operation." *IEEE Journal of Solid-State Circuits*, vol. 33, no. 4, Apr. 1998.

[27] Jongshin Shin, In-Young Chung, Young June Park and Hong Shick Min. "A New Charge Pump Without Degradation in Threshold Voltage Due to Body Effect." *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, Aug. 2000.

[28] John F. Dickson. "On-Chip High-Voltage Generation in MNOS Integrated Circuits Using an Improved Voltage Multiplier Technique." *IEEE Journal of Solid-State Circuits*, vol. sc-11, no. 3, Jun. 1976.

[29] Chistl Lauterach, Werner Weber, Dirk Romer. "Charge Sharing Concept and New Clocking Scheme for Power Efficiency and Electromagnetic Emission Improvement of Boosted Charge Pumps." *IEEE Journal of Solid-State Circuits*, vol. 35, no. 5, May 2000.

[30] Ivan E. Sutherland. "Micropipelines." *Communications of the ACM*, vol. 32, no. 6, pp. 720-738, Jun. 1989.

[31] N. R. Poole. "Self-timed logic circuits." *Electronics & Communication Engineering Journal*, Dec. 1994.

[32] S. B. Furber and J. Liu. "Dynamic Logic in Four-Phase Micropipelines." *Proceedings., Second International Symposium on Advanced Research in Asynchronous Circuits and Systems*, 1996.

[33] John Compton and Alexander Albicki. "Self-Timed Pipeline with Adder." *Proceedings of the Second Great Lakes Symposium on VLSI*, 1992.

[34] Gregg N. Hoyer, Gin Yee, and Carl Seechen. "Locally Clocked Pipelines and Dynamic Logic." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, Feb. 2001.

[35] Ivan E. Sutherland and Jon K. Lexau. "Designing Fast Asynchronous Circuits." *Proceedings of the International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 184-193, Mar. 2001.

[36] Maitham Shams, Jo C. Ebergen, Mohamed I. Elmasry. "A Comparison of CMOS Implementations of an Asynchronous Circuits Primitive: the C-Element." *ISLPED*, pp. 93-96, 1996.

[37] Maitham Shams, Jo C. Ebergen, and Mohamed I. Elmasry. "Modeling and Comparing CMOS Implementations of the C-Element." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 4, Dec. 1998.

[38] Maitham Shams, Jo C. Ebergen. "Optimizing Implementations of the C-Element." *Proceedings of the 1997 International Conference on Computer Design*, 1997.

[39] K. v. Berkel. "Beware the isochronic fork." *Integration, the VLSI journal*, 13(2):103–128, Jun. 1992.

[40] Ivan Sutherland and Scott Fairbanks. "GasP: A Minimal FIFO Control." *Seventh International Symposium on Asynchronous Circuits and Systems*, 2001.

[41] Hermander Singh Deogun, Rahul Rao, Dennis Sylvester and Kevin Nowka. "Adaptive MTCMOS for Dynamic Leakage and Frequency Control Using Variable Footer Strength." *Proceedings of the IEEE System-On-Chip Conference*, 2005.

[42] Chanseok Hwang, Changwoo Kang, Massoud Pedram. "Gate Sizing and Replication to Minimize the Effects of Virtual Ground Parasitic Resistances in MTCMOS Designs." *Proceedings of the 7th International Symposium on Quality Electronic Design*, 2006.

[43] Kyeon-Sik Min, Hun-Dae Choi, H.-Y. Choi, Hiroshi Kawaguchi and Takayasu Sakurai. "Leakage-Suppressed Clock-Gating Circuit With Zigzag Super Cut-Off CMOS (ZSCCMOS) for Leakage-Dominant Sub-70-nm and Sub-1-V-VDD LSIs." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 4, April 2006.

[44] K. Min. "Zigzag super cut-off CMOS (ZSCCMOS) block activation with self-adaptive voltage level detector." *Proceedings of the International Solid-State Circuits Conference*, 2003, pp. 400-401.

[45] Canh Q. Tran, Hiroshi Kawaguchi and Takayasu Sakurai. "More Than Two Orders of Magnitude Leakage Current Reduction in Look-Up Table for FPGA's." *IEEE International Symposium on Circuits and Systems*, 2005.

[46] Ahmad Shaheer and Michitaka Kameyama. "Real-Time Threshold-Voltage Control Scheme for Low-Power VLSI Under Fluctuation of a Supply Voltage." *International Symposium on Signals, Circuits and Systems*, vol. 1, pp. 15-18, 2005.

[47] Tin Wai Kwan, Maitham Shams. "Design of Multi-GHz Asynchronous Pipelines Circuits in MOS Current-Mode Logic." *Proceedings of the 18th International Conference on VLSI Design*, pp. 301-306, 2005.

[48] Y. Zafar and M. M. Ahmed. "A Novel FPGA Compliant Micropipeline." *IEEE Transactions on Circuits and Systems*, vol. 52, no. 9, pp. 611-615, 2005.

[49] Jo Ebergen, Jonathan Gainsley, Paul Cunningham. "Transistor Sizing: How to Control the Speed and Energy Consumption of a Circuit." *Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems*, pp. 51-56, 2004.

[50] K. Usmani and M. Horowitz. "Clustered voltage scaling techniques for low-power design." *International Symposium on Low Power Electronics and Design*, pp. 3 – 8, 1995.

[51] Chunhong Chen, A. Srivastava and M. Sarrafzadeh. "On gate level power optimization using dual-supply voltages." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 5, pp. 616 – 629, 2001.