Examensarbete

Near threshold operation of 16-bit adders in 65nm CMOS technology

Master Thesis Performed in Electronic Devices
Author: Ravi Maddula

Report number: LITH-ISY--14/4756--SE

Linköping April 2014
Near threshold operation of 16-bit adders in 65nm CMOS technology

Master Thesis Performed in Electronic Devices at Linköping Institute of Technology by Ravi Maddula

LiTH-ISY-EX--14/4756--SE

Supervisor: Dr. Behzad Mesgarzadeh
Examiner: Professor Atila Alvandpour

Linköping, April 2014
Abstract
The main objective of the thesis is to implement different architectures of 16-bit adders such as; Ripple Carry Adder (RCA), Manchester Carry Chain Adder (MCCA) and Kogge Stone Adder (KSA), in 65nm CMOS technology and to study their performance in terms of power, operating frequency and speed at near threshold operating regions. The performance of these adders are evaluated and compared with each other and a final conclusion is made as to which adder structure is more suitable for implementation in a 65nm technology for low power applications. Several optimisation techniques are performed for the adders to reduce the delay and power consumption. Propagation delay is the most critical or essential parameter to be considered, hence, to minimise the delay of the adder, a technique called sizing and ordering are required for the transistors. The purpose of the thesis is to make a fair comparison among adders over several metrics which include linearity, delay and power.

Simulation results of MCCA achieved a greater significant performance upon or over RCA and KSA, and proved it is the best suitable adder for low power applications.

Key words: RCA, MCCA, KSA, Linearity, Average Power, PDP, Operating Frequency, Optimisation
# Table of Contents

List of Figures  
List of Tables  
Abstract  
Acknowledgement  
List of Acronyms  
1 Introduction  
   1.1 Thesis organisation  
   1.2 Overview of full adder topologies  
      1.2.1 Static complementary CMOS full adder  
      1.2.2 Differential cascode voltage switch logic full adder  
      1.2.3 Complementary pass-gate logic (CPL) full adder  
      1.2.4 Transmission gate full adder  
   1.3 Static CMOS design styles  
      1.3.1 CMOS inverter  
      1.3.2 Modelling delay of an inverter  
      1.3.3 Mirror network  
   1.4 Optimisation techniques  
      1.4.1 Transistor sizing  
      1.4.2 Progressive sizing  
      1.4.3 Ordering Transistors in a CMOS network  
      1.4.4 Logical effort  
      1.4.5 Optimum number of repeaters in a transmission gate line  
      1.4.6 Optimum size of a repeater  
2 Design and implementation of a 16-bit architectures  
   2.1 Ripple-Carry-Adder  
      2.1.1 Critical path delay  
   2.2 Manchester Carry Chain Adder  
      2.2.1 Working principle  
      2.2.2 Circuit configuration  
      2.2.3 Carry chain circuit  
      2.2.4 Improvements in transmission gate line  
   2.3 Kogge-stone look ahead logarithmic adder  
      2.3.2 Illustration  
      2.3.3 Execution  
      2.3.4 Pre processing  
      2.3.5 Post processing  
      2.3.6 Carry generation process  
      2.3.7 Intermediate processing  
3 Simulation results  
   3.1 Test bench  
   3.2 Voltage scaling technique  
   3.3 Dynamic power consumption  
   3.4 Power delay product  
   3.5 Optimised results  
4 Comparison of results and analysis  
   4.1 Linear comparison of delay  
   4.2 Operating frequency
4.3 Average power comparison 33
4.4 Power delay product comparison 34
5 Conclusion and future work 37
6 References 39

List of Figures

Figure 1-1 Standard CMOS full adder 2
Figure 1-2 Schematic diagram of DCVSL full adder 3
Figure 1-3 Schematic diagram of CPL full adder 4
Figure 1-4 Circuit diagram of TG based full adder 5
Figure 1-5 Block representation of complementary based CMOS logic circuit 6
Figure 1-6 Inverter schematic circuits and its symbol 6
Figure 1-7 Mirror implementation of XOR circuit and its symbol 7
Figure 1-8 CMOS transistor stack 8
Figure 1-9 Ordering of transistors in a stack 9
Figure 1-10 Sizing of transistors through LE a) Inverter gate b) NAND gate C) NOR gate 10
Figure 1-11 Schematic representation of transmission gate line 10
Figure 2-1 Block representation of one bit adder cells connected in a cascoded form 13
Figure 2-2 1-bit MCCA 14
Figure 2-3 Schematic representation for 1-bit adder slice of MCCA 15
Figure 2-4 16-bit architecture of MCCA 16
Figure 2-5 Schematic of carry chain network 17
Figure 2-6 Modified version of carry chain network 18
Figure 2-7 Block diagram of Kogge stone adder 19
Figure 2-8 Functional representation of 4-bit KSA 20
Figure 2-9 Block diagram of propagate and generate logic 21
Figure 2-10 Schematic diagram for group propagate and generate function 22
Figure 2-11 Schematic representing group generate function 22
Figure 3-1 Common test bench setup 24
Figure 3-2 Dynamic behaviour of 16-bit adder 25
Figure 3-3 Simulation waveforms at near threshold operation 28
Figure 4-1 Propagation delay with respect to number of bits (N) 31
Figure 4-2 Operating frequencies of adders at different supply voltages 32
Figure 4-3 Average power with reduced supply voltages (Vdd) of 16-bit adders 33
Figure 4-4 Combined PDP Vs supply (Vdd) of 16-bit adders 34

List of Tables

Table 3-1 Simulation results for RCA 27
Table 3-2 Simulation results for MCCA 27
Table 3-3 Simulation results for KSA 28
Table 3-4 Optimised delay results 28
Abstract

The main objective of the thesis is to implement different architectures of 16-bit adders such as; Ripple Carry Adder (RCA), Manchester Carry Chain Adder (MCCA) and Kogge Stone Adder (KSA), in 65nm CMOS technology and to study their performance in terms of power, operating frequency and speed at near threshold operating regions. The performance of these adders are evaluated and compared with each other and a final conclusion is made as to which adder structure is more suitable for implementation in a 65nm technology for low power applications. Several optimisation techniques are performed for the adders to reduce the delay and power consumption. Propagation delay is the most critical parameter to be considered, hence to minimise the delay of the adder a technique called sizing and ordering is required for the transistors. The purpose of the thesis is to make a fair comparison among adders over several metrics, which include linearity, delay and power.

Simulation results of MCCA achieved a greater significant performance upon or over RCA and KSA, and proved it is the best suitable adder for low power applications.
Acknowledgement

Without great support and help from the people with whom I have an attachment, it would not be possible for me to write my Master’s thesis. I would like to thank the following people and organisations:

- My supervisor and advisor Assistant Professor Behzad Mesgarzadeh and my Examiner Professor Atila Alvandpour, for their guidance, patience, and support. Thanks for giving me the opportunity to do my Master’s Thesis.

- My Father Mr. M. V. Krishna Rao and my elder brothers Mr. Sampath Maddula, Mr. Shyam Sundar Maddula who always motivate me and have helped me financially during my difficult times.

- I would like to thank all of my friends, Srikanth Nakshatram, Suresh Babu Kollipara, Naga Kishan, Vamsee Krishna Bellamkonda, Koushal and Mohan Sure who always encouraged me by providing some tips and sharing personal life experiences during my studies in Linköping.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>Arithmetic Logic Unit</td>
</tr>
<tr>
<td>AOI</td>
<td>And Or Inverter Logic</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal-oxide Semiconductor</td>
</tr>
<tr>
<td>CPL</td>
<td>Complementary Pass Logic</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DCVSL</td>
<td>Differential Cascode Voltage Switch Logic</td>
</tr>
<tr>
<td>FP</td>
<td>Floating Point</td>
</tr>
<tr>
<td>KSA</td>
<td>Kogge Stone Adder</td>
</tr>
<tr>
<td>MCCA</td>
<td>Manchester Carry Chain Adder</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PDP</td>
<td>Power Delay Product</td>
</tr>
<tr>
<td>PDN</td>
<td>Pull Down Network</td>
</tr>
<tr>
<td>PUN</td>
<td>Pull Up Network</td>
</tr>
<tr>
<td>TG</td>
<td>Transmission Gate</td>
</tr>
<tr>
<td>LE</td>
<td>Logical Effort</td>
</tr>
<tr>
<td>VLSI</td>
<td>Very Large Scale Integrated circuits</td>
</tr>
</tbody>
</table>
Uphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Uphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se fölagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Ravi Maddula
1 Introduction

Addition is one of the critical and fundamental binary logic operations carried in digital circuits. Most electronic devices such as mobile phones, personal computers and tablet PCs, which are equipped with microprocessors contain Arithmetic and Logic Units (ALU). Being part of ALU, addition circuits are responsible for performing computation in determining the floating point calculation, memory address generation, index and similar operations in the Digital Signal Processor (DSP), where large volumes of visual and voice signals are processed at a very high speed. Hence, these require complex adder circuits.

Today most wireless communications are digitised and conditioned in the DSPs, which require a lot of computation power to have a complete control over analogue signals. Choosing the best adder is an utmost consideration in designing the processors, consuming less power and being efficient in high speed at the lowest possible supply voltage.

An adder cell in a digital circuit represents a one bit addition. There are two types of additions; the addition carried out by two bits is said to be half adder and the addition which adds three bits is named full adder. The third bit corresponds to a carry signal, which is fed to next stage of a full adder. In this work, the full adder is selected as a standard cell in implementing more complex adder structures. We mainly focused on the static style, since the performance over other logic styles is very attractive because of low dynamic power consumption and high noise margin [1] [2]. The Ripple carry adder has been chosen as a benchmark in this work. The other two selected adders are Manchester carry chain adder and a Kogge stone adder (parallel prefix-2) are of interest, which have 16-bit word length. The comparison is made with respect to supply voltage for a maximum of 1 volt and by scaling with a factor of 100 millivolts to determine the average power, power delay product and the operating frequencies. Simulations are carried for the worst case delay of carry signal, since critical path delay of each adder has been chosen as a main performance metric [3] [4].

1.1 Thesis organisation

This thesis is organised by the following Sections. Section 1; focuses on introduction and the theory behind different full adder topologies, static CMOS technology and about optimisation techniques. In Section 2, the design procedure of 16-bit architectures, where Ripple carry adder, Manchester carry chain adder and Kogge Stone adder and their functionalities are explained. Section 3 describes test-bench setup and the process of extracting the values using dynamic behaviour. Further, in the Section 4, the comparison and analysis has been made from the simulation results.

1.2 Overview of full adder topologies

In this Section, a background based on full adder circuits such as complementary static CMOS full adder, DCVSL full adder, CPL full adder and TG based full adder are thoroughly discussed.

Full adder is the critical block in the arithmetic and logic circuits. The usage of full adders is enormous in most of the digital circuits [3], particularly in the field of digital signal processing circuits [5]. These adders are better for forming multi-bit addition. There are several adders, each particular adder has been selected depending on the specific application. These full adders are described to show the variations in each logic style through the following circuit diagrams. These adders have some
advantages and disadvantages, the description of these full adders is explained in the following Sections.

1.2.1 Static complementary CMOS full adder
Static complimentary full adder is a standard logic based on CMOS technology. These full adders are constructed by pull-up and pull-down networks containing NMOS and PMOS transistors as shown in figure 1-1. These transistors are connected in such a way that they form a dual network. These adders operate on 28 transistors in total for performing binary addition. However, these adders are slow because of the large stacks that appear in both pull-up and pull-down network. The delay of these adders increase linearly with respect to increased bit length also because of the gate and diffusion capacitances associated with the node $Co$ [3].

![Figure 1-1 Static CMOS full adder [3]](image)

1.2.2 Differential cascode voltage switch logic full adder
The concept of DCVSL was first presented by Heller et al [6]. These gates are very complex and faster than conventional CMOS logic, later a comparison between DCVSL and conventional CMOS was made by Chu et al [6]. By using a straightforward technique, DCVSL circuits are built instantaneously based on karnaugh maps and tabular methods [7].

DCVSL full adder eliminates the use of a pull-up network, instead it uses cross coupled PMOS transistors. Therefore, less area is utilised [6] [7]. The majority of transistors in this full adder are NMOS transistors and their positive feedback helps to make the transitions as fast as possible [4]. The advantage of DCVSL over static CMOS logic is, that there is a very low static power consumption, which can be negligible. These adders use single gate architectures for producing true and complementary outputs. Therefore, the purpose of adding an inverter for generating complementary outputs can be eliminated [9].
Figure 1-2 Schematic diagram of DCVSL full adder [11]

DCVSL circuits are composed in two parts; one, a binary tree and the other, a load. There are two binary trees shown in figure 1-2, which are required to form a complete full adder. Figure 1.2 (a) generates the sum and figure 1.2 (b) generates the carry. These are designed in K-MAP procedure and treated as a tree structure. The cross coupled PMOS circuits shown in figure 1-2 (a) and figure 1.2 (b) acts as a load [9]

1.2.3 Complementary pass-gate logic (CPL) full adder
Complementary circuits are becoming more popular in implementing a special class of digital integrated circuits [3], especially for forming Exclusive OR (XOR) and Multiplexer (MUX) operations [10]. These adders have several advantages over static CMOS logic in power consumption and delay.

CPL full adder incorporates a small size NMOS transistor tree for logic function and a large size inverter for driving the output. The cross coupled PMOS transistor pair is made for compensating the threshold voltage drop of the NMOS trees [11]. The circuit diagram of a CPL full adder is shown in figure 1-3. When compared to static CMOS logic style, the input signals are given to source of the transistors [11]. These logic circuits produce strong outputs when compared to pass transistor logic, this is because of threshold voltage drop \( V_{out} = V_{DD} - V_{th} \) across NMOS transistors generating static currents at subsequent logic gates. This can be rectified by using level restoration at gate outputs. Due to cross coupled PMOS transistors, a fast differential stage is observed. The advantage of pass logic gates over static CMOS logic is a lower number of transistors, low input capacitance and low internal voltage swing. The disadvantage is, it is not suitable for low power applications due to higher switching rate at the internal nodes with respect to inputs and complementary inputs [10][11].
1.2.4 Transmission gate full adder

Transmission gates are essential for forming complex logic gates. These gates consist of complementary PMOS and NMOS pass transistors. The output of these gates becomes $V_{DD} - V_{Tn}$ where an input signal $(V_{DD})$ to the NMOS transistor is applied. Similarly, for a PMOS transistor when low voltage $(V_{SS})$ is applied, the output becomes $V_{SS} + V_{Tp}$. Where $V_{Tn}$ is the absolute threshold voltage of NMOS transistor and $V_{Tp}$ is the absolute threshold voltage of PMOS transistor, the output is reduced to $V_{Tn}$ or $V_{Tp}$ only when one transistor is considered to be a transmission gate [12].

One way of implementing full adder based on TG is by inverting the XOR gate, XNOR logic is generated. The other ways of implementation are explained in [1] [2] [11]. These based adders need inverters to provide complementary inputs, which allows rail to rail swing by arranging NMOS and PMOS transistors in parallel. The main disadvantage of TG based full adder is that it has less driving capability, this is because of the 'threshold voltage drop'. These adders do not have full voltage swing [14].
The performance of TG full adder degrades when they are connected in cascade form. Even if these adders have full voltage swing outputs and an intrinsic low power consumption, the delay of these adders increases quadratically according to Elmore delay [14]. Therefore, an extra effort should be made while designing these adders. This can be performed by properly selecting repeaters (buffers) along the signal path, which is critical [14].

1.3 Static CMOS design styles
In CMOS technologies, most of the digital circuits are implemented in static logic instead of dynamic logic, since dynamic circuits are more power hungry and uses much area in on-chip and off-chip applications. Hence, they are not suitable for handheld devices, which require long battery life.

In this Section, we mainly focused on simple logic gates to a more complex gates, which are used for implementing arithmetic logic function. These logic gates are common in most adder topologies that are used for constructing large bit addition.

Static CMOS circuits are designed by using pull-up network and pull-down network as shown in figure 1-5, where the pull-up network consists of PMOS transistors and pull-down networks use NMOS transistors [3]. The logic functions are designed in static CMOS circuits and these circuits may not be a dual network, but can also be symmetrical. An example of such a logic gate is “XOR” gate, since these gates have a pull-up and pull-down network, where NMOS and PMOS transistors are connected in such a way as they replicate with each other, while the other logic gates such as inverter, NAND and OR gates are good examples of a dual network.
The duality means, it is a straight approach in forming static CMOS circuit, which is implemented with NMOS and PMOS transistors. In order to have a correct function, duality is sufficient; but not necessary [15].

Figure 1-5 Block representation of complementary based CMOS logic circuit

Static circuit is a logical function [1] [2] in a digital circuit, which produces real outputs with respect to real inputs regardless of time [4]. In static networks the pull-up network establishes a path from $V_{DD}$ to $F$ (output), whereas the pull-down network establishes a path from $F$ (output) to ground as shown in figure 1-5.

1.3.1 CMOS inverter

Inverter is the critical and basic logical function in digital circuits, representing the logical value is either 1 or 0. A ‘1’ in a digital domain represents the highest potential ($V_{DD}$) and ‘0’ represents the lowest potential (Ground). A simple inverter with its symbol and circuit diagram are shown in figure 1-6. Inverters are not ideal in nature instead a close approximation is taken with respect to the ideal. They are also used to form a buffer to drive the input signals without any signal attenuation [2] [3]. The transistor’s M1 and M2 are PMOS and NMOS devices respectively. They are connected in such a way, which is shown in figure 1-6.

Figure 1-6 Inverter schematic circuit and its symbol
1.3.2 Modeling delay of an inverter

The delay of an inverter has been modelled in two regions for equally sized NMOS and PMOS devices. The delay corresponding to super threshold has been expressed in equation (1.1a) and the delay corresponding to the sub threshold region is expressed in equation (1.1b) [11].

\[ t_p = \frac{K C_i V_{DD}}{(V_{DD} - V_{th})^\alpha} \]  

(1.1a)

where \( K \) is the delay fitting parameter, \( \alpha \) is velocity-saturation parameter, and \( V_{th} \) is the threshold voltage.

\[ t_p = \frac{K C_i V_{DD}}{I_{o,g} \exp\left(\frac{V_{DD} - V_{th}}{n V_T}\right)} \]  

(1.1b)

\( V_T \) is the thermal voltage, \( n \) is the subthreshold slope factor, and \( I_{o,g} \) and \( V_{Th,g} \) are the fitting parameters.

1.3.3 Mirror network

Transistors in a mirror network are connected in such a way, that they are replicating with each other in a pull-up and pull-down network, but performing the same logic function as a complementary static CMOS logic, which has dual network. An example of such a logic gate is an Exclusive OR (XOR) function, which is a widely used logic function.

These circuits are fast because of symmetry. Therefore, charge and discharge time at output node is minimised by proper sizing using logical effort [16]. The Exclusive OR (XOR) gate is shown in figure 1-7.
1.4 Optimisation techniques

There are numerous methods employed by the designers to minimise the delay of the CMOS circuits. These particularly involve transistor sizing, transistor ordering and by using logical effort (LE) [16] [17].

1.4.1 Transistor sizing

Sizing in CMOS circuits are performed to have an equal charge sharing across the nodes, associated between pull-up and pull-down transistor networks. Hence, equal rising and falling times across the nodes with respect to the input signals are observed. Such techniques are followed by sizing and progressive sizing of a transistor having large stacks [17].

During the optimisation of a simple CMOS inverter, if we assume no wire capacitance the ratio for minimum delay with respect to the widths of the transistors is given in the equation (1.2) [17].

\[
\frac{W_p}{W_n} = \sqrt{\frac{\mu_n}{\mu_p}}
\]

(1.2)

1.4.2 Progressive sizing

A stack in a CMOS circuit represents transistors, lumped together as a single load capacitance, having no internal capacitance between pull-up and pull-down networks. This can be over-simplified by a simple model as shown in figure 1-8 [17].

In order to extract the delay of the circuit shown in figure 1-8, the network of capacitors and resistors between each node have to be solved. In this circuit, transistor T_N has to discharge the load of the capacitance C_1 while T_1 has to carry the discharge current from the total capacitance C_T = C_1+C_2+C_3+...C_N, which has been considerably larger. Hence progressive scaling is beneficial [18].

![Figure 1-8 CMOS transistor stack [17]](image)
1.4.3 Ordering transistors in a CMOS network

Transistor ordering is a well-known technique used in reducing the delay of the circuit [17]. The path through the combinational network, which determines the ultimate speed of the structure, is called critical path [3]. Therefore, placing the transistors, which drive critical input signals closer to the F (output) can result in higher speed. Below figure 1-9 explains the ordering of transistors in detail [17].

Assume $C_{in}$ to be a critical signal, when $C_{in}$ undergoes 0-1 transition and the inputs A and B are kept at high where $C_L$ is initially charged, so no path to gnd exists until $T_1$ is turned on, as it is the last event to happen. The delay associated during the interval between $C_{in}$ to F (output) is extracted by considering the time taken to discharge $C_{in} + C_1 + C_2$. This can be observed in the figure 1-9 (a). By arranging the $C_{in}$ close to the output F, only $C_L$ (which is shown in figure 1-9 (b) has to be discharged, whereas the rest of the capacitance $C_1$ and $C_2$ are in a state of discharge, which results in a faster response time [18].

1.4.4 Logical effort

The characteristics of logic gates are first determined by logical effort and by parasitic effects. The process that determines the logic gates are by using fewer process parameters, using circuit simulations and by using fabricated test structures [16].

**Logical effort for a group of signals:** Several logic gates are combined together to form more complex gates, having a large number of input signals. The logical effort of such gates are determined by grouping all the input signals followed by the equation (1.3).

$$g_b = \frac{C_b}{C_{inv}} = \frac{\sum C_i}{C_{inv}} \quad (1.3)$$

Where $g_b$ is the logical effort of group b, $C_b$ is the combined input capacitance of each signal in group b, and $C_{inv}$ is the input capacitance of the inverter [16].
Calculated logical effort of the basic gates: To have a same current drive between the gates, logical effort plays a major role. Figure 1-10 shows the calculated logical efforts for CMOS inverter, two input NAND and NOR-gates. In general, PMOS devices have lower mobility charge carriers than NMOS devices. In order to have an equal conductance between the devices the width of the PMOS is adjusted, by doing so, the obtained values shows that \( W_p = 2.645 \) times the value of \( W_N \) [16].

![Image](image1)

**Figure 1-10** Sizing of transistors through LE a) Inverter gate b) NAND gate c) NOR gate

### 1.4.5 Optimum number of repeaters in a transmission gate line

The delay of an RC line increases quadratically when the RC line gets longer [3], in addition to this there is significant loss in power when the signal is travelling along the line. Therefore, repeaters are employed to improve its performance. Splitting the RC line into branches and by placing the repeaters in between can help to restore the signal; and a small amount of delay is added. The problem with the minimum number of stages \( N \) is determined by first assuming for equal rise and fall times (\( W_p = W_n \)) through the following equation (1.4); and we can observe in the figure 1-11 [18] [14].

\[
\frac{\partial N}{\partial N} = 0 \Rightarrow N = \sqrt{\frac{R_C}{C_L}} = \sqrt{\frac{T}{T}}
\]

(1.4)

![Image](image2)

**Figure 1-11** Schematic representation of transmission gate line

Where \( R_L \), \( C_L \) are resistance and capacitance per unit length of a transmission gate line, \( W_{min} \) is the
minimum width of the transistors, $R_t$, $C_t$ are the minimum size parameters of an inverter, and the
$\tau_L$, $\tau$ are the time constants and is given by $\tau_L = R_t C_L$ and $\tau = R_t C_t$.

1.4.6 Optimum size of a repeater
Repeater is a chain of inverters connected to drive large load capacitance such as long buses, I/O
buffers and off-chip capacitive loads. When designing the buffer, each inverter is made larger than the
previous inverter in order to have maximum performance. The chosen number of stages $M$ and the
ratio $g$ between the two inverter stages is shown in the equations (1.5a) and (1.5b) [18].

\[
M = \ln \left( \frac{C_{load}}{C_{min}} \right) \quad (1.5a)
\]
\[
g = e \quad (1.5b)
\]

If delay is the main concern, reducing the ratio $g$, will decrease the delay and lead to a greater number
of stages per repeater [18].
2 Design and implementation of 16-bit architectures

2.1 Ripple Carry Adder (RCA)

The architecture of a RCA is simple and takes less time to design. These adders are constructed by connecting each full adder in series with each other in a cascaded form, so that the carry generated in any of the full adders should be propagated to the next stage as shown in the figure 2-1. This effect is called rippling. Therefore, it is named Ripple carry adder. For an n bit adder, it requires n full adders. The main drawback of a Ripple carry adder is, it is not very efficient for large number of bits, particularly for the bits ranging from 64 to 256. Thus the delay increases with the increased bit length [3].

![Figure 2-1 Block representation of 1-bit adder cells connected in a cascode form](image)

2.1.1 Critical path delay

The performance of any adder can be judged by the longest path, where input signal travels to the output with a certain amount of delay. In RCA, time taken by the signal carry-in (C\text{in}) to reach carry-out (C\text{out}) is the longer than the time taken by A to carry-out or from carry-in to sum, which is shown in figure 2-1. Therefore, it is considered as the critical path. The carry propagation will determine the latency of the whole circuit for a Ripple carry adder.

The delay through the circuit is mainly due to the number of logic stages that must be passed over and it is the function of applied input signals. In RCA the worst case delay occurs when a carry generated at the last significant bit position ripples all the way to the most significant bit position, which is approximated in the equation (2.1) [3].

\[
T_{\text{rca}} = (N - 1)T_{\text{carry}} + T_{\text{sum}}
\]

(2.1)

where N represents the number of bits, T\text{carry} and T\text{sum} are the time taken for the signal C\text{in} to travel to C\text{out} and from C\text{in} to sum (S4) of the Most Significant Bit (MSB).

The standard full adder shown in figure 1-1 has been used in our design for constructing RCA. This adder is implemented by reusing the carry term [3]. The Boolean representation for sum and carry are given in the equation (2.2a) and (2.2b) [3].

\[
\text{sum} = A \oplus B \oplus C\text{in}
\]

(2.2a)

\[
\text{carry} = (A \cdot B) + C\text{in} \cdot (A \oplus B)
\]

(2.2b)
This cell utilises 28 transistors for designing the full adder in a complementary MOS circuit; however, because of large stack of the transistors connected in both pull-up and pull-down networks, this circuit is slow. By considering optimisation techniques, the circuit is thoroughly optimised for minimum delay. In this case we mainly concentrate on the input signal of the transistors that connect the nodes of the critical path. Since the critical path is much more important than any other factors, and it mainly effect the adder performance [3].

2.2 Manchester carry chain adder (MCCA)

The design procedure of 16-bit MCCA adder based on transmission gate has been constructed by using static logic style. These adders depend on the following logic functions Propagate (P), Generate (G) and an extra Delete (D). The signal carry \( C_{in} \) propagates through a transmission gate when the propagation function is true. A Delete signal is added in order to discharge the output node, carry-out \( C_{out} \) as shown in figure 2-2. The Generate function produces a carry at node, when the conditions for both propagation and delete is not true [3].

We have selected MCCA to model a fair comparison with reference to RCA. The purpose of choosing MCCA for a comparison is, since it is distinct from the RCA in terms of 1 bit full adder topology, and the carry propagation circuit, which is not similar. Instead, the carry signal travels through a chain of transmission gates, resembling the transmission line path. However, considering all the parameters the MCCA is well optimised in terms of logical equation and by using fast logic gates such as XOR, NAND and Exclusive NOR (XNOR) for implementing a complete adder slice [3].

2.2.1 Working principle of MCCA

The circuit in figure 2-2 shows a 1 bit implementation of Manchester carry chain adder. The signals coming from the Propagate, Generate and Kill or Delete functional blocks acts in accordance with the input signals A, B and are not dependent on \( C_{in} \), whereas the sum and carry functional blocks completely rely on \( C_{out} \) of a previous stage [3].

![Figure 2-2 One bit MCCA](image)

MCCA use propagate and generate logic functions to produce sums. These logic functions utilise static implementation of XOR and NAND gates as shown in figure 2-3. This adder slice has less stack height upto a maximum of two transistors and uses an And Or Invert (AOI) logic function unlike a traditional standard static CMOS adder shown in figure 1-1. The Boolean equations for Propagate,
Generate and Kill signals are explained in the following equations (2.3a) and (2.3b) [3].

\[ P_i = A_i \oplus B_i \]  \hspace{2cm} (2.3a)

\[ G_i = A_i \cdot B_i \]  \hspace{2cm} (2.3b)

Figure 2-3 Schematic representation for 1-bit of MCCA

2.2.2 Circuit configuration
A simplified architecture for a 16-bit word length of an MCCA is shown in figure 2-4. These blocks are well defined with a functional and schematic explanation in the following Sections 2.2.3 (Carry chain circuit), 2.2.4 (Carry chain delay) and 2.2.5 (Improvements in transmission gate line).
2.2.3 Carry chain circuit
The whole performance of a Manchester carry chain adder completely relies on a carry chain circuit as shown in figure 2-5. In static implementation, the carry chain of an adder is constructed by TG, which contains complementary NMOS and PMOS transistors. Where as the dynamic logic makes use of NMOS only pass transistors in a carry chain. Therefore, in this work the dynamic circuit is eliminated for comparison.

Integrating transmission gates takes up much area, but the power consumption is less due to static design. These transmission gates are connected in series resembling the transmission line having equivalent on-resistance and capacitance associated across each NMOS and PMOS transistors. The generated carry at any bit position (in case of 16-bit adder) has to propagate through each TG to the Most Significant Bit (MSB) position without any signal attenuation. Hence, careful optimisation is needed in order to reduce the delay and signal attenuation.

2.2.4 Carry chain delay
The performance of the MCCA is determined by the carry chain path. The longest delay in MCCA is the time taken by the C_{in} to reach C_{out} than the C_{in} to sum. Hence, C_{in} to C_{out} is the critical path in the MCCA. The equation (2.4) gives the information about critical path delay [3].

\[
t_p = 0.69 \left( \sum_{i=1}^{N} C_i \right) \left( \sum_{i=1}^{l} R_i \right)
\] (2.4)
2.2.5 Improvements in transmission gate line
During MCCA design, “The carry computation and propagation has become a major performance bottleneck” [19]. Various design issues related to linearity and full voltage swing across the TG in a carry chain have been explored and evaluated.

Buffers: In CMOS circuits, the concept of introducing a buffer has been considered to drive the input signal without any attenuation, hence buffers treated as repeater in a transmission line [14]. These repeaters are widely used in wireless communication systems. Since at some distance, the power of a radio signal gets attenuated. This is mainly due to absorption and reflections of the medium.

Resolved low voltage swing across the nodes of the TG chain: The problem with full voltage swing across the nodes of a carry chain has been solved by placing the buffer at a regular interval of two transmission gates associated at the node and by progressive sizing of the transistors along the transmission gate line. Since delay increases quadratically along a chain, therefore, the signal along the path gets attenuated.
2.3 Kogge Stone look ahead logarithmic adder (KSA)

The concept of KSA was first developed by Peter M. Kogge and Harold S. Stone. Hence, it is named as Kogge Stone adder. During the year 1973, the paper entitled “A parallel algorithm for the efficient solution of a general class of Recurrence Equation” has been published. It is a complex adder based on “parallel prefix form carry lookahead adder” [20]. So being complex, it has lower fan-out, which requires much area and more interconnecting circuits than previously described adder architectures. The delay of KSA is directly proportional to the number of levels in the carry propagation network and the carry bits are generated at $O(\log n)$ and takes less time. These carries are computed in parallel at a cost of increased area [20]. The functioning of the KSA is well explained through different functional blocks; these blocks are explained in detail by the following subsections. This KSA implementation uses radix-2 which refers to two results which are generated from the previous stages, which is shown in figure 2-7. The theory based on prefix circuits provides a solid understanding for a wide range of design trade-offs between delay, area and wire complexity [21].
2.3.1 Illustration

The vertical stages in the architecture produces propagate and generate bits as shown in figure 2-8. The carry bits are produced at the last stage (vertically) and are passed through the sum block with initial propagate bits, which are passed through XOR gate to produce sum [21] [22].
2.3.2 Execution
Several adder structures are considered as parallel prefix adder architectures consisting of three main functional blocks. These blocks are built by basic logic gates. The selected architecture of KSA is a class of radix-2 tree adder that combines the generate and propagate signals. The carry-out is computed in \( \log_2(N) \) time. It has a frequent repeating structure and requires a lot of interconnections. The implementation of KSA is straightforward followed by several steps from the equation (2.5a) to (2.5g) [3] [22].

- \( S_i = P_i \text{ XOR } C_{i-1} \) \hspace{1cm} (2.5a)
- \( C_i = G_{(i:0)} \) \hspace{1cm} (2.5b)
- \( \text{Generate : } G_i = A_i \text{ AND } B_i \) \hspace{1cm} (2.5c)
- \( \text{Propagate: } P_i = A_i \text{ XOR } B_i \) \hspace{1cm} (2.5d)
- \( \text{Dot Product : } (G_1, P_1) \ast (G_0, P_0) = (G_1 + P_1 \ast G_0, P_1 \ast P_0) \) \hspace{1cm} (2.5e)
- \( \text{Empty dot product : } (G_1, P_1, G_0) = G_1 + P_1 \ast G_0 \) \hspace{1cm} (2.5f)
- \( G_{(i:j)} = G_{i:k} + P_{i:k} \ast G_{((i-1):j)} \) \hspace{1cm} (2.5g)

2.3.3 Preprocessing
The preprocessing stage can be perceived as the half adder, or an AND gate and an XOR gate as shown in figure 2-9. Here the combination of NAND and inverter represents the AND logic. This block produces propagate and generate signal through a pair of input signals A and B, also treated as ‘Bit Propagate’ and ‘Bit generate’ [3] these signals are given by logic equations (2.6a) and (2.6b) [22].

\[
P_i = A_i \text{ xor } B_i \hspace{1cm} (2.6a)\\
G_i = A_i \text{ and } B_i \hspace{1cm} (2.6b)
\]

![Figure 2-9 Block diagram of propagate and generate logic](image)

2.3.4 Post processing
The calculation of sum is allocated to post processing, which is a simple implementation of XOR gate and it is calculated from the equation (2.7). This process is common to all the adder family involved in carry-lookahead network [22].

\[
S_i = P_i \oplus C_{i-1} \hspace{1cm} (2.7)
\]
2.3.5 Carry generation process

In KSA, carry bits are produced through a “Logarithmic Look ahead logic network” [3]. Inside the KSA, these bits are precomputed before the generation of complete sum. The carry propagation and generations are developed in a recursive tree for fast adders, especially in a KSA. This can be followed by decomposing the carry propagation into sub blocks of N-bits hierarchically, since wide gate and large stacks provide low performance and should be limited to 2 or 4 bits respectively [3]. For a four bit adder, carry bits at each stage are calculated from the equations (2.8a) to (2.8d) [22].

\[ C_{o,0} = G_{o} + P_{0}C_{i,0} \]  \hspace{1cm} (2.8a)
\[ C_{o,1} = G_{1} + P_{1}G_{0} + P_{1}P_{0}C_{i,0} = (G_{1} + P_{1}G_{0}) + (P_{1}P_{0})C_{i,0} = G_{1:0} + P_{1:0}C_{i,0} \]  \hspace{1cm} (2.8b)
\[ C_{o,2} = G_{2} + P_{2}G_{1} + P_{2}P_{1}G_{0} + P_{2}P_{1}P_{0}C_{i,0} = G_{2} + P_{2}C_{0,1} \]  \hspace{1cm} (2.8c)
\[ C_{o,3} = G_{3} + P_{3}G_{2} + P_{3}P_{2}G_{1} + P_{3}P_{2}P_{1}G_{0} + P_{3}P_{2}P_{1}P_{0}C_{i,0} \]
\[ = (G_{3} + P_{3}G_{2}) + (P_{3}P_{2})C_{0,1} = G_{3:2} + P_{3:2}C_{o,1} \]  \hspace{1cm} (2.8d)

2.3.6 Intermediate processing

Here in this process, calculating the carry’s corresponding to each bit uses group propagate and group generate signal. This network differentiates the Kogge Stone adder to any other adders that were described in the previous Sections (2.1) and (2.2) acts as a main reason behind being high performance complex adder [18]. Equations (2.9a) and (2.9b) provide the information for group propagate and group generate logic function.

\[ P_{i:j} = P_{i:k+1} \text{ AND } P_{k:j} \]  \hspace{1cm} (2.9a)
\[ G_{i:k} = G_{i:k+1} \text{ OR } P_{i:k+1} \text{ AND } G_{k:j} \]  \hspace{1cm} (2.9b)

\[ \text{Figure 2-10 Schematic for group propagate and group generated function} \]

Figure 2-10, theoretically represents an AND operator, which is responsible for calculating the group propagate and group generate function. For the group generate the circuit schematic is shown in figure 2-11 producing similar generate function as in the equation carries are generated at all \(2^{i-1}\) positions (i.e. 1, 3, 5, 7, 15,...) for \(i = 1 \ldots \log_{2}(N)\) through steps by exploring the associative property of a dot
product and the main advantage behind it is; it takes \( \log_2(N) \) times faster when compared to previous RCA and MCCA adders [3].

Figure 2 -11 Schematic representing group generate function
3 Simulation results

3.1 Test bench
A common test bench is shown in figure 3-1. The chosen input test vectors which drive the $C_{in}$ to $C_{out}$ is taken as critical path delay for the selected adders, it is designed in the 65nm CMOS process. The simulations are carried for reduced voltages and the data activity of every adder in this work is limited to a $C_{in}$ by applying a clock signal ranging from 0 to $V_{DD}$. When reduced supply voltage ($V_{DD}$) is taken, the same should be applied for $C_{in}$. The functional simulations are taken at near threshold operation as shown in figure 3-3.

Test vector:
A0-15: 1111111111111111
B0-15: 0000000000000000
$C_{in}$: 0 → 1

![Common test bench setup](image)

Figure 3-1 Common test bench setup
3.2 Voltage scaling technique
The technique of lowering the supply voltage is also called “voltage scaling” and has been used for estimating the power consumption of the circuits at lower voltages. During supply voltage scaling ($V_{DD}$) with a scale factor of 100 millivolt, firstly delay ($t_{delay}$) of an adder is calculated, which is taken from $C_{in}$ to $C_{out}$. Further, by inverting the delay, the frequency of operation has been calculated i.e. ($1/t_{delay}$). These values are used for finding the average power consumption. The other inputs such as $A_{0-15}$, $B_{0-15}$ should be in a propagation mode i.e. ($A_{0-15} = 1, B_{0-15} = 0$) or vice versa such that $C_{in}$ is propagated to $C_{out}$. This can be explained in figure 3-2. The results are collected for all the adders which are shown in tables (3-1), (3-2), (3-3), and (3-4).

![Figure 3-2 Dynamic behaviour of 16-bit adder](image)

3.3 Dynamic Power consumption
Static CMOS adders often dissipate power by charging various internal load capacitance, which includes wiring capacitance, gate capacitance and due to source and drain capacitance. During switching activity for one clock cycle, the current drifts from the $V_{DD}$ through the CMOS circuits for charging the load capacitance and drops the charge to ground by discharging. Therefore, the total charge over one cycle is $Q=C_{total}V_{DD}$ and is carried from $V_{DD}$ to ground. So the average current and average power were calculated through the following equation (3.1a) and (3.1b) [3].

$$I_{avg} = \frac{Q_{total}}{T} = \frac{V_{DD} \cdot C_{total}}{T}$$  \hspace{1cm} (3.1a)

$$P_{avg} = V_{DD} \cdot I_{avg} = \frac{C_{total} \cdot V_{DD}^2}{T} = C_{total} \cdot V_{DD}^2 \cdot F_{clk}$$  \hspace{1cm} (3.1b)

where $C_{total}$ is the load capacitance.

3.4 Power Delay Product
Power Delay Product (PDP) also known as switching energy or energy consumption per switching. It is defined as the product of power consumption and the delay taken from input to output. In this work the delay is the time taken from the critical path of each adder. It has been used as a metric correlated with energy efficient of logic gates [24].
<table>
<thead>
<tr>
<th>$V_{DD}$ (supply voltage)</th>
<th>Propagation delay ($t_{delay}$)</th>
<th>Average power $P_{avg} = I_{avg} * V_{DD}$</th>
<th>Operating frequency $f_o = 1/t_{delay}$</th>
<th>Power Delay Product (PDP) $P_{avg} * t_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>$ns$</td>
<td>$\mu W$</td>
<td>$Mhz$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.4</td>
<td>194</td>
<td>729</td>
<td>265</td>
</tr>
<tr>
<td>0.9</td>
<td>1.72</td>
<td>124</td>
<td>583</td>
<td>213</td>
</tr>
<tr>
<td>0.8</td>
<td>2.3</td>
<td>92</td>
<td>436</td>
<td>211</td>
</tr>
<tr>
<td>0.7</td>
<td>3.45</td>
<td>47</td>
<td>292</td>
<td>164</td>
</tr>
<tr>
<td>0.6</td>
<td>6.21</td>
<td>18.8</td>
<td>161</td>
<td>116</td>
</tr>
<tr>
<td>0.5</td>
<td>14.45</td>
<td>4.9</td>
<td>66</td>
<td>73</td>
</tr>
<tr>
<td>0.4</td>
<td>61.36</td>
<td>0.654</td>
<td>16</td>
<td>40</td>
</tr>
<tr>
<td>0.3</td>
<td>443</td>
<td>0.44</td>
<td>2.25</td>
<td>19</td>
</tr>
<tr>
<td>0.2</td>
<td>4003</td>
<td>0.002</td>
<td>0.23</td>
<td>19.3</td>
</tr>
</tbody>
</table>

Table 3-1 Simulation results for RCA

<table>
<thead>
<tr>
<th>$V_{DD}$ (supply voltage)</th>
<th>Propagation delay ($t_{delay}$)</th>
<th>Average power $P_{avg} = I_{avg} * V_{dd}$</th>
<th>Operating frequency $f_o = 1/t_{delay}$</th>
<th>Power Delay Product (PDP) $P_{avg} * t_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>$ns$</td>
<td>$\mu W$</td>
<td>$Mhz$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.98</td>
<td>107</td>
<td>1018</td>
<td>104.8</td>
</tr>
<tr>
<td>0.9</td>
<td>1.25</td>
<td>58</td>
<td>800</td>
<td>73.0</td>
</tr>
<tr>
<td>0.8</td>
<td>1.75</td>
<td>28.9</td>
<td>588</td>
<td>49.13</td>
</tr>
<tr>
<td>0.7</td>
<td>2.65</td>
<td>11.6</td>
<td>377</td>
<td>30.74</td>
</tr>
<tr>
<td>0.6</td>
<td>4.9</td>
<td>3.9</td>
<td>204</td>
<td>19.11</td>
</tr>
<tr>
<td>0.5</td>
<td>12.5</td>
<td>0.98</td>
<td>80</td>
<td>12.25</td>
</tr>
<tr>
<td>0.4</td>
<td>57.2</td>
<td>0.1</td>
<td>17</td>
<td>5.70</td>
</tr>
<tr>
<td>0.3</td>
<td>424</td>
<td>0.044</td>
<td>4</td>
<td>10.08</td>
</tr>
<tr>
<td>0.2</td>
<td>3036</td>
<td>0.002</td>
<td>0.5</td>
<td>12.25</td>
</tr>
</tbody>
</table>

Table 3-2 Simulation results for MCCA
3.5 Optimised results

We have taken several steps in order to minimise the delay of each selected architecture. Optimisation has been done with respect to sizing, progressive sizing, transistor ordering, and by logical effort [16]. Table 3-4 shows an improvement in delay for each adder at a maximum supply of 1V.

<table>
<thead>
<tr>
<th>Type</th>
<th>Before optimisation</th>
<th>After optimisation</th>
<th>% improved</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCA</td>
<td>1.5ns</td>
<td>1.37ns</td>
<td>10%</td>
</tr>
<tr>
<td>MCCA</td>
<td>1.6ns</td>
<td>1.04ns</td>
<td>35%</td>
</tr>
<tr>
<td>KSA</td>
<td>0.583ns</td>
<td>0.396ns</td>
<td>32%</td>
</tr>
</tbody>
</table>

Table 3-4 Optimised delay results

<table>
<thead>
<tr>
<th>$V_{dd}$ (supply voltage)</th>
<th>Propagation delay ($t_{delay}$)</th>
<th>Average power $P_{avg} = I_{avg} \times V_{dd}$</th>
<th>Operating frequency $f_o = \frac{1}{t_{delay}}$</th>
<th>Power Delay Product (PDP) $P_{avg} \times t_{delay}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>ns</td>
<td>$\mu$W</td>
<td>Mhz</td>
<td>$f_j$</td>
</tr>
<tr>
<td>1</td>
<td>0.396</td>
<td>1300</td>
<td>2500</td>
<td>514</td>
</tr>
<tr>
<td>0.9</td>
<td>0.4476</td>
<td>896.4</td>
<td>2100</td>
<td>426</td>
</tr>
<tr>
<td>0.8</td>
<td>0.638</td>
<td>496</td>
<td>1500</td>
<td>316</td>
</tr>
<tr>
<td>0.7</td>
<td>0.952</td>
<td>238</td>
<td>1005</td>
<td>226</td>
</tr>
<tr>
<td>0.6</td>
<td>1.600</td>
<td>90</td>
<td>600</td>
<td>144</td>
</tr>
<tr>
<td>0.5</td>
<td>6.006</td>
<td>20</td>
<td>150</td>
<td>132</td>
</tr>
<tr>
<td>0.4</td>
<td>17.75</td>
<td>3.48</td>
<td>50</td>
<td>61</td>
</tr>
<tr>
<td>0.3</td>
<td>130.5</td>
<td>285</td>
<td>7</td>
<td>37</td>
</tr>
<tr>
<td>0.2</td>
<td>1002</td>
<td>0.014</td>
<td>0.6</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 3-3 Simulation results for KSA
Figure 3-3 Simulation waveforms at near-threshold operation
4 Comparison of results and analysis

4.1 Linear comparison of delay
In CMOS logic circuits, delay is a key parameter, failure to consider it may lead to malfunction and it also varies with different logic styles. In the case of adder circuits, delay ($t_{\text{delay}}$) through the longest path is always proportional to the number of bits ($N=16$) (i.e. $t_d \alpha N$). The propagation delay ($t_{\text{delay}}$) is expressed in the equation (4.1). In RCA the delay is said to be linear as shown in figure 4-1. Whereas the delay from the MCCA is not linear initially before modifying the carry chain. The delay increases “exponentially,” this is because of the longest chain of transmission gates, which are connected in a cascoded form. This corresponds to an equivalent RC line constant consisting of both on-resistance and off- resistance of the NMOS and PMOS transistors of a TG. However, the delay of MCCA is made linear by employing buffers in a carry chain. These buffer acts as a repeater, which has not only made a significant improvement in linearity but also introduces a certain amount of delay and helps in rectifying low voltage swing across the nodes in the carry chain. The KSA has been eliminated in the graph for linear comparison, since the delay is a logarithmic function for carry bits, which are generated at $O(\log n)$.

$$t_{\text{delay}} = \frac{t_{\text{phL}} + t_{\text{plH}}}{2}$$

(4.1)

where $t_{\text{phL}}, t_{\text{plH}}$ are the respective high-to-low and low-to-high transitions.

![Delay Linearity](image)

**Figure 4-1** propagation delay with respect to the number of bits (N)
4.2 Operating Frequency

In digital CMOS circuits “operating frequency” represents the highest speed at which the circuits operate and produces the valid outputs. In these circuits, as the supply voltage is reduced then the frequency also reduces. We can notice this in the following graph shown in figure (4-2). The expression for the operating frequency is shown in the equation (4.2).

\[ f_o = \frac{1}{t_{\text{delay}}} \]  

(4.2)

where \( t_{\text{delay}} \) is the propagation delay of the carry chain.

The results have been plotted in figure 4-2 representing the operating frequencies of selected adder architectures, plotted for different supply voltages with a scaling factor of 100 millivolt. Among them KSA has the highest operating frequency at a maximum supply ranging from 900mV to 1V and have recorded the lowest frequency for low voltages ranging between 400mV to 700mV, while compared to a Ripple carry adder and a Manchester carry chain adder, as these adders are linear by the previous comparison shown in figure 4-2. So by careful observation, the operating frequency of Ripple carry adder and the Manchester carry adder decreases with a reduced voltages and linearly with change in voltage. The Manchester carry adder shows a gradual improvement over Ripple carry adder in terms of operating frequency and linearity.

![Operating Frequency](image_url)

Figure 4-2 Operating frequencies of adders at different supply voltages
4.3 Average power comparison

In order to increase the battery lifetime of portable digital electronics, power is the main issue to be addressed. By applying reduced voltage scaling techniques the problem with power consumption has been solved to some extent. Due to lowering the supply voltage, the power consumption associated with the CMOS circuits is reduced. The results obtained for operating frequencies on the three adder circuits are shown in figure 4-3. These results are utilised in finding the average current \( I_{\text{avg}} \) in the equation (3.1a) through the dynamic behaviour, figure 3-2. Thus the average power is calculated by substituting the value of \( I_{\text{avg}} \) in equation (4.3).

\[
P_{\text{avg}} = I_{\text{avg}} \times V_{DD}
\]

From the simulation results, a fair comparison is made for the average power consumption of all the 16-bit adder architectures. The collected results are plotted in figure 4-3 which shows that KSA has the highest power consumption, whereas the MCCA adder consumes the lowest power for each reduced voltage. So, we can conclude that the circuits with the highest operating frequency and more interconnects lead to consume higher power.

\[\text{Figure 4-3 Average power with reduced supply voltages (} V_{DD} \text{) of 16-bit adders}\]
4.4 Power delay product comparison

The power delay product has been used to quantify the efficiency of the digital design technology [24]. Although the speed of the CMOS circuits are dependent on the supply voltage and one alternate performance is the power delay product (PDP), which is defined in equation (4.4) [23] [24].

\[ PDP = P_{avg} \cdot \tau_p \] (4.4)

where \( \tau_p \) is the propagation time and \( P_{avg} \) is the average power dissipation.

PDP has been chosen as cost metric for optimising the digital circuits [23], it represents the average energy dissipated for one switching activity [24]. So in digital design technology, designing such circuits needs greater power. The unit of PDP is joules. From the simulation results, the graph shown in figure 4-4 plots PDP of the three adders with respect to voltage scaling. This proves that MCCA has the lowest PDP for all the reduced voltages, whereas Kogge Stone adder has the highest PDP.

![Figure 4-4 Combined PDP vs supply voltage (V_{DD}) of 16-bit adders](image)
5 Conclusion and future work

Finally, voltage scaling has been recognised as an acceptable technique in reducing power consumption. Through several comparisons made in the previous works described in Section 4 (Comparison of results and analysis), MCCA showed the best PDP for all the reduced supply voltages. The speed of the MCCA has been achieved with great linearity over reduced voltages. Similarly for KSA, being complex, the PDP is highest for all the voltages and consumes more power. From this, we can conclude that MCCAs are well suitable for low power circuit designs over different tunable voltages and operating frequencies within the range of 1 volt.

This work mainly contributes in the field of Very Large Scale Integrated (VLSI) circuit designs over several metrics related to reduced voltages, for estimating the trade-offs among power, speed and temperature in low power applications. It can also be employed as a base for other few advanced techniques such as dynamic voltage and threshold voltage scaling, which are common and potential techniques suitable for regulating the trade-offs. These advanced techniques are particularly involved in power management for battery powered devices such as mobile phones and PCs.
6 References


