ANALYSIS OF VARIOUS DCVSL STRUCTURES AND IMPLEMENTATION OF FULL ADDER WITH THEM

A THESIS SUBMITTED IN THE PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

IN

VLSI Design and Embedded System

BY

Subhrajit Roy

213EC2210

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela

Odisha, India – 769008

2015

ANALYSIS OF VARIOUS DCVSL STRUCTURES AND IMPLEMENTATION OF FULL ADDER WITH THEM

A THESIS SUBMITTED IN THE PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

IN

VLSI Design and Embedded System

BY

Subhrajit Roy

213EC2210

Under the Guidance of

Prof. Munshi Nurul Islam

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ODISHA, INDIA – 769008

2013-2015

Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela, Odisha – 769008

India

CERTIFICATE

This is to certify that the **Thesis Report** entitled

Analysis of various DCVSL Structures and Implementation of Full Adder with them

submitted by

Mr. Subhrajit Roy

bearing Roll No. **213EC2210**

in partial fulfilment of the requirements for the award of

Master of Technology

in

VLSI Design and Embedded System

during session 2013-2015 at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any degree or diploma.

Prof. Munshi Nurul Islam.
Department of ECE,
National Institute of Technology
Rourkela.

Date - 31.05.2015

Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela, Odisha – 769008

India

DECLARATION

I certify that –

- The work contained in this thesis is original and has been done by myself under the general supervision of my supervisor.
- The work has not been submitted to any other Institute for any degree or diploma.
- I have followed the guidelines provided by the Institute in writing the thesis.
- Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
- Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving the required details in the references.

SUBHRAJIT ROY

subhrajit.royy@gmail.com

Acknowledgement

First and foremost, I would like to express my sincerest appreciation to my Project Guide, **Prof. Munshi Nurul Islam**, for his guidance and tremendous support throughout the course of this project. His knowledge and expertise have been really helpful in this research work. The regular discussions with him and his ideas for approaching each problem statement in a different manner have been extremely useful. Without his utmost help, this dissertation would not have been possible.

Next, I would like to thank **Prof. K. K. Mahapatra**, **Prof. P. K. Tiwari**, **Prof. D. P. Acharya**, and **Prof. A. K. Swain** for their thoughtful teaching and suggestions during my course of M.Tech and making available all necessary facilities and infrastructure for studies.

Next, I would like to express my thanks to all the Ph. D. and Research scholars who have readily provided a helping hand in my time of need and acknowledged my numerous requests without any pique regarding any technical support.

I would also like to thank all the other faculty members and non-teaching staff of the Department of ECE for helping me directly or indirectly in completion of this project.

Away from home in a distant land, the people who constantly support and encourage a person are his friends. I am utmost grateful to my friends, here at NIT Rourkela. Apart from the regular education from my professors, I have learnt many things from my friends. I would like to express my thanks to all of them for making these two years a fantabulous journey.

Finally, I owe my heartiest gratitude to the most important people in my life, i.e. my parents and my elder brother for their unconditional support, love, inspiration and sacrifices. They have always been a source of inspiration throughout my life. The discussions with them, especially my brother, have assisted me through many a time of ordeal. I would like to thank them from the bottom of my heart.

SUBHRAJIT ROY

Dedicated To

My Parents, My Brother

and

All My Dear Friends

Abstract

The Differential Cascode Voltage Switch Logic (DCVSL) is a CMOS circuit technique which has potential advantages over conventional NAND/NOR logic in terms of power dissipation, circuit delay, layout density and logic flexibility. In this paper, a detailed comparison of all the DCVSL structures are provided including the implementation of Full Adder circuit with the help of those DCVSL structures, which includes Static DCVSL, Dynamic DCVSL and Modified DCVSL. The performance analysis is done in Cadence Virtuoso 90nm CMOS Technology.

The working of these DCVSL structures is based on the concept of 'Multiplexer'. A multiplexer also known as 'mux', which is a device where from a number of input signals, selection is done. It is basically a combinational logic circuit. The multiplexer is a unidirectional device, which is used in applications where a data must be switched from multiple sources to a destination.

The analysis of all these DCVSL structures is followed by the implementation of Full Adder. Adders are the building blocks in computer systems. Digital Computer Systems widely uses Arithmetic operations. Addition is a necessary arithmetic operation, which is also the root for arithmetic operation such as multiplication. Similarly, adding another XOR gate, the basic adder cell can be modified to function as subtractor, which can be used for division. Therefore, 1-bit Full Adder cell is the ultimate and simple block of an arithmetic unit of a system. So, the basic 1-bit Full Adder cell must be improved, so that the performance of the digital circuits. In VLSI, there is always a trade-off between speed and power dissipation. One parameter is improved, the other gets degraded. Hence, the parameter power delay product is introduced.

So, to achieve speeds, high drivability hybrid-DCVSL design methodologies are used to build adder cell in this work. The DCVSL gates produces both complementary and true outputs using single gate architecture. And, the multipliers in the design are based on the pass transistor logic (PTL), because these occupies less chip area per component and also are simple to construct.

The parameters compared are power dissipation, propagation delay time, power delay product, transistor number and power dissipation (average). The Static DCVSL structure produces best result in terms of power dissipation, delay and power delay product. Whereas, in case of the Adder circuit, the power consumption is best for the Dynamic DCVSL Adder, along with the delay and the power delay product for the output Sum; but for the output Cout, the best option is Static DCVSL Adder, as the delay and the power delay product is least in this case.

CONTENTS

List of Figures

List of Tables

Chapter 1

INTRODUCTION

(1.1) Topic Overview

A comparison between Differential Cascode Voltage Switch Logic (DCVSL) and traditional NAND/NOR Logic resulted in the former having several advantages over the latter one in terms of logic flexibility [1], device count, layout area, power dissipation and circuit delay. Less number of transistors are required, including both p and n- type than the NAND/NOR Logic [1]. It also provides advantages such as stuck-at and dynamic faults because of its self-testing property [2]. A leverage circuit provides a perfect example where a single CVS gate is obtained from numerous stages of delay [3]. The CVS circuit has increased stack height but it also has certain advantages such as smaller chip area, lesser power dissipation and shorter circuit delay which is achieved due to the decrease in quantity of stages. The advantage of logic flexibility is achieved in situations; where in Domino CMOS Logic, complex functions are implemented. As Standard Domino Logic cannot implement functions having inverting logic gates, hence Clocked DCVS Logic is used which overcomes this restriction by providing complementary outputs [1]. The various advantages of DCVS Logic over the standard CMOS NAND/NOR Logic can be outlined as-

- Circuits having great complication and fan-in gates can be implemented as it may be done with lesser transistor count. For definite circuits having large complex gates, the clocked DCVS Logic families have very low propagation delay since it can integrate a single complex gate from both sequential and combinatorial portions. Hence, for high-speed VLSI this logic style is appropriate.

- In DCVS Logic, the faster response comes through the reduction of parasitic capacitances at the output as it is not having a complementary pull-up network. Therefore, it has the speed advantage over domino circuit and also eliminates the static power consumption.

- The output voltage swings from rail to rail and it doesn't provide the direct current path between V_{DD} and ground in steady states.

- In DCVS Logic, the completion of gate evaluation is easy to detect since true and complementary outputs are formed. For this reason, DCVS Logic is the suitable choice for implementing self-timed circuits.

- Due to its true and complementary outputs, the performance is further improved by elimination of inverting stage [4].

- While the standard domino logic cannot implement inverting logic gates, but the DCVSL logic style can implement both inverting and non-inverting logic style [1].

- The DCVSL circuit also saves area by sharing the common transistors in the logic network for both of the outputs when a complex logic gate is designed.

These above advantages indicates the DCVS Logic as a new dimension in CMOS Logic design. However, along with these advantages there are certain disadvantages such as the high power along with extra area and complexity due to dual logic networks having complementary signals are the definite hindrances towards their acceptance as a design logic. The power consumption in DCVS gate comprises of the outputs switching and the switching at internal nodes of the gate. With gate complexity, the number of internal nodes switched in NMOS evaluation tree increases. The switching in internal nodes is the dominant factor of total gate power.

The **Algebraic Technique** is used for the design of DCVS trees, which is the ultimate existing procedure and is also based on the identification of sub-expressions which is common to two or numerous Boolean functions [5]. It also has decomposition and factorization technique which is quite mathematical, hence this method is not suitable for IC designers as it doesn't provide the insight into circuit behaviour. Along with the algebraic technique, there are two other techniques which are more practical and also much simpler for constructing DCVS trees.

- The **Karnaugh Map (K-Map) Method** – Here, the pictorial nature is used. This hand-processing method is quite efficient in realizing circuits with low device count and is implemented with functions having five or six variables. More than the specified variables, the complexity of K-maps increases.

- The **Quine-McCluskey Method** – It is uniform and hence procedural for complexities upto n number of variables and it is tabular in nature [6].

The worthwhile features of the DCVS Logic makes it a very promising CMOS circuit technique. To investigate this possibility, we have done analysis on various types of DCVS Logic namely,

- **Static DCVSL**.
- **Dynamic DCVSL**.
- **Modified DCVSL**.

Along with it, we have also implemented these various DCVS Logic on Full Adder circuit since the full adder is common, yet reasonably complex building block in digital circuits and design. The work is done on CADENCE VIRTUOSO 90nm technology to assess the performance parameters of power dissipation, delay, area, speed and power delay product. The area is represented from the layout drawn in cadence. Speed is reflected through the worst-case propagation time.

(1.2) Literature Review

When the differential pairs of MOSFET devices are cascoded into strong combinational logical tree networks, then we can achieve a design leverage in CVSL which is within a single circuit delay and is capable of implementing complex Boolean logic. When there is a Boolean function which might have input variables upto (2^N-1) is being processed with N-high cascoding of differential pairs of NMOS devices. CVSL offers a performance advantage of up to 4X compared to standard CMOS NAND/NOR logic families, while maintaining the expected low power characteristics of CMOS circuitry. The primitive NAND/NOR Logic and the CVSL, both are potentially dense and are well-suited with previous design automation tools. Using cascoded high-performance NMOS devices, logic trees which are compatible in nature are designed and the unstacked P mosfet devices are used as pull-up devices in load and buffer circuitry. Therefore the P mosfet devices can be optimized and hence critical spacing between P and N devices are relaxed, releasing the device complexity burden for CVSL designs. CMOS is widely used in making logic circuits, but then also the DCVSL has its own features such as no static power, higher speed [7] as it produces complementary outputs due to dual rail logic and is very efficient in designing full adder circuits [8]. Heller et.al presented DCVSL [1] & then Chu et.al compared the conventional and DCVS Logic Full Adder circuit [9], [10]. Full Adder is considered as a vital building block for circuits like adders and multipliers, therefore reducing the delay would improve the circuits in terms of speed [7]. In Very Large Scale Integrated Circuits, by abandoning the signal slope effect, similar value of capacitance for rising and falling transistors is obtained with uniform gate capacitance and diffusion capacitance rate/unit width and assuming uniform P to N size ratio for various logic gates; the delay modelling for conventional circuits is done. Along with it, Shockley's α-power law is used to calculate the MOS transistor current where the α is a value amid 1 and 2 and is termed the velocity saturation index [7]. There is an additional precise model for MOS transistors proposed in [11] by Shams with same level of complexity. The propagation delay of conventional and some unconventional logic styles like DCVSL, CPL and PTL can be modelled and designed accurately and exhaustively using the stated current relation supposing non-similar capacitance values for rising and falling transistors [11], [12] and also the signal slope effect. This helps us in optimizing CMOS circuits which is of mixed styles of logic. Figure 1 shows the structure of a DCVS circuit which consisting of a load which is push-pull in formation and also consist of a couple of interconnected binary decision trees.

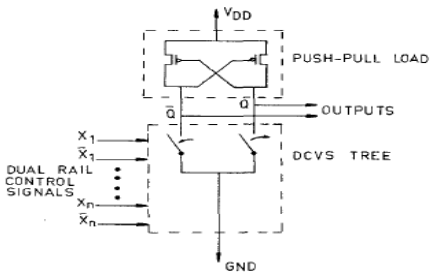

Figure 1 – The structure of a DCVS circuit

(1.3) Thesis Overview

Chapter 1 introduces the topic, discusses about the importance of DCVSL structures, its evolution, past significance, advantages and disadvantages. It also gives an idea of its discovery as mentioned in the Literature Review portion of it.

Chapter 2 starts with the details of the Differential Cascode Voltage Switch Logic (DCVSL), it shows the basic structure of DCVS circuit. Apart from that, three different types of DCVSL structures are defined, namely Static DCVSL, Dynamic DCVSL and Modified DCVSL.

Chapter 3 gives the idea about the circuit techniques by which one can measure the various parameters of DCVSL structures. Here, three types of circuit techniques are explained – Algebraic Method, K-map Method and Tabular Method; out of which only algebraic method is used in recent forms.

Chapter 4 gives the idea about the testing schemes of the DCVSL circuits.

Chapter 5 gives the detail study of each of these DCVSL structures, with parameters such as power consumption, temperature, delay, PDP and transistor number. These analysis determines the best DCVSL structures among the three.

Chapter 6 uses the three DCVSL structures to implement the adder circuits with each of them. A performance analysis is done with the same parameters used in the previous chapter and helps determine the best DCVSL structure for the adder circuit, as per the power delay product.

Chapter 7 – This chapter concludes with the overall work done in this thesis, generalising the best DCVSL structure, whether it may be the structure itself or the implementation of adder circuit with it. This chapter ends the thesis work.

Chapter 2

DETAILS on **DIFFERENTIAL CASCODE VOLTAGE SWITCH LOGIC**

(2.1) Introduction

With the advancement in CMOS technology, there is a new interest in designing simple functional units for digital systems. ICs are widely used in consumer electronics, telecommunications and high performance computing. This is to continue with power-efficient VLSI and system designs. CMOS circuits are normally used by digital integrated circuits as building blocks. Power Consumption is a chief worry in VLSI with increasing chip density and operating frequency along with corresponding decreasing feature size. The main setback of portable systems is excessive power dissipation which not only reduces chip life due to overheating but also degrades performance [13], [14]. Portable systems with low power consumption has directed to advanced developments in Low Power VLSI design in current years. The driving forces that are essential for portable devices are low power consumption and high throughput due to their increased complexity, small chip area, large density of components and high frequencies. A DCVS Logic is based on 2:1 Multiplexer which is used as an important element in many various circuit designs such as implementation of memory circuits and FPGA. It is valuable in situations where price is a factor and for modularity. A 2:1 Multiplexer is considered to be the basic building block of the "switch logic". "Switch Logic" basically proposes that logic circuits are implemented not as logic gate but as combination of switches. Multiplexers are used to create a single line from two or more digital signals, by engaging them there at changed times. This is basically known as Time-Division Multiplexing. Multiplexers are used as programmable logic devices, such as in digital video, computer networks and in telecommunications. It is also used in building digital semiconductors such as in graphics controller and in CPUs. The Multiplexers are based on Pass Transistor Logic (PTL) where the transistors are used as voltage controlled switches to implement the logic. The logic is bidirectional and there is no static power dissipation in PTL gates. However, the PTL suffers degradation of its logic at the output by an amount of Threshold Voltage (V_{th}). Figure 2 shows the schematic of DCVSL structure which is quite general, along with corresponding drive & also having load and then the output and its complementary output of the gate on both sides.

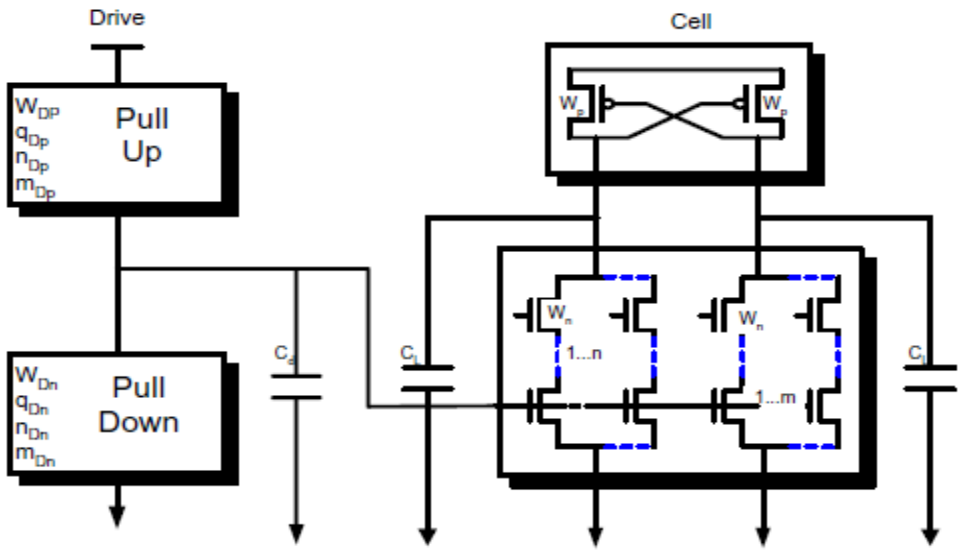

Figure 2 – DCVSL structure

So, we can see, according to evenness, the loads obtained from the result are equivalent to Lumped Capacitances including both fan-out & interrelated capacitances. The rising delay and falling delay amid the drive's input and the cell's output are shown by [11].

$$De^{'} = \left(1 + Sf_{n}^{`}\right)De_{D}^{'} + \left(1 + Sf_{p}^{'}\right)De_{C}^{`} + De_{C}^{'} \text{.....(1)}$$

$$De^{'} = \left(1 + Sf_{n}^{`}\right)De_{D}^{'} + De_{C}^{`} \text{.....(2)}$$

where De_D – driver step delay, De_C – cell step delay, '\' and '/' – rising and falling transistors. Sf_N and Sf_P are the N & P's slope factors & its calculation is done by Sakurais Relation [15], where V_{TH} is Threshold Voltage of transistor, De_T is ramp delay; τ is the rise/fall time of input and α is the velocity saturation index.

$$Sf = 2\left(\frac{1}{2} - \frac{1 - \frac{V_{TH}}{V_{DD}}}{1 + \alpha}\right) \text{.....(3)}$$

$$\alpha = \frac{\left(1 - \frac{V_{TH}}{V_{in}}\right)}{\left(\frac{1}{2} - \frac{1 - De_{T}}{\tau}\right)} \text{.....(4)}$$

In DCVSL gates, output which is falling produces the rising output, therefore rising delay always represents the worst case delay. Calculation of rising delay of circuit can be obtained using the above scenario and is shown in the equation [11]:

$$\dot{D}e = \left(1 + \dot{Sf}_{T_n}\right)\dot{De}_{D} + \left(1 + \dot{Sf}_{T_p}\right)\dot{De}_{C} + \dot{De}_{C} \text{.....(5)}$$

$$\dot{Sf}_{T_n} = \frac{\dot{Sf}_{n}\, DM_{n}\left(\dot{\upsilon}_{n}\, \dot{Y}_{n}\right)}{DM_{n}\left(\dot{\upsilon}_{n}\, \dot{Y}_{n}\right) - DM_{p}\, \upsilon_{p}^{'}} \text{.....(6)}$$

$$\dot{De}_{D} = \frac{\upsilon_{p}^{'}}{DM_{D_p}} Y_{D_p}^{'}\left(m\, g_{n}^{'}\, DM_{n} + C_{D}^{'}\right) \text{.....(7)}$$

$$\dot{De}_{C} = \frac{\upsilon_{p}^{'}\, \dot{\upsilon}_{n}\, \dot{Y}_{n}}{DM_{n}^{'}\, \dot{\upsilon}_{n} - 0.5 DM_{p}\, \dot{\upsilon}_{n}\, \dot{Y}_{n}} \times \left[\left(\dot{gc}_{p} + \dot{dc}_{p}\right)DM_{p} + q\, \dot{d}_{n}\, DM_{n} + \dot{C}_{L}\right] \text{.....(8)}$$

$$\dot{De}_{C} = \frac{\dot{\upsilon}_{p}}{DM_{p}}\left[\left(g\dot{c}_{p} + \dot{dc}_{p}\right)DM_{p} + q\, \dot{dc}_{p}\, DM_{n} + \dot{C}_{L}\right] \text{.....(9)}$$

where $\dot{gc}_{p}$ and $\dot{dc}_{p}$ are represented as gate and diffusion capacitance/unit length in the rising output stage.

Solving $\frac{\partial D^{'}}{\partial W_{p}} = \frac{\partial D^{'}}{\partial W_{n}} = 0$ gives us the following set of equations for the optimal DM_p and DM_n for the delay minimization [11], [12]:

$$DM_n' = \sqrt{A\frac{(1+Sf_p')\left[C_L' + DM_p\left(dc_p' + gc_p'\right) + 0.5ADM_p d_n'\right]}{(1+Sf_n')Y_{D_p}' mg_n' / DM_{D_p} + qd_n' / DM_p}} + \frac{1}{2}ADM_p \quad \text{.....(10)}$$

$$\frac{1}{\Gamma'} = \frac{DM_n'}{DM_p'} = \sqrt{(1+Sf_p')\left[\frac{1}{2}A^2\frac{C_L' + qDM_n d_n'}{C_L' + qDM_n d_n'} + A\frac{W_n\left(dc_p' + gc_p'\right)}{C_L' + qDM_n d_n'}\right]} + \frac{1}{2}A \quad \text{.....(11)}$$

where Γ is used for minimizing the rising delay as it is the finest width ratio of P to N of MOS structure, S is slope factor, Y is the delay degradation factor, C_L is the total output capacitance/unit length, g is the gate capacitance/unit length and d is the diffusion capacitance/unit length. $A = n/\sqrt{\mu_n/\mu_p}$. With n increasing, the pull-down network becomes weaker and the theory of delay optimization for the Conventional Logic Style is not applicable to the DCVSL gates directly [7], [11].

(2.2) Basic DCVSL Circuit

The DCVSL circuit is illustrated in Figure 3.

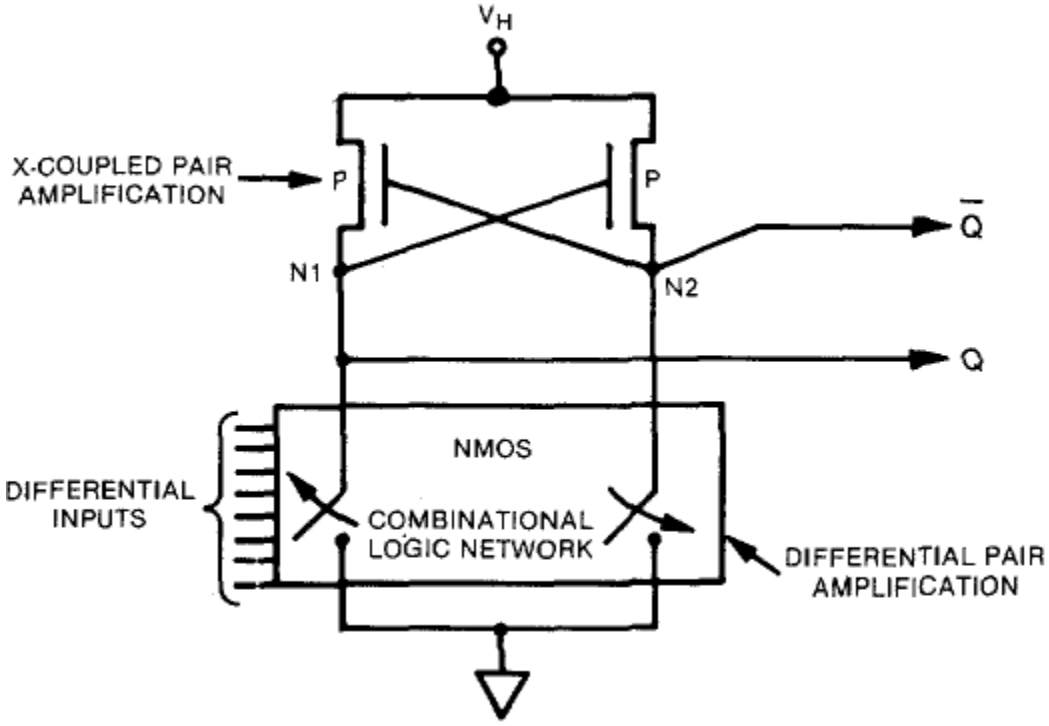

Figure 3 – Basic DCVSL circuit

Either the node N1 or the node N2 is pulled down by the NMOS combinational logic tree network which depends on the differential inputs. The PMOS latch is set to the static outputs Q, Qbar of full differential V_H and ground logic levels by regenerative actions. After the latch sets, the logic trees are being cut-off from direct current. The input gate capacitance loading is a factor of 3X smaller than CMOS circuits which requires a complementary N-channel and P-channel devices to be driven since the NMOS tree devices is driven by the inputs only. Using the existing logic minimization algorithms [5], logic tree networks are

designed spontaneously. Example of DCVSL circuit, having 12 devices is shown in Figure 4. The differential version has six less large P-channel devices & also the input capacitance is small in number.

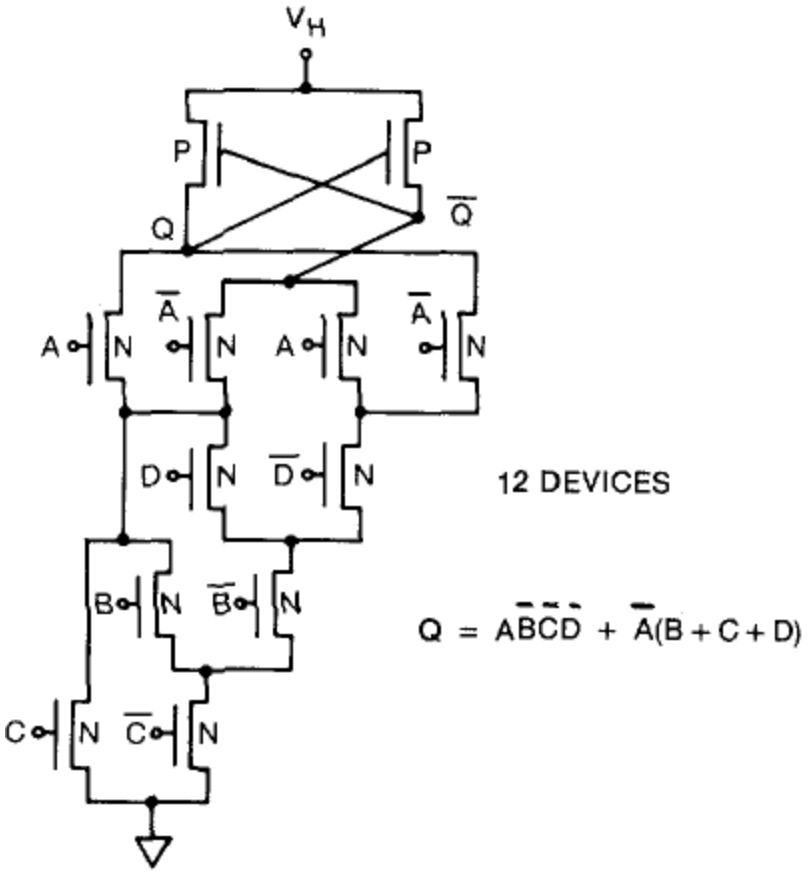

Figure 4 – CVSL implementation of Q

The functional power of the differential logic trees reduces the device redundancy. Using the similar Boolean function in simple CMOS NAND gates, requires 5 NAND gates and 28 devices, without having additional inverters for the complementary outputs, shown in Figure 5.

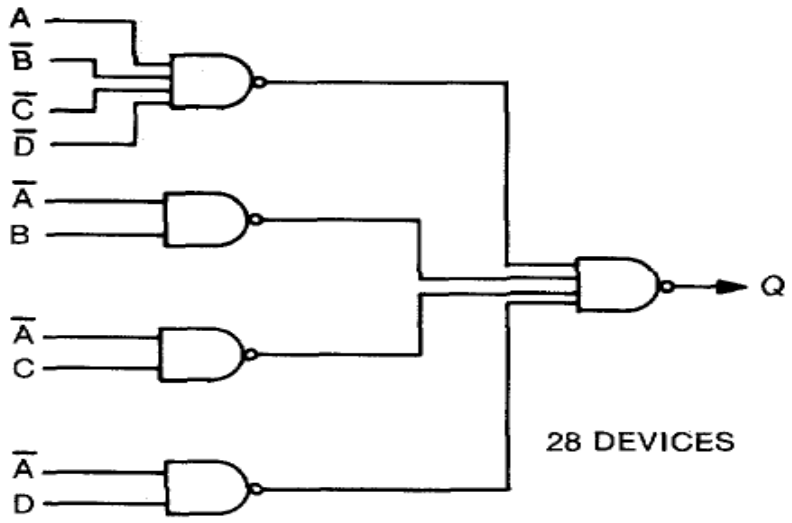

Figure 5 – CMOS NAND implementation of Q

The number of circuit delays is reduced in CVSL compared to primitive logic, which enhances the performance leverage of the circuit. The Boolean function Q can also be designed in a cascaded fully CMOS circuit with 16 devices.

There is a limitation to the circuit's performance in Figure 3 which is considered as the set time for the P mosfet latch. Therefore a high-performance Clocked CVSL circuit is proposed, which is shown in Figure 6.

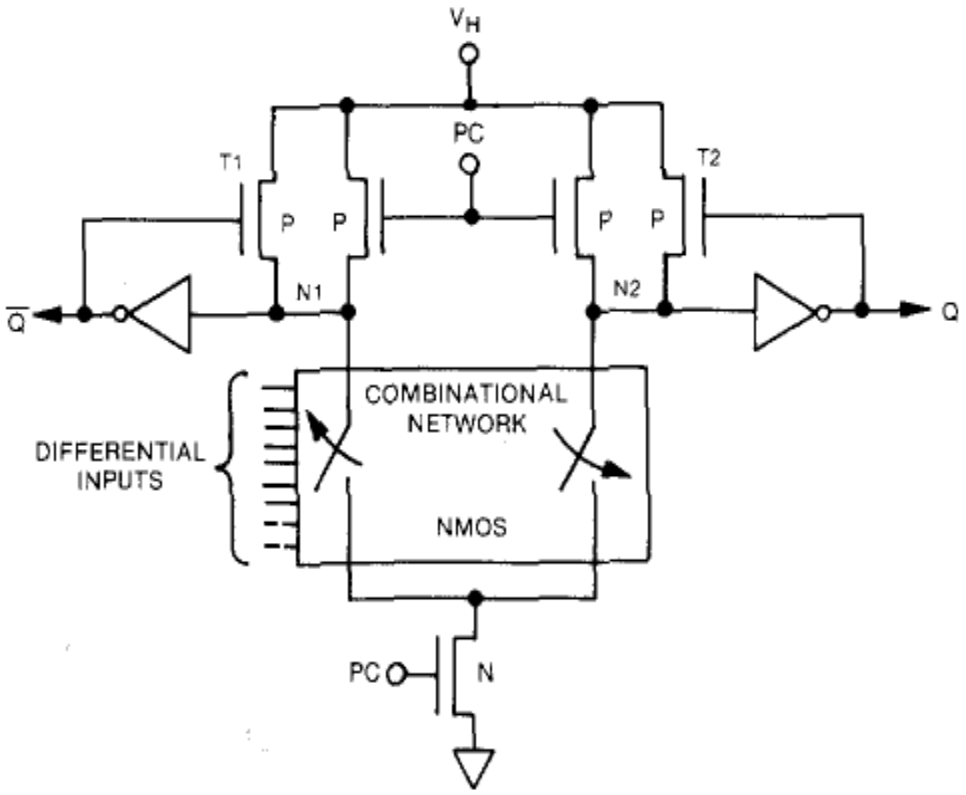

Figure 6 – Clocked CVSL

When the clock pulse is low, the outputs Q & Qbar are precharged low and as soon as the PC goes high, the data are generated in a domino mode [16]. The internal nodes N1 and N2 are set statically high preceding the switching inside the logic tree by the feedback devices T1 and T2. The feedback devices improves the noise margin with only a little bit of detriment in performance and it also reduces the charge sharing noise within the tree. Either the node N1 or the node N2 is drawn downward during switching & either of the device T1 or the device T2 is shut off. And after switching, there was no direct flow of current. The Clocked DCVSL circuit is inherently a clear benefit over the other partial domino type logic families due to the logic invert function. All the logic functions could be executed including the XOR. A 4-way clocked XOR is in Figure 7.

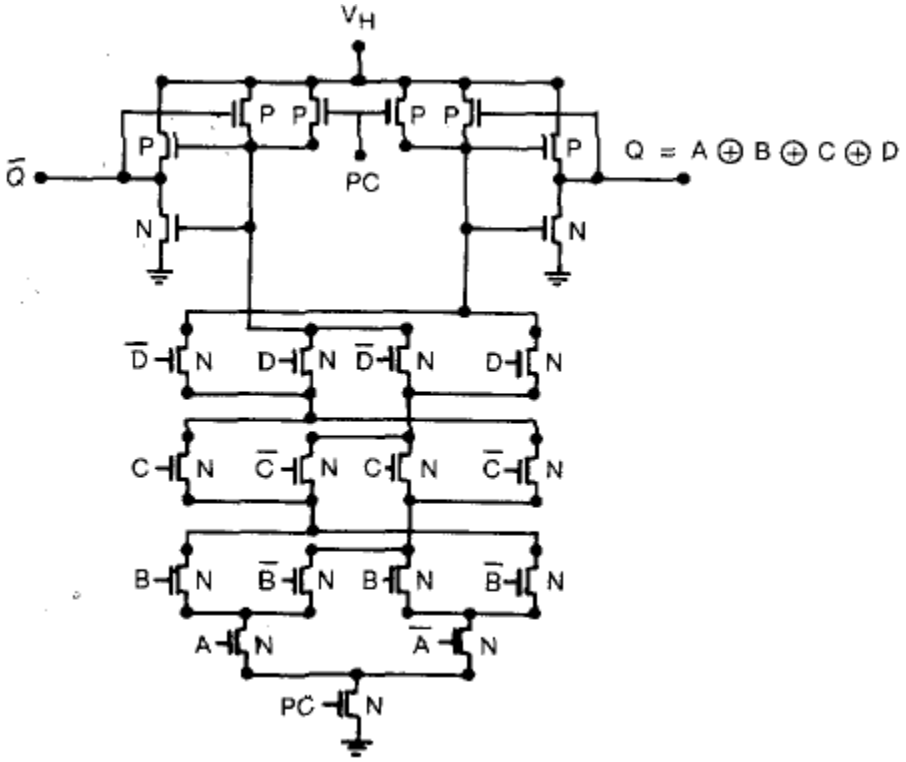

Figure 7 – Clocked CVSL 4-way XOR

(2.3) Different Types of DCVSL structures

In this paper, we have explained three types of DCVSL structures –

- **Static DCVSL** – It is actually a static version of Differential Cascode Voltage Switch Logic (DCVSL) and is shown in Figure 8.

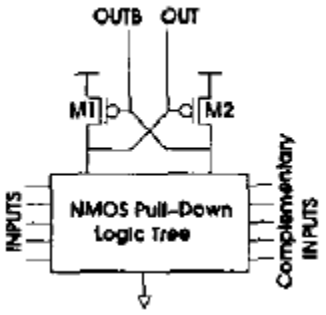

Figure 8 – Static DCVSL

According to the figure shown above, the nodes OUT and OUTB are either pulled high or low according to the switching of the inputs. The static version of DCVSL transits slowly and it consumes high current since during the switching period, the PMOS pull-ups fight the NMOS pull-down trees. It is the differential style of logic in which the true and complementary inputs to the gate provides the complementary outputs. This structure consumes no static power (like standard CMOS) and it utilizes latch to calculate output rapidly [14], [17], [18]. In this logic style, large PFETS are eliminated from each logic function. It allows complex gates, doesn't need inverter in its logic path and consumes low power. A logic function and

complement of it is inevitably realized [18], [19]. The complementary output is generated by the pull-down network which is implemented by the NMOS logic tree. Therefore, it can be divided into two basic parts – a differential latching circuit and a cascaded complementary logic array [1], [20], [21], [22], [23].

- **Dynamic DCVSL** – Many clocked versions of DCVSL gate was also introduced to increase the performance and reduce the power consumption. Figure 9 shows that

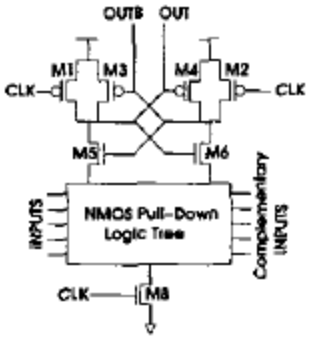

Figure 9 – Dynamic DCVSL

The PMOS pull-up transistors are connected by two complementary NMOS switch structures. By setting the clock low, OUT and OUTB are pre-charged first. The NMOS logic tree determines the true and its complementary output, once the clock goes high and either side is pulled down depending upon the input signals. The gate switches when the positive feedback is applied to the PMOS pull-ups (M3 and M4). With the additional accelerating circuitry (M5 and M6), the performance of the dynamic DCVSL gate is improved. Here are few examples shown in Figure 10.

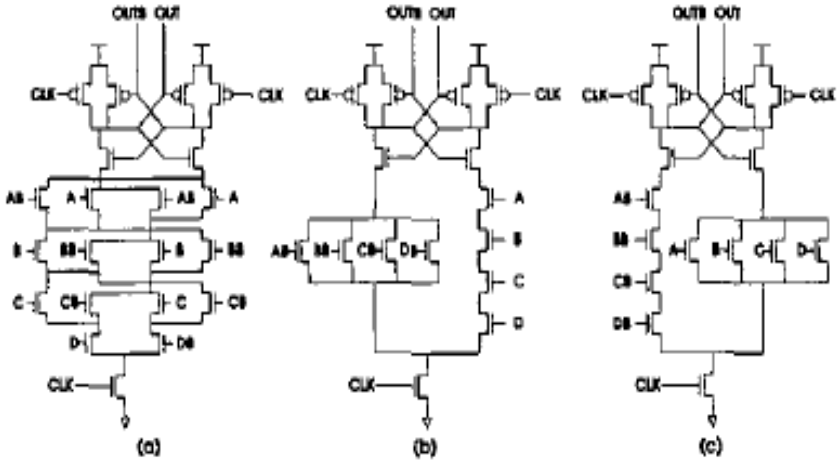

Figure 10 – (a) XOR-XNOR, (b) AND-NAND, and (c) OR-NOR

- **Modified DCVSL** – The modified DCVSL is nothing but the static DCVSL but with extra added NMOS transistors in the pull up part.

Chapter 3

CIRCUIT TECHNIQUES for **DCVSL**

(3.1) Introduction

The algebraic technique is the only prevailing method for the design of DCVS trees which is constructed on the basis of sub-expressions mutual to various Boolean functions which is two or more in number [5]. In this approach, the decomposition and factorization technique is involved which is quite mathematical. Therefore this method doesn't provide the details of circuit behaviour which is important for IC designers. There are other two techniques which is simple and also very practical than the algebraic one, when construction of DCVS trees is to be done. The first procedure is the Karnaugh Map which depicts the pictorial nature. It is a hand-processing method and is a well-organised approach for realizing circuits with low device-count, with functions having five to six variables. Moreover, if the variables increases more than five, then the complexity of K-map suddenly increases. Here comes the second procedure which replaces the first one, due to the advantage of having even routined complexity for variables in terms of n numbers. So the method is Quine-McCluskey method which is also a modified version and also tabular in nature [6]. There is no existence of a unique, one-to-one correspondence between a DCVS tree structure and a Boolean expression [24]. Therefore for realizing specific logic operation, several tree structures can be constructed using the above design procedures. Several of the input variables might be acceptable to rearrange for a given structure.

Therefore for construction of Boolean function of any numbers, one certainly need the help of truth tables which must be accurate too. DCVS structures can be implemented with the two design procedures mentioned here.

(3.2) Different Techniques

- **Design by Intuition** – DCVS circuits consisting of a couple of interrelated binary decision trees and a push-pull load. It is shown in Figure 11.

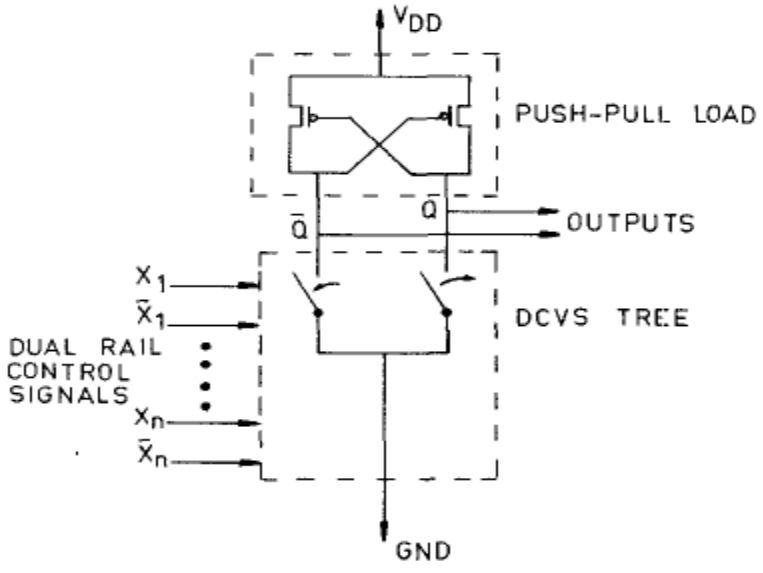

Figure 11 – Structure of DCVSL

The DCVS tree is designed as such –

- ✓ Node T and Node T' (considered T in case of Q) is disconnected and associated to ground respectively via some sole conducting path over the trees, when the input vector a = (a_1 , a_2 ,…., a_n) {instead of x, a is considered} is the vector which in true in nature of the switching function Q(a).
- ✓ The reverse is obtained when a = (a_1 , a_2 ,…., a_n) is the vector which is false in nature of Q(a).

An example is shown in Figure 12.

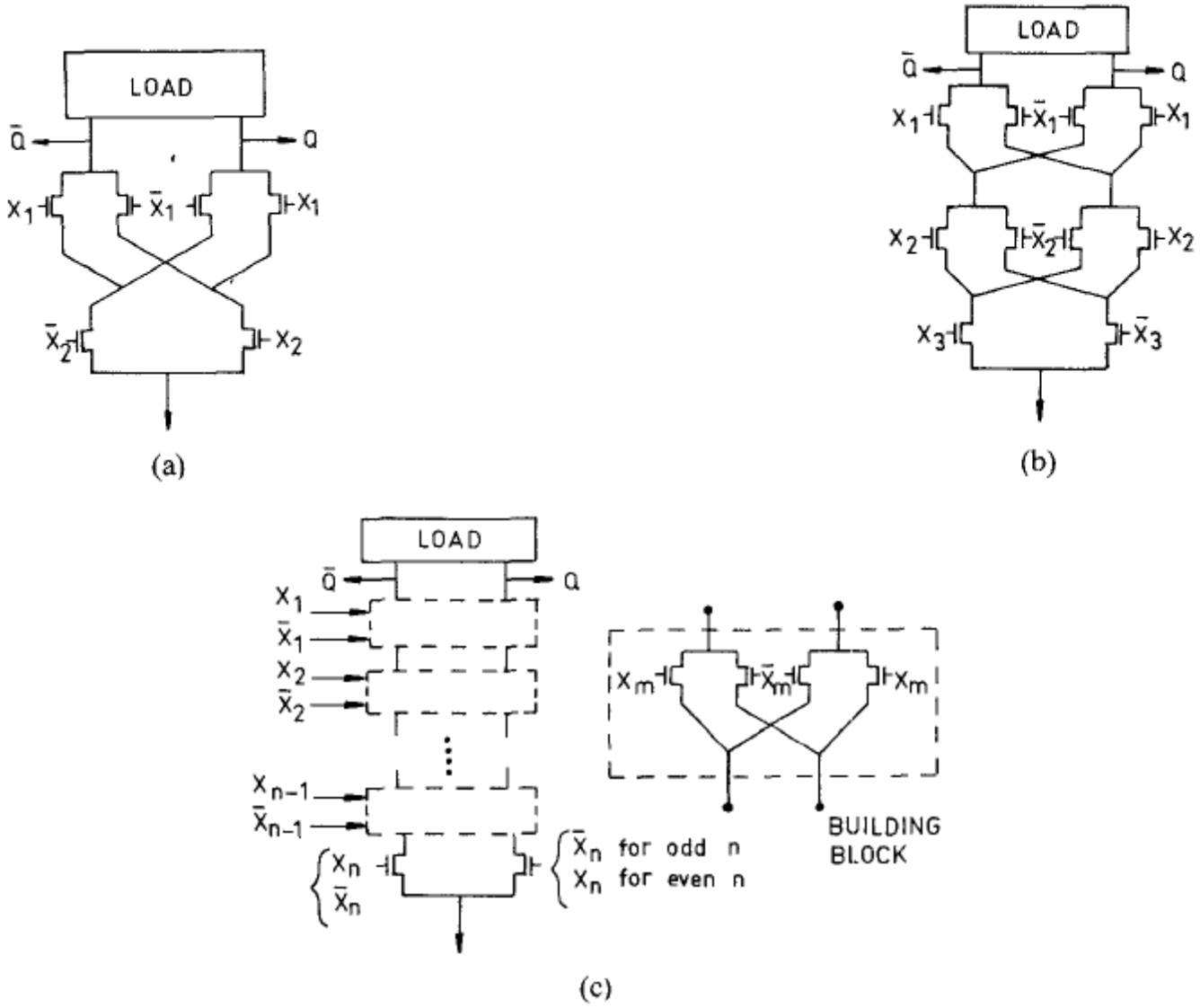

Figure 12 – DCVS XOR circuits – different way gates

Through analysing all the likely blends of the input paths, the functionality of this circuit can be easily verified. Then also we can verify the circuit by observing the set of unique paths from nodes T and T' to the ground. For Node T, the expression T'(a) = $a_1^{'}a_2^{'} + a_1a_2$ and for Node T', the expression T(a)= $a_1a_2^{'} + a_1^{'}a_2$.

For Boolean functions with a recursive nature, the DCVS tree can sometimes be created effortlessly by perception. Example such as, a 3-way XOR tree (Figure 12(b)) can be implemented by replacing the $a_2, a_2^{'}$ couple in (Figure 12(a)) with alternative 2-way XOR tree. The n-way XOR tree is shown in Figure 12(c), which is having a stacking height equal to n.

Boolean functions with recursive nature can also be shown in carry-look ahead circuit [25]. Given the recursive expression –

$$cla_n = s_n + t_n cla_{n-1} \text{ (for n = 1, 2, 3,….)},$$

a circuit with cla_n and $cla_n^{'}$ (considered instead of c_n and $c_n^{'}$) as outputs, with input vector equal to $\left(s_n, s_n^{'}, ..., s_1, s_1^{'}, t_n, t_n^{'}, ..., t_1, t_1^{'}, cla_0, cla_0^{'}\right)$ {considered instead of $\left(g_n, g_n^{'}, ..., g_1, g_1^{'}, p_n, p_n^{'}, ..., p_1, p_1^{'}, c_0, c_0^{'}\right)$ }. First of all, the function $cla_1 = s_1 + t_1 cla_0$ is realized as the circuit in Figure 13(a), and is the basic circuit for recursion. The general structure for cla_n with stacking height equal to 2n+1 is shown in Figure 13(b).

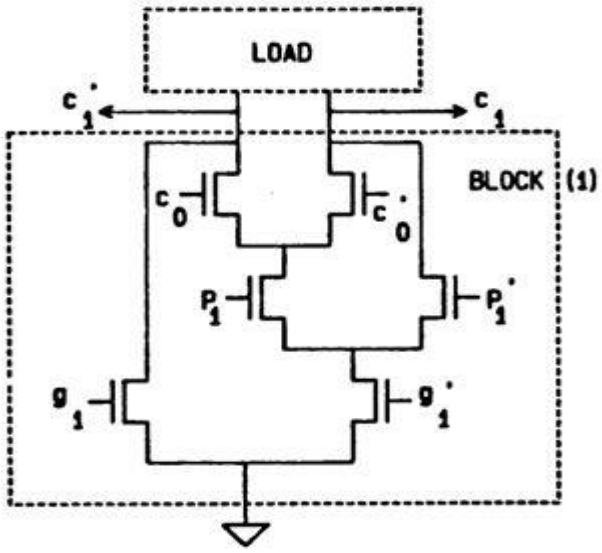

Figure 13(a) – Function $cla_1 = s_1 + t_1 cla_0$ for DCVS circuit

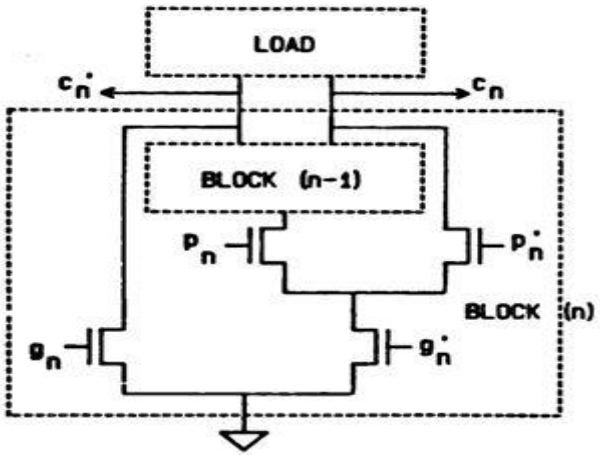

Figure 13(b) – Recursive DCVS structure for function $cla_n = s_n + t_n cla_{n-1}$

It is easy to construct the DCVS tree network, for Boolean expressions consisting of only a few product terms. Considering a simple function S (considering P as S) = $a_1a_2 \ldots a_n$; the corresponding structure and its symbolic representation are shown in Figure 14(a).

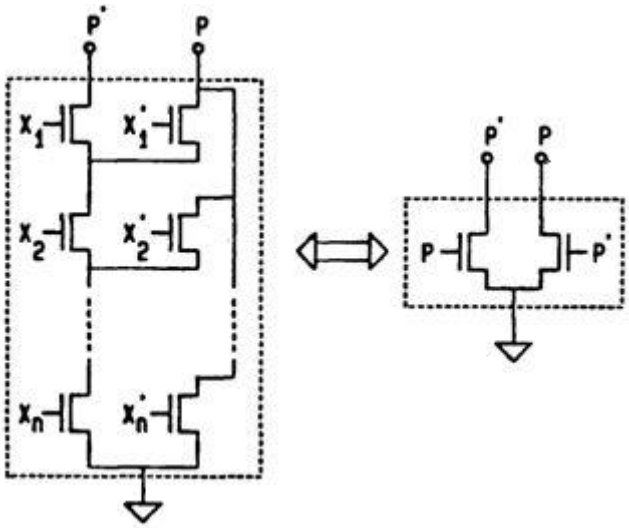

Figure 14(a) – The DCVS tree with symbolic representation for the function S= $a_1a_2 \ldots a_n$

Using the above figure as basic building block, we can construct numerous complex functions.

$$i_1 = R_1 y + R_2 y'$$
$$i_2 = R_1 + R$$

where the variable R's (considered instead of P) are two different product terms, and the variable y's are literals. The structures are shown in Figure 14(b) and Figure 14(c).

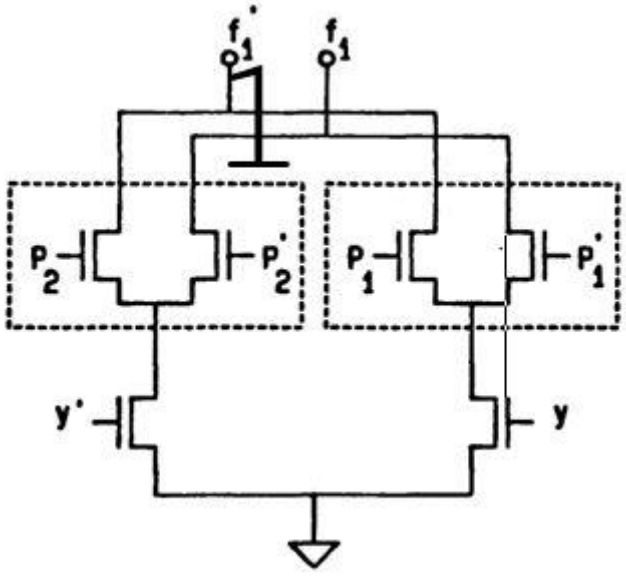

Figure 14(b) – For the function i_1

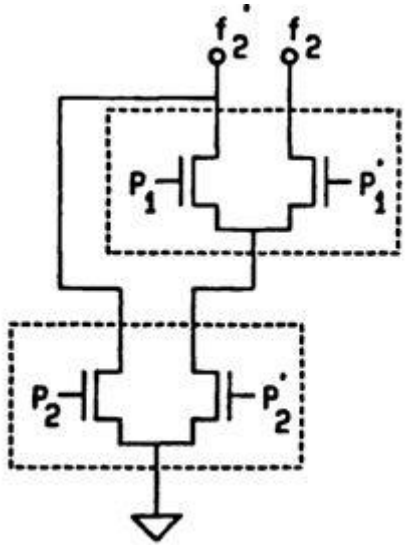

Figure 14(c) – For the function i_2

➢ **K-map Procedure** – The DCVS tree's input variable is represented by q_i, for i = 1,2,….,n, where there is a variable q_i and also the complement $q_i^{'}$. Also the set P is represented by a cube, where $q_i \in P$ represents $q_i \notin P$.

In a K-map which is having n number of variables, there are 2^n boxes and each which is having exactly n literals represents a cube. Boxes containing 0's are called 0-boxes (similarly, 1-boxes). A 0-loop encircling two adjacent 0-boxes containing a cube which is having literals but one less in number than each of the cubes signifying the original 0-box (same for 1-loop). If we consider two rectangular 0-loops, which is adjacent to K-map and is consisting 2^i 0-boxes. The 0-loops expresses cubes, say $Cx_k and Cx_k^{'}$, and we get a new rectangular 0-loop consisting 2^{i+1} 0-boxes with combination of two 0-loops and the new 0-loop which is expressing cube C (which is same for 1-loops).

Here is an example introducing the K-map algorithm, which exhibits certain ideas, given the Boolean function S (considered instead of Q) = $x_1x_2+x_2x_3+x_3x_1$ (representing the FA's carry function), constructing the Differential Cascode Voltage Switch Logic. Figure 15(a) shows the K-map.

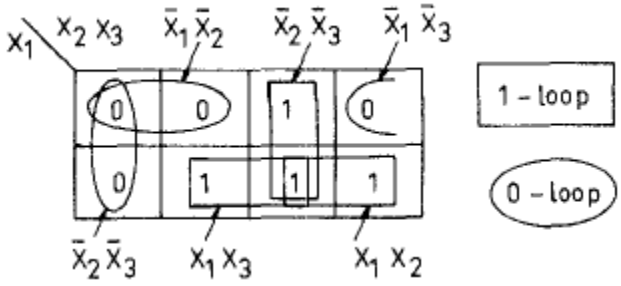

Figure 15(a) – K-map of a Full Adder showing the carry-out function

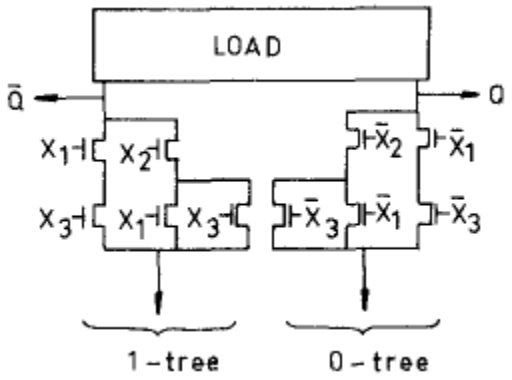

Figure 15(b) – DCVS implementation on Full Adder representing carry-out

The 0-loops and 1-loops are surrounded appropriately to represent the marginal cover for both the loops. The Figure 15(b) represents the subsequent DCVS tree pair. The tree which is committed to Node T (T instead of Q) is called the 0-tree and is derived from 0-cells. Similarly, for the node T' representing 1-cells and is derived from 1-cells. But both the trees are disjointed because both the cells are grouped separately. To realize the function S, this DCVS circuit requires 10 N-devices. Apart from constructing the two disjointed 0 and 1 trees, it does a lot more. Maximum commonality is allowed between the trees for its discovery. A 'shared' structure can be developed which leads to the minimization of device count.

Taken into consideration, if a 0-box (1-box) which is representing cube $x_1^{'}R$ and 1-box (0-box) representing x_1R exist concurrently, then the cell representing the cube R can be demarcated as a 10-box (01-box). These 01 or 10-boxes are treated as separate boxes of dissimilar forms. When two or more contiguous 01-cells (10-cells) are being encircled, then a 01-loop (10-loop) is formed.

Let's revisit the previous example, with these new concepts added. See the Figure 16(a), where three types of encirclements are there, as 0-loop, 1-loop and 10-loop.

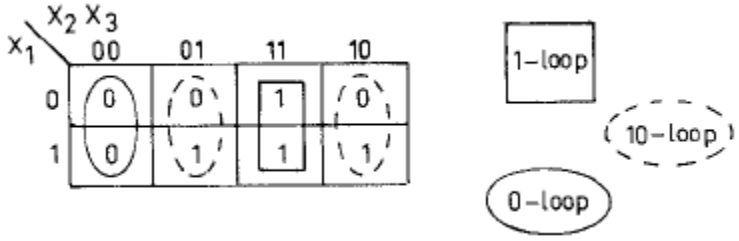

Figure 16(a) – K-map with different encirclements than the Figure of 15(a)

The ‘shared’ tree with 10-loops is first constructed in Figure 15(b).

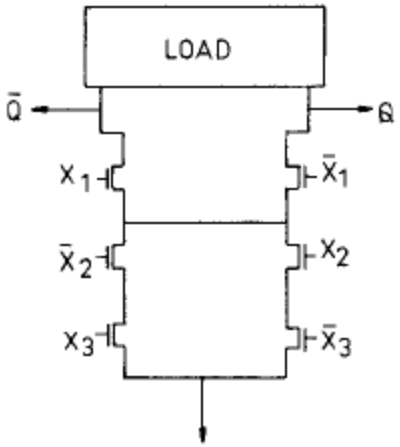

Figure 16(b) – DCVS tree with implementation on 10-loops

The complete DCVS tree with branches 0-loop and 1-loop are added and shown in Figure 16(c).

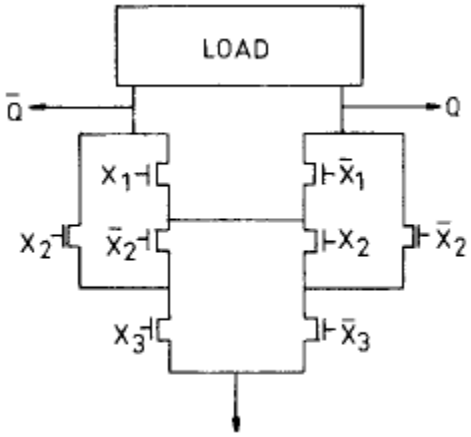

Figure 16(c) – Complete DCVS tree

Here only 8 N-devices are essential, and it is two devices less than disorganised tree which is shown in Figure 15(b). But the quantity of the levels which are stacked is amplified by 3.

The K-map procedure has 4 steps:

- Identify the 4 different types of cells, 0-boxes, 1-boxes, 01-boxes & 10-boxes.

- The marginal cover for all the 01-boxes are found out. Then the tree is constructed equivalent to marginal envelope. With magnitude i in ascending order, the variables x_i are arranged from top to bottom in each of the tree branches. The construction of tree branches is done according to the size of the loops, starting with small size loop. The control inputs which are of top pair are y_1' related with Node T’ and y_1

which is associated with Node T. Gate inputs y_1 and $y_1^{'}$ having the sources of the transistors are always connected together.

- A minimal cover is found out from the prime implicates of all the 10-cells such that the tree which is constructed may share some of the branches with the one in the second step. y_1 which is related with Node T' and $y_1^{'}$ which is related with Node T are the top pair of control inputs according to the second step.

- Then, a minimal cover is found out for the remaining 0-cells and 1-cells. The sharing of tree branches is always looked out for, while constructing the tree. The Node T (Node T') are connected to base of 0-tree (1-tree).

If $y_i^{'}s$ are rearranged (e.g. the variables y_1 and y_2 are inter-changed), the above technique might generate altered tree structures. Also to share tree branches and to choose minimal cover, there may be several ways to choose.

A 4-variable K-map is in Figure 17(a) and the first two steps are applied to generate the tree structure in Figure 17(b).

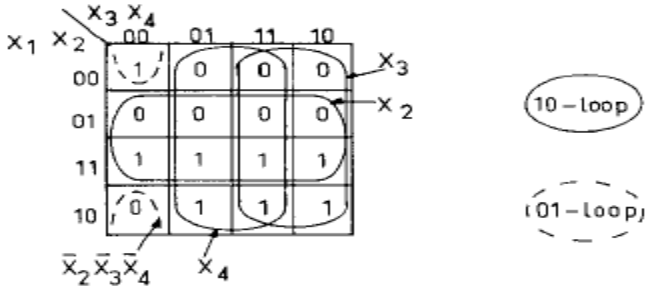

Figure 17(a) – K-map with function Q = $y_1^{'}y_2^{'}y_3^{'}y_4^{'} + y_1(y_2 + y_3 + y_4)$ showing 01 and 10-encirclements

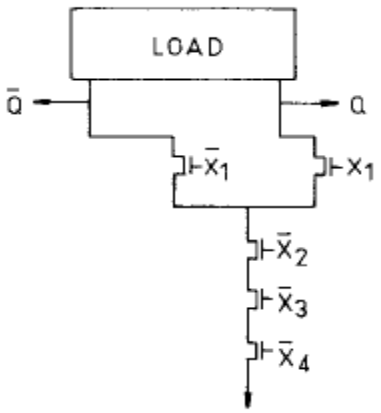

Figure 17(b) – DCVS circuit representing 01-loop

The complete DCVS tree is generated by applying step (3) and is shown in Figure 17(c).

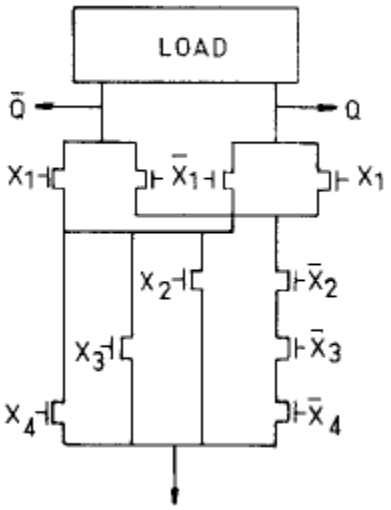

Figure 17(c) – Complete DCVS tree

There is no remaining 0-cells and 1-cells, therefore step (4) has been skipped.

Here is another example, where a changed method of encompassing K-map is in Figure 18(a), which points to an altered structure shown in Figure 18(b).

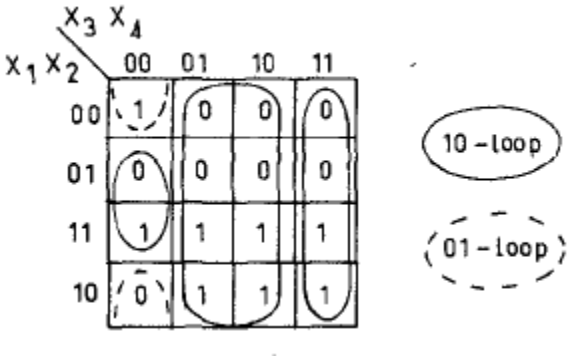

Figure 18(a) – Another example of K-map arrangement

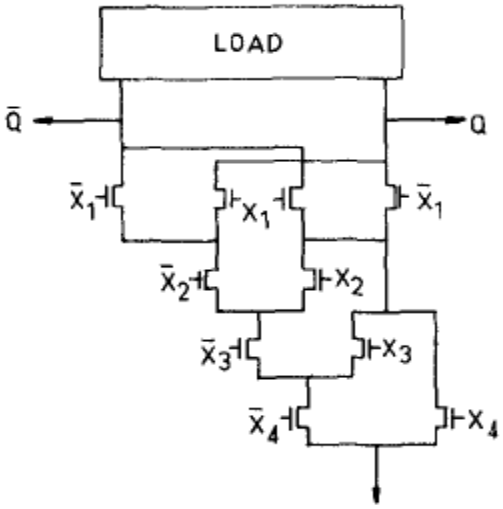

Figure 18(b) – Circuit resulting from the previous figure. Compare it with that of Figure 17(c)

In some tree branches, the levels which are stacked gets incremented since the 10-cells are not covered marginally in this demonstration. This adverse article, in combination with huge parasitic capacitances which is also linked with several shared source and drain connections, indicates 18(b) has inferior electrical performance than that of Figure 17(c).

If a DCVS is having levels which are stacked and is more than 6, then the presentation might be degraded, the reason might be that the parasitic source & drain capacitances keeps on charging & also along with it discharges too, having transistors with long chains. Therefore it is much better to divide DCVS circuits having 5 or less variables when the logic is complicated, for circuits where speed is required. In this case, the design procedure is very useful.

- **Tabular Method** – Quine-McCluskey Method is used in this tabular method where the prime implicants and the minimal covering set are found out [6]. A list is provided where it is consisting of two fields, viz. the input vector (y_1, …., y_n) {y taken instead of x} on the right and left is having decimal portrayal which is shown in Table 1.

Decimal representation of input vector	Input vector x_1 x_2 ····· x_n	
e.g. 1	0 0 1	} Record i
4	1 0 0	
3	0 1 1	} Record j>i
6	1 1 0	
. . . .	. . . .	

Table 1 – The list format for Tabular Method

The input vectors are arranged in a rising order of their index (number of 1's in binary representation) and are grouped into records. We start with a 0-list (list containing 0's) and a 1-list (list containing 1's) of the functions. Two other lists are also generated from the above two lists' namely, 10-list and 01-list. This procedure is same as the generation of 01-boxes and 10-boxes from the 0-boxes and 1-boxes, in the K-map method. The choice of marginal envelope from the 1-list, 0-list, 01-list and 10-list by the modified Quine-McCluskey method yields a DCVS structure which is in Figure 19.

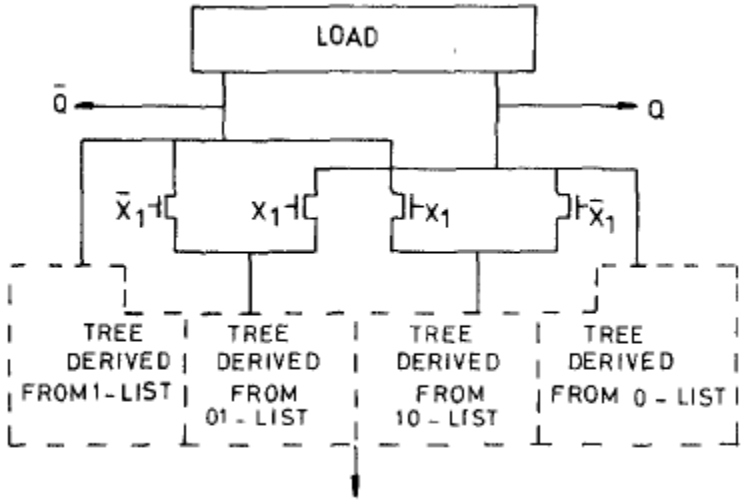

Figure 19 – The basic DCVS tree structure developed from Table 1

The Tabular method consists of five procedures –

- The 1-list is drawn with vectors which are true (y_1, …., y_n) of f(Q). Then the list is split into accounts containing growing i from topmost to lowermost. Thereafter, the 0-list is drawn containing vectors which are false of f(Q) and also increasing i from top to bottom.

- For i which is from 1 to n;

In the 1-list, rows which are beginning with y_1 =1 within account i is matched with rows which is within record i-1 of the 0-list. Considering the degraded vector (y_2, …., y_n) of two rows are same, then the 1-list and the 0-list are checked correspondingly & a new row is added to 10-list. The arrangement of 10-list is a bit altered, which is in Table 2. The variable x_1 is not required anymore.

Decimal representation of reduced input vector	Reduced input vector	
	x_2 x_3 ..... x_n	
e.g.		
1	0 1	} Record i
2	1 0	
3	1 1	} Record j>i
.	.	
.	.	
.	.	
.	.	

Table 2 – Format of 10-list

- For i = 0 to n-1;

For the 1-list, the rows which is starting with y_1 =0 within account i is matched with the rows of the 0-list which is starting with i+1. Here also, considering the degraded vector (y_2, ….,y_n) of the two rows are same,

then the two rows in the 0-list and 1-list are checked correspondingly & a new row is added to 01-list. Both the formats of 01-list and 10-list are same.

- The Quine-McCluskey Method is applied to the rows in the 01-list and 10-list for finding the prime implicants. The marginal cover set is selected for each of the two lists by row and column supremacy measures, and while constructing the corresponding trees the maximum amount of sharing of tree branches is looked out for. Thus, a 'shared' is built.

- For selecting a minimal cover set, a conventional procedure is applied for abandoned rows in the 0-list and 1-list. Then the trees are constructed according to these two marginal sums, by addition of more branches to the tree which is 'shared'. By this way, a DCVS structure is constructed which is shown in Figure 19.

3-Bit Magnitude Comparator designed is considered by Tabular Method. The circuit equates between binary numbers, $A=A_3A_2A_1$ & $B=B_3B_2B_1$, and which also gives an output Q=1, whenever A>B. The variables (y_1, y_2, y_3, y_4, y_5, y_6) are equal to (A_3, B_3, A_2, B_2, A_1, B_1). A dissimilar duty will lead to altered structure.

From the initial step, we tabulate the 0-list (36 rows) and 1-list (28 rows). Then when the second stage is completed, a 10-list is drawn as shown in Table 3.

Decimal representation of reduced input vector	Reduced Input vector				
	x_2	x_3	x_4	x_5	x_6
0	0	0	0	0	0
4	0	0	1	0	0
1	0	0	0	0	1
5	0	0	1	0	1
6	0	0	1	1	0
12	0	1	1	0	0
3	0	0	0	1	1
24	1	1	0	0	0
18	1	0	0	1	0
7	0	0	1	1	1
13	0	1	1	0	1
25	1	1	0	0	1
26	1	1	0	1	0
15	0	1	1	1	1
27	1	1	0	1	1
30	1	1	1	1	0

Table 3 – The 10-list of a 3-bit Magnitude Comparator

The third step doesn't generate any 01-list. From the fourth step, a prime implicant table is obtained from the 10-list which is shown in Table 4 and these prime implicants actually form a marginal envelope.

Decimal representation / prime implicants	0	1	3	4	5	6	7	12	13	15	18	24	25	26	27	30
$x_2\ \overline{x_4}\ x_5\ \overline{x_6}$ / $x_2\ x_3\ x_5\ \overline{x_6}$											x			x x		x
$\overline{x_2}\ \overline{x_3}\ \overline{x_5}$	x	x		x	x											
$\overline{x_2}\ \overline{x_3}\ x_4$				x	x	x	x									
$\overline{x_2}\ x_4\ \overline{x_5}$				x	x			x	x							
$\overline{x_2}\ \overline{x_3}\ x_6$		x	x		x		x									
$\overline{x_2}\ x_4\ x_6$					x		x		x	x						
$x_2\ x_3\ \overline{x_4}$												x	x	x	x	

Table 4 – Prime implicant table of the 10-list

The 'shared' tree is shown in Figure 20.

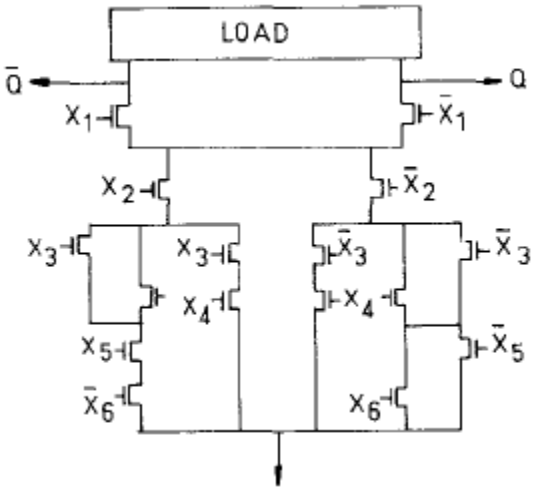

Figure 20 – The tree circuit with the 10-list

From the fifth step, the abandoned rows of the 0-list and 1-list results in Table 5 and Table 6.

Decimal representation of input vector	Input vector					
	x_1	x_2	x_3	x_4	x_5	x_6
1	0	0	0	0	1	0
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	0	1	0	1	0
40	1	0	1	0	0	0
34	1	0	0	0	1	0
11	0	0	1	0	1	1
14	0	0	1	1	1	0
41	1	0	1	0	0	1
42	1	0	1	0	1	0
43	1	0	1	0	1	1
46	1	0	1	1	1	0

minimal sum = $\overline{x_2}\ x_3\ x_5\ \overline{x_6} + \overline{x_2}\ \overline{x_4}\ x_5\ \overline{x_6} + \overline{x_2}\ x_3\ \overline{x_4}$

Table 5 – The 1-list and its minimal sum for the Magnitude Comparator

Decimal representation of input vector	Input vector					
	x_1	x_2	x_3	x_4	x_5	x_6
16	0	1	0	0	0	0
20	0	1	0	1	0	0
17	0	1	0	0	0	1
48	1	1	0	0	0	0
21	0	1	0	1	0	1
22	0	1	0	1	1	0
28	0	1	1	1	0	0
19	0	1	0	0	1	1
52	1	1	0	1	0	0
49	1	1	0	0	0	1
29	0	1	1	1	0	1
23	0	1	0	1	1	1
53	1	1	0	1	0	1
54	1	1	0	1	1	0
60	1	1	1	1	0	0
51	1	1	0	0	1	1
31	0	1	1	1	1	1
61	1	1	1	1	0	1
55	1	1	0	1	1	1
63	1	1	1	1	1	1

minimal sum = $x_2 \overline{x_3}\ \overline{x_5} + x_2\ \overline{x_3}\ x_6 + x_2\ x_4\ x_6 + x_2\ x_4\ \overline{x_5} + x_2\ \overline{x_3}\ x_4$

Table 6 – The 0-list and its minimal sum for the Magnitude Comparator

The equivalent marginal sums done by the Quine-McCluskey Method are specified and the complete DCVS is shown in Figure 21.

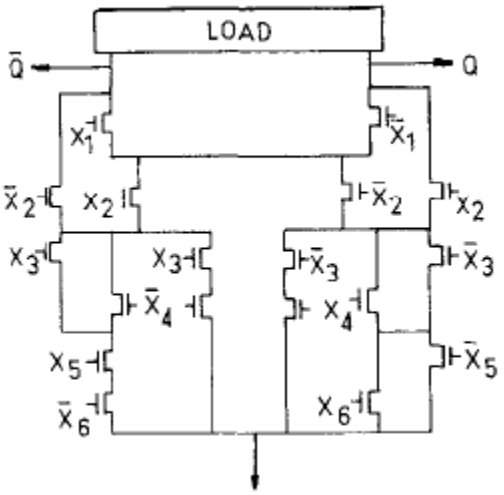

Figure 21 – 3-bit Magnitude Comparator.

Chapter 4

TESTING SCHEMES on **DCVSL STRUCTURES**

There is a concept in the most structured design practices that with some additional circuitry, all the memory elements in an IC can be threaded together into a shift register. A control switch is able to switch the memory elements to shift register mode from their normal modes of operation and that is when the current state of IC is frozen and shifted out for examination. Level-Sensitive Scan Design (LSSD) is IBM's discipline for structured design for testability. Here, the term 'Scan' refers to the ability for any network to shift into or out of any state and the term 'Level-Sensitive' indicates to the logic depth, circuit excitation and handling of clocked circuitry. The key element in the design is the 'shift register latch' (SRL), which is being implemented in DCVS Logic [26].

The self-testing property of DCVS trees is described through a different scheme [2]. Due to the presence of complementary outputs for every tree, the DCVS circuit has unique property of online testability. This provides us with stuck-at and dynamic fault coverage (e.g., due to alpha particles or power glitches).

Differential paths from two nodes (Q and Qbar) are produced by a DCVS tree, which is represented by an ordered pair (Q,Q') to ground. According to fault-free operation, only one of the two paths to ground is active, which produces a legal (code space) output (1,0) or (0,1). For a faulty operation, the output changes state from one legal state to an illegal (non-code space) state such as (0,0) or (1,1) [2]. Thus, the presence of fault in the tree is clearly indicated by the detection of an illegal state at the output of any tree. This is called the self-testing property of DCVS trees.

A DCVS tree also has another property called the fault-secure property, which describes that any single illegal input to a functioning tree causes the tree to produce either an illegal output or correct output [2]. In spite of the DCVS tree having one of its inputs in an illegal state, it can produce correct legal output provided if its output is independent of other illegal input. An example of 3-input NAND gate is shown in Figure 22.

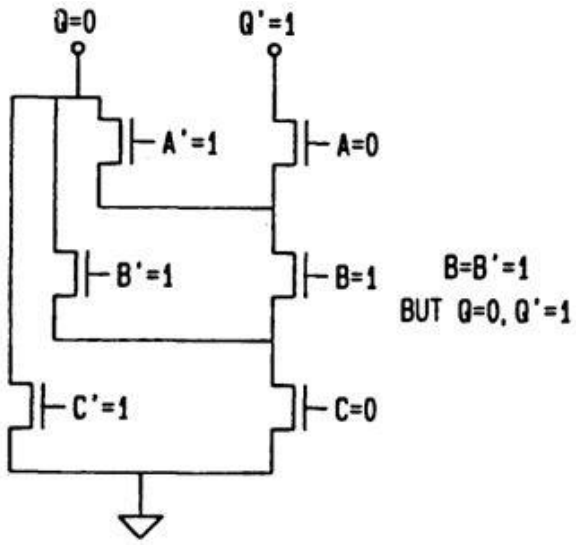

Figure 22 – 3-input NAND gate with illegal inputs

To increase the error observability, we can place the illegal state detector at the outputs of internal DCVS trees rather than placing them only at the latches of logical block boundaries. Thus the illegal state detector can be a XOR circuit shown in Figure 23. Whenever (Q,Q') is equal to (0,0) or (1,1), the 'error flag' is pulled down.

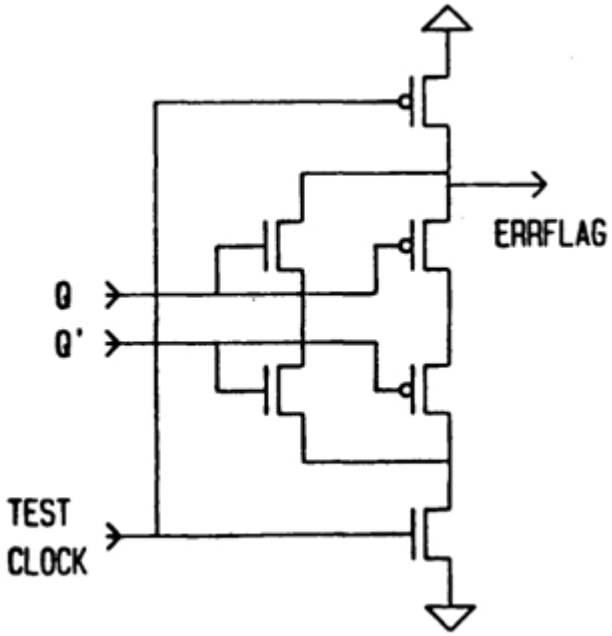

Figure 23 – Illegal state detector for DCVS trees

Chapter 5

Performance Analysis of **DCVSL Structures**

A comparison of all the three DCVSL structures are shown in this following section, with parameters such as Temperature, Voltage, Power Consumption, Delay and Power Delay Product. All the analysis shows that the proposed structures shows better result in terms of power consumption than the conventional ones. But the delay is comparatively a bit more for the proposed structure than the conventional one. So, for scheming a 'faster' digital gate, one generally requires a better power plus for designing a gate to lower power consumption, the digital device is slowed down. Therefore, both way there is a hindrance and so propagation delay and power dissipation generally form a design trade off, i.e. you improve one and the other is degraded. To quantify how effective, or efficient a digital design is in terms of delay and power, a product of both propagation delay and power dissipation is used. Hence, the term PDP. The DCVSL structures are based on the 2:1 Multiplexer. Here is the Truth Table of 2:1 Multiplexer shown in Table 7.

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table 7 – Truth Table of 2:1 Multiplexer

In all the DCVSL Structures, there are three inputs, namely (a, b, s). 'a' and 'b' are the two inputs of a multiplexer, and 's' is the select line. 'z' and 'z1' are the outputs, where 'z1' is the complementary output of z. Similarly, 'a1', 'b1' and 's1' are the complementary inputs which is provided by the inverter. Vpulse from Cadence Virtuoso is used to supply voltage to the three inputs, where V1 is 0V and V2 is 1V. The supply voltage is varied from 1V to 1.4V for all the DCVSL structures on the basis of which analysis is done, starting from Static DCVSL, then Dynamic DCVSL and ending with Modified DCVSL.

For the input 'a', the 'Period' is kept at 44ns and the 'Pulse Width' is kept at 22ns.
For the input 'b', the 'Period' is kept at 23ns and the 'Pulse Width' is kept at 11ns.
For the input 's', the 'Period' is kept at 10ns and the 'Pulse Width' is kept at 5ns.

➢ (5.1) Static DCVSL –

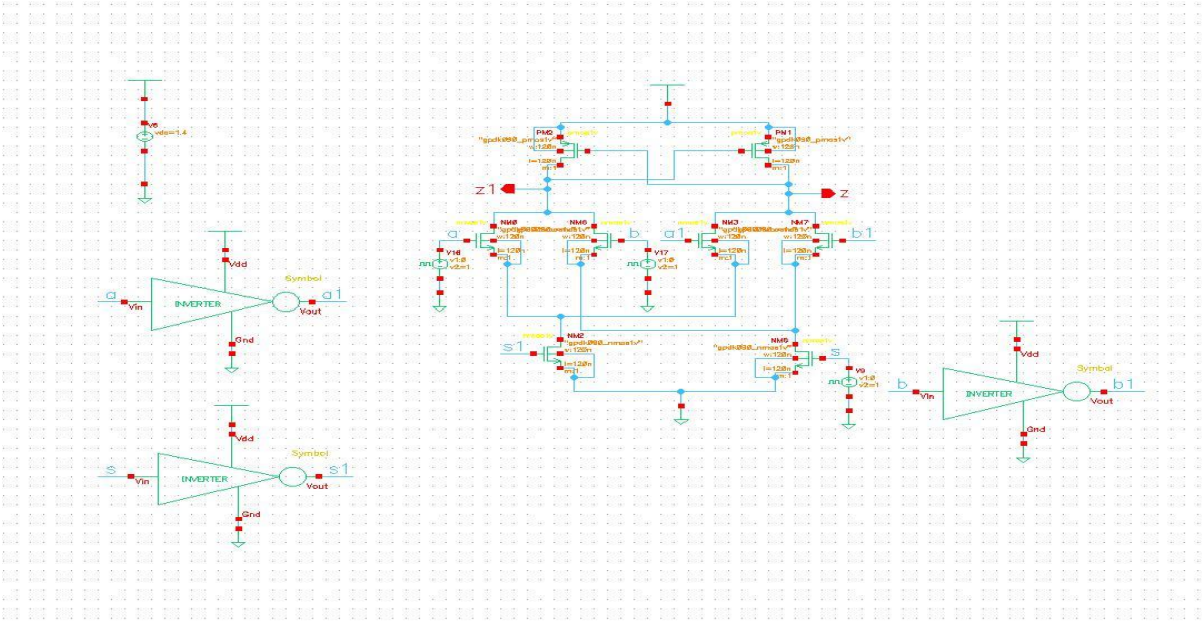

Figure 24(a) – Conventional Static DCVSL

There are 2 PMOS and 6 NMOS in the main DCVSL Structure. Considering the three inverters, the PMOS count goes up to 5 PMOS and 9 NMOS. Here, the W/L ratio of each transistor is kept at 120nm/120nm, i.e. 1:1. Whereas, the transistors within the inverter is kept at – for PMOS, it is 240nm/120nm, i.e. 2:1 and for NMOS, it is 120nm/120nm, i.e. 1:1. Moreover, in conventional structure, for the DCVSL portion; out of the 6 NMOS, the above 4 NMOS's body are connected to the source of the NMOS but the source of the NMOS is not connected to the ground and is internally connected to other NMOS's source. The rest 2 NMOS's body are connected to the source, which is ultimately connected to the ground. For PMOS, the body and the source are connected to V_{DD}.

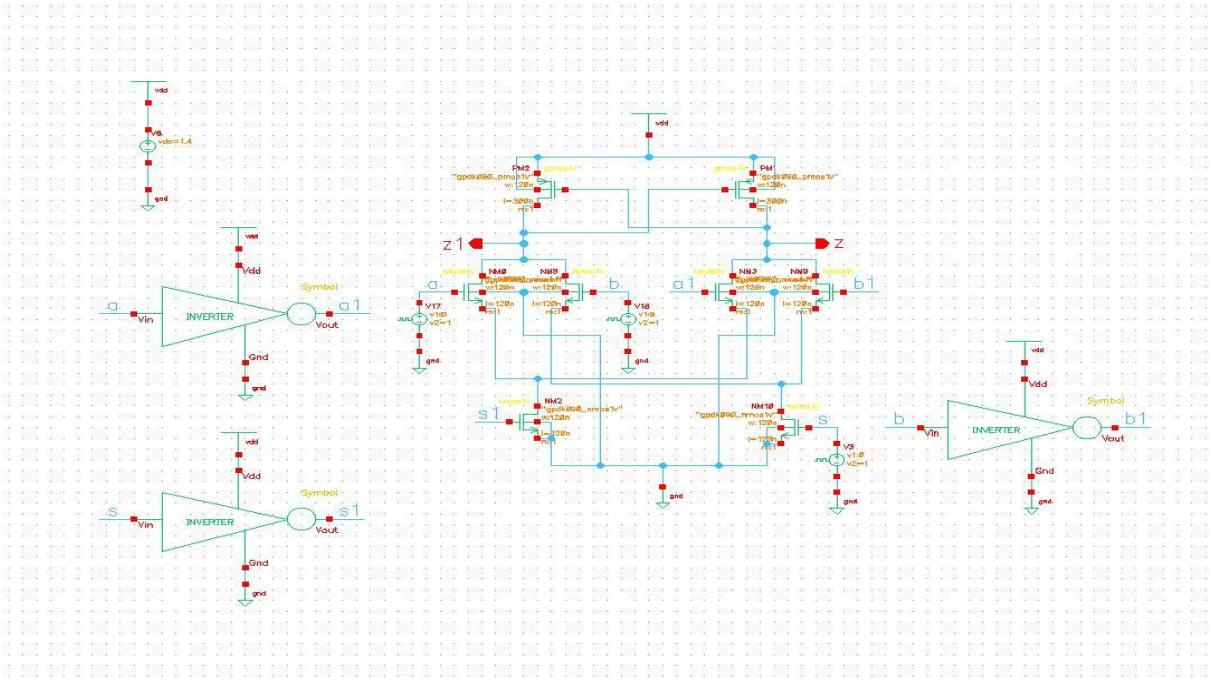

Figure 24(b) – Proposed Static DCVSL

In the proposed design, the PMOS's (W/L) ratio is kept at 120nm/300nm, i.e. 2.5:1 and NMOS's (W/L) ratio is kept at 120nm/120nm, i.e. 1:1. Here, the above 4 NMOS's body are connected to the ground and the rest 2 NMOS's body are connected to the source, which is ultimately connected to the ground. For PMOS, the body's and source's connection is done to the $V_{DD.}$

The transient responses for both the conventional and proposed circuits are shown below, with varying Voltages from 1V to 1.4V.

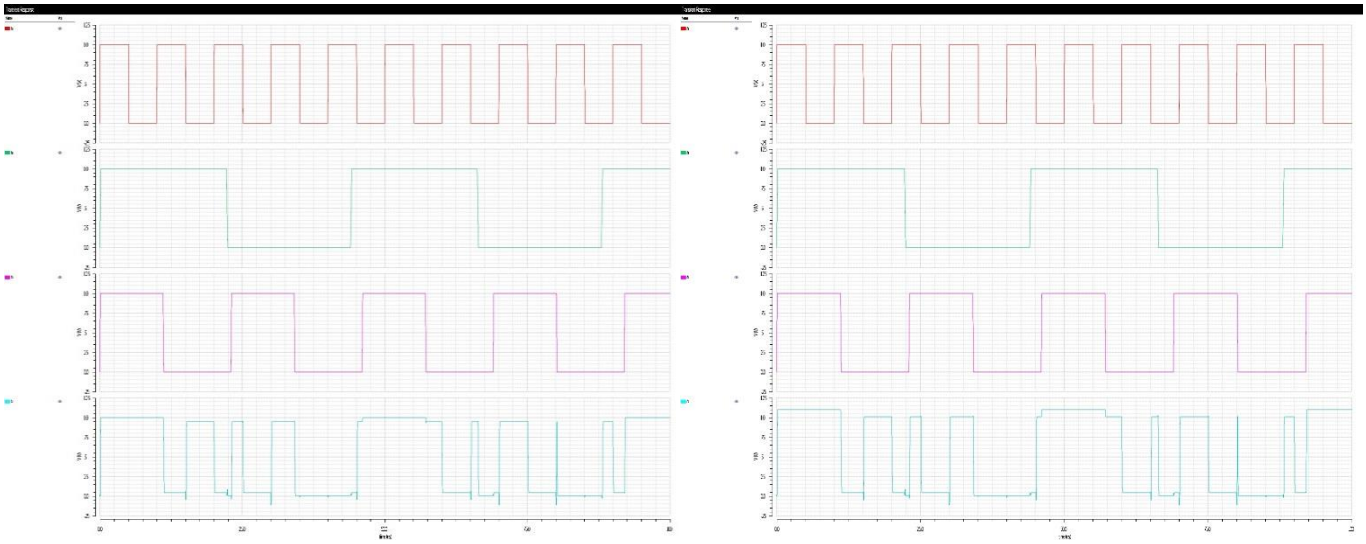

Figure 25(a) and 25(b) – Conventional (1V) and (1.1V)

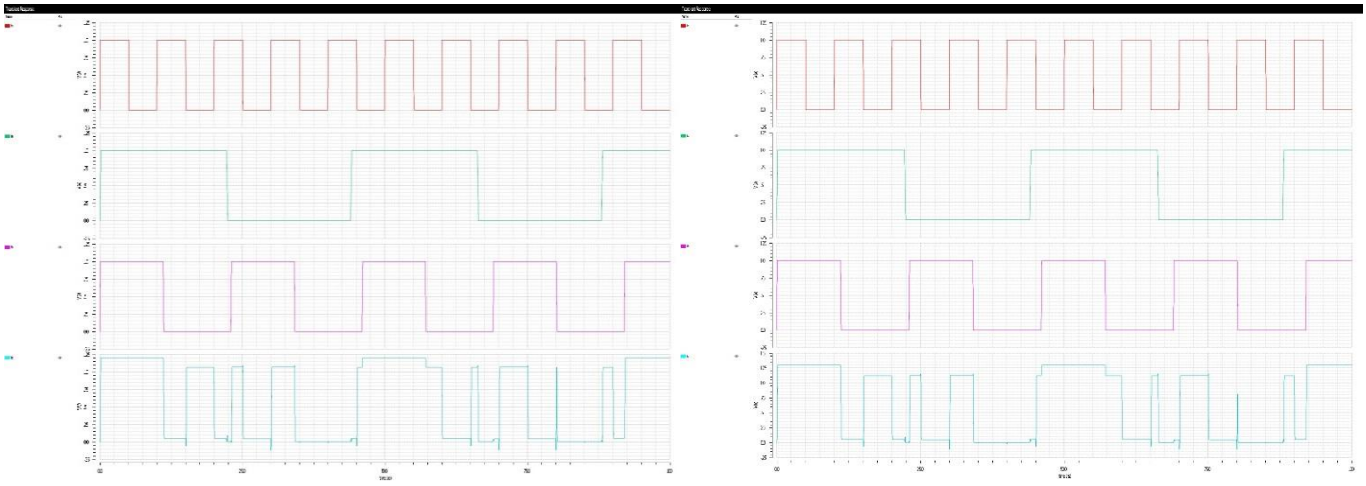

Figure 25(c) and 25(d) – Conventional (1.2V) and (1.3V)

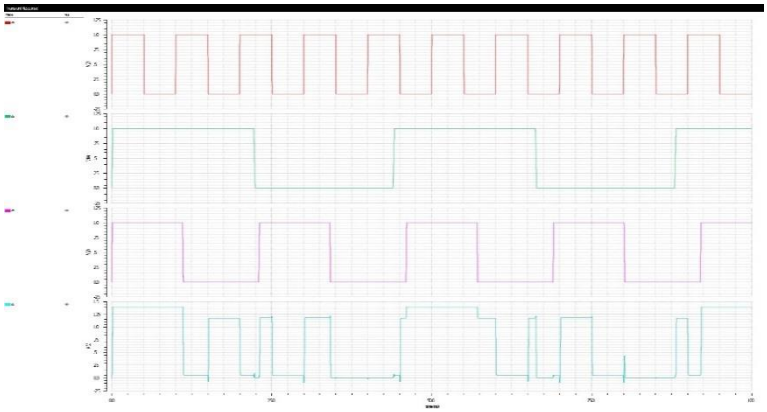

Figure 25(e) – Conventional (1.4V)

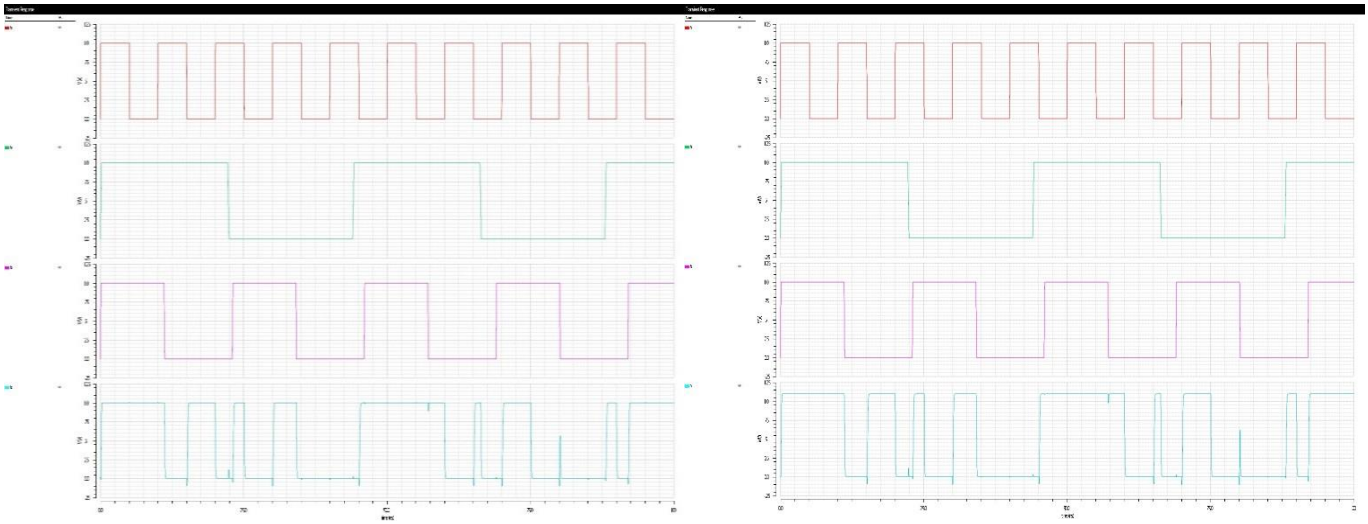

Figure 26(a) and 26(b) – Proposed (1V) and (1.1V)

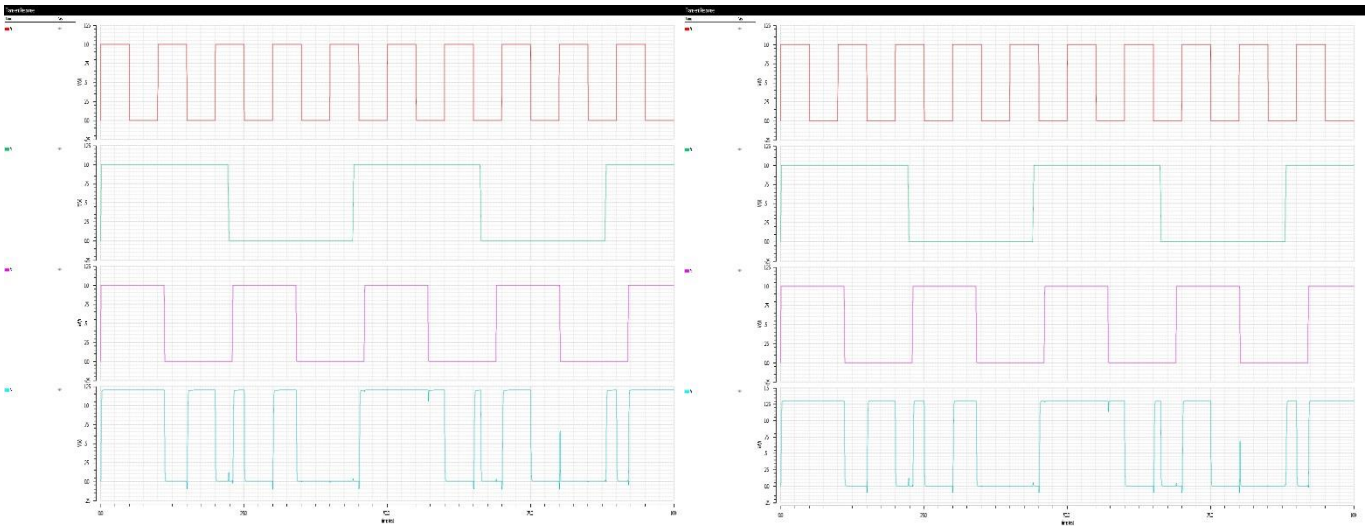

Figure 26(c) and 26(d) – Proposed (1.2V) and (1.3V)

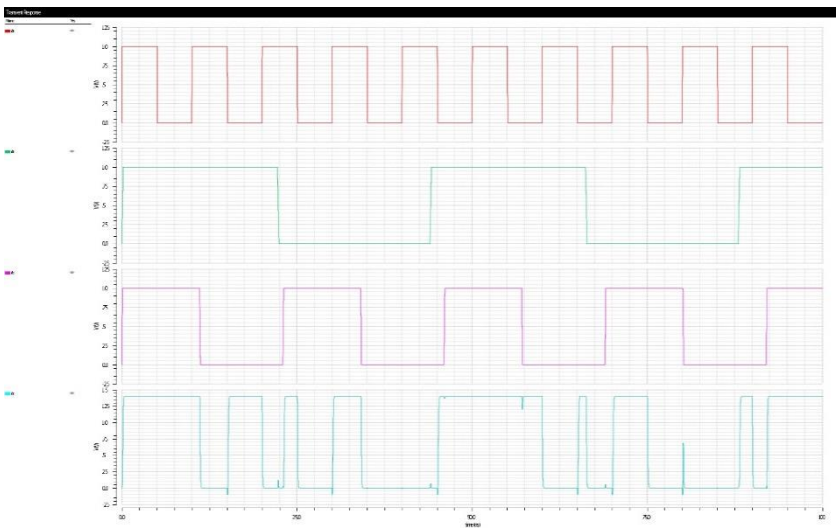

Figure 26(e) – Proposed (1.4V)

Between the conventional and proposed Static DCVSL circuits, a comparison is done between the power consumption and temperature, varying the Voltage from 1V to 1.4V. The comparison shows better results for the proposed DCVSL circuit, in terms of power consumption.

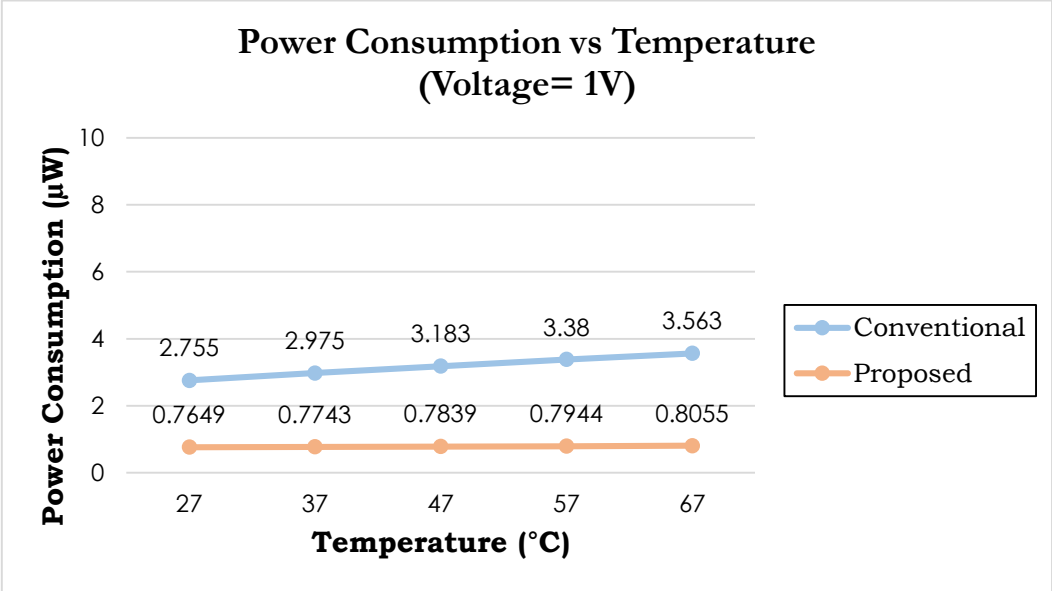

Figure 27 (a) – Power Consumption vs Temperature (1V)

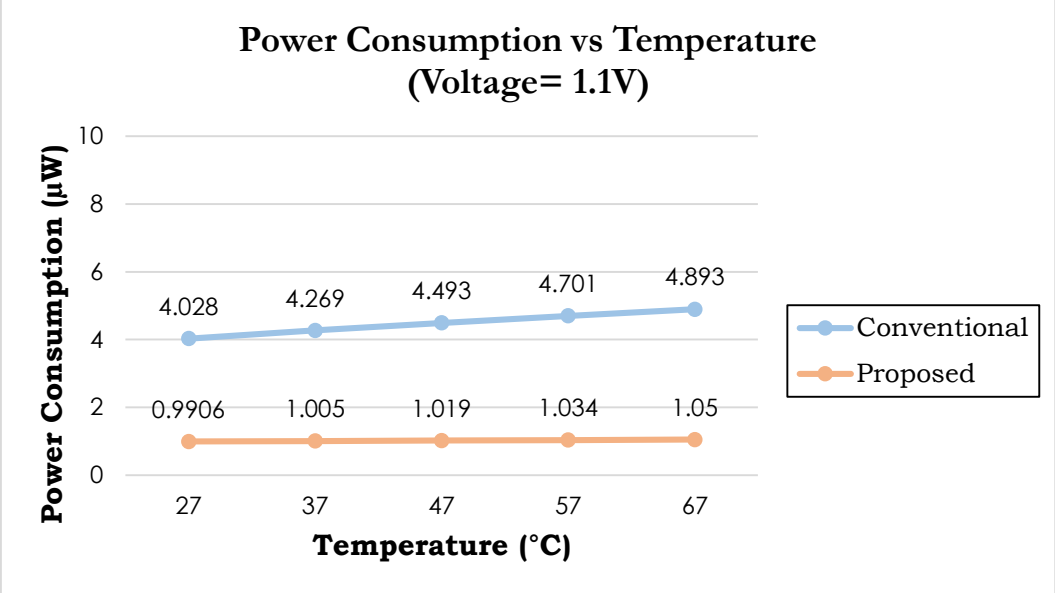

Figure 27(b) – Power Consumption vs Temperature (1.1V)

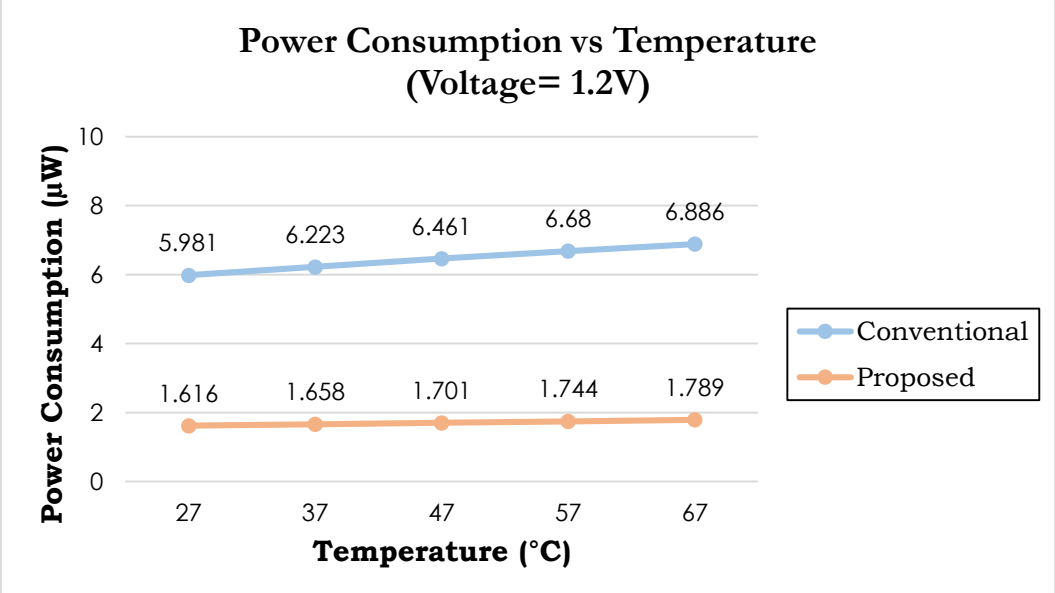

Figure 27(c) – Power Consumption vs Temperature (1.2V)

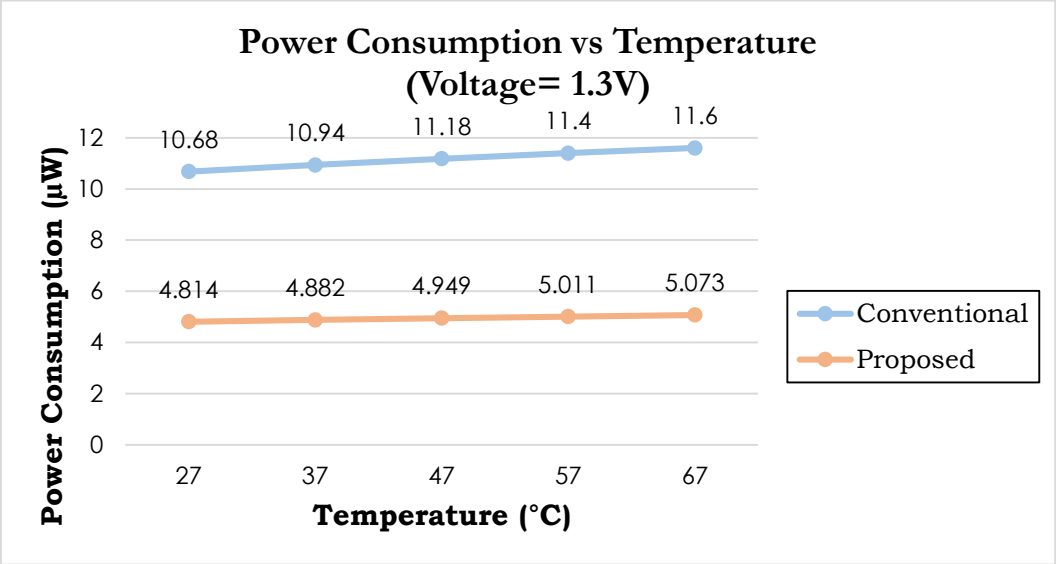

Figure 27(d) – Power Consumption vs Temperature (1.3V)

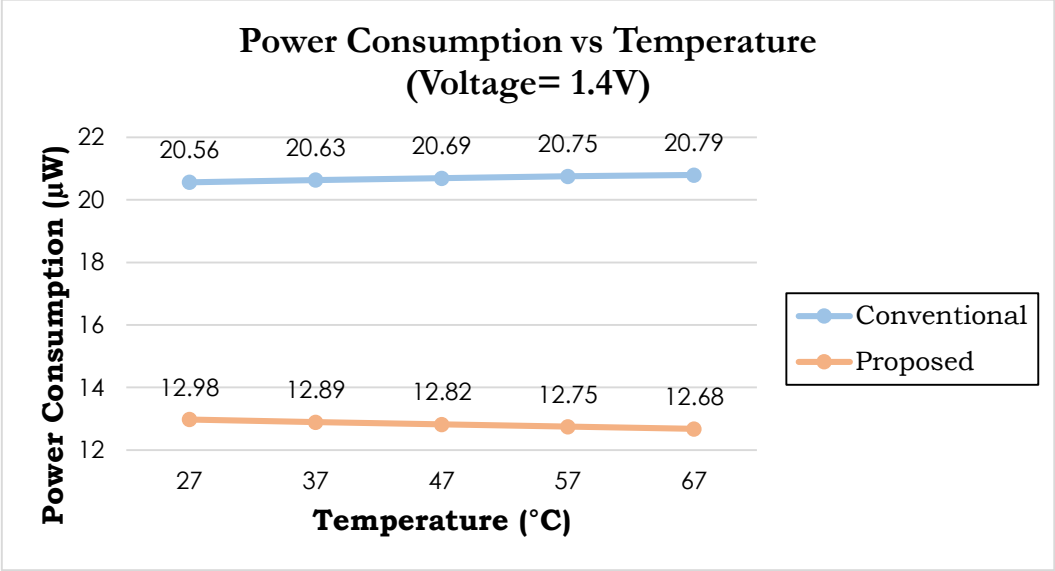

Figure 27(e) – Power Consumption vs Temperature (1.4V)

Now, we compare between the delay and temperature.

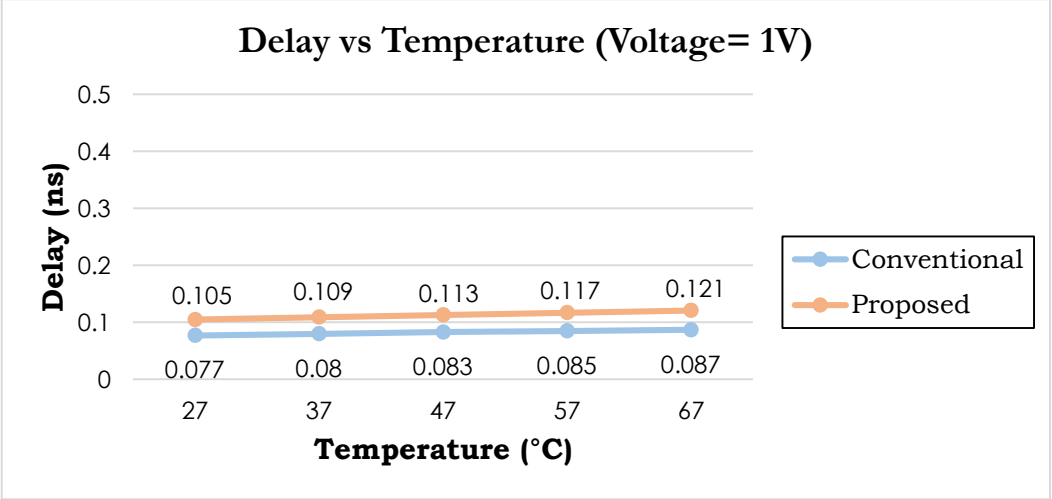

Figure 28(a) – Delay vs Temperature (1V)

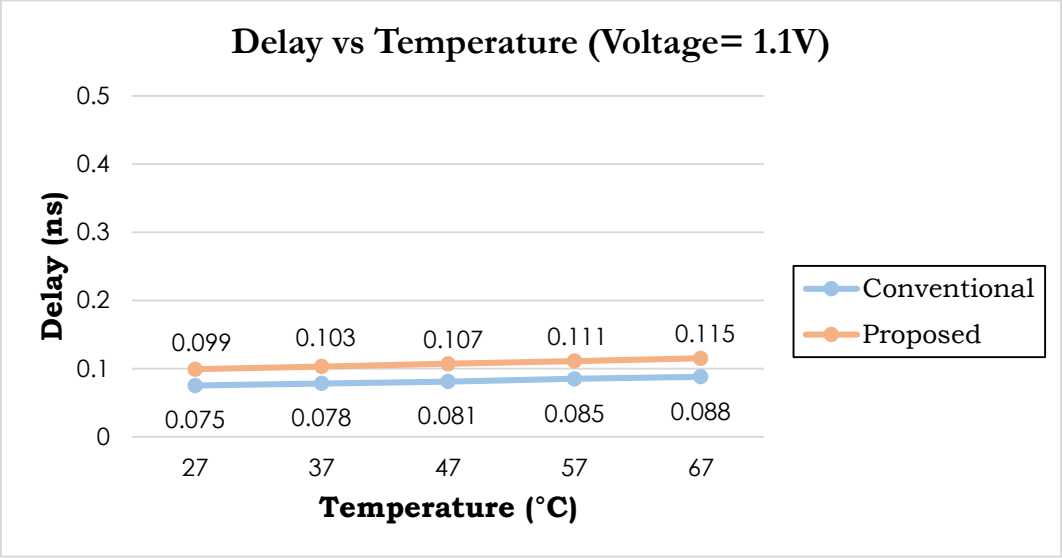

Figure 28(b) – Delay vs Temperature (1.1V)

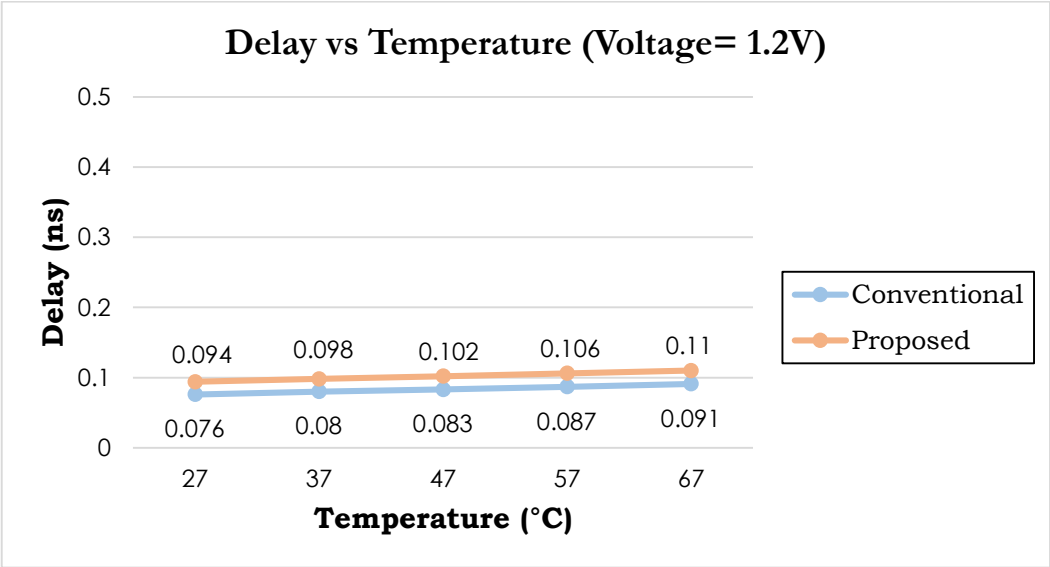

Figure 28(c) – Delay vs Temperature (1.2V)

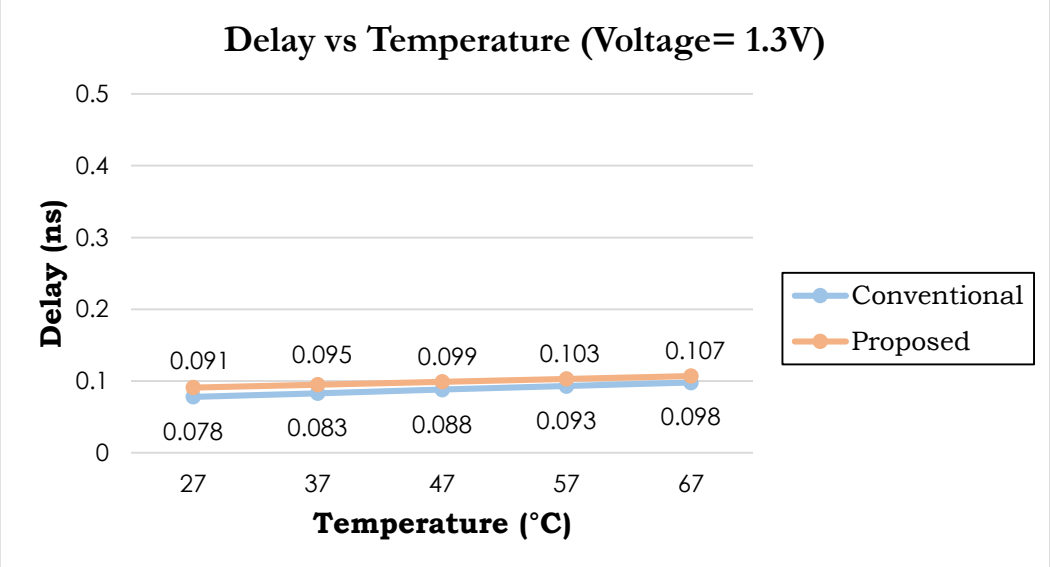

Figure 28 (d) – Delay vs Temperature (1.3V)

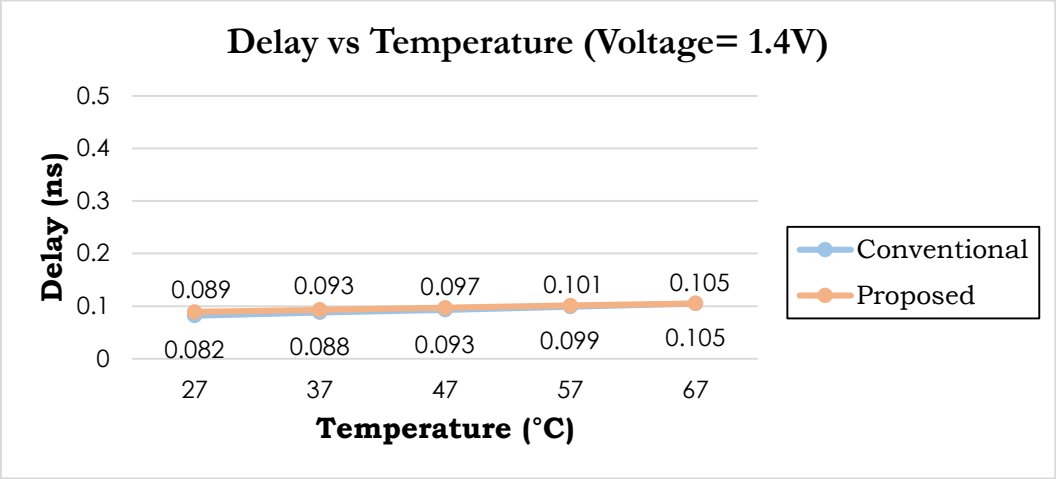

Figure 28(e) – Delay vs Temperature (1.4V)

Here, the delay is a bit more for the proposed DCVSL circuit.

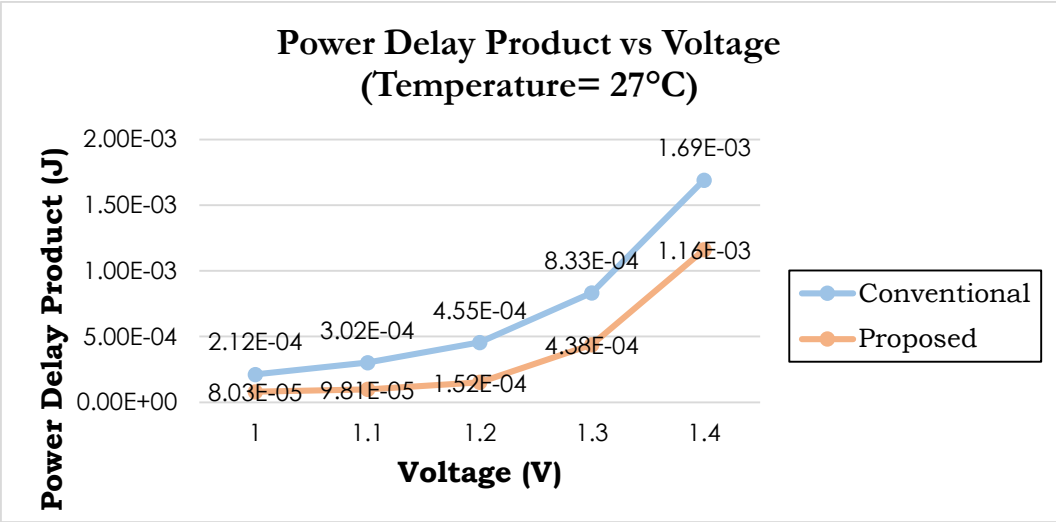

Figure 29 – Power Delay Product vs Voltage

➢ (5.2) Dynamic DCVSL –

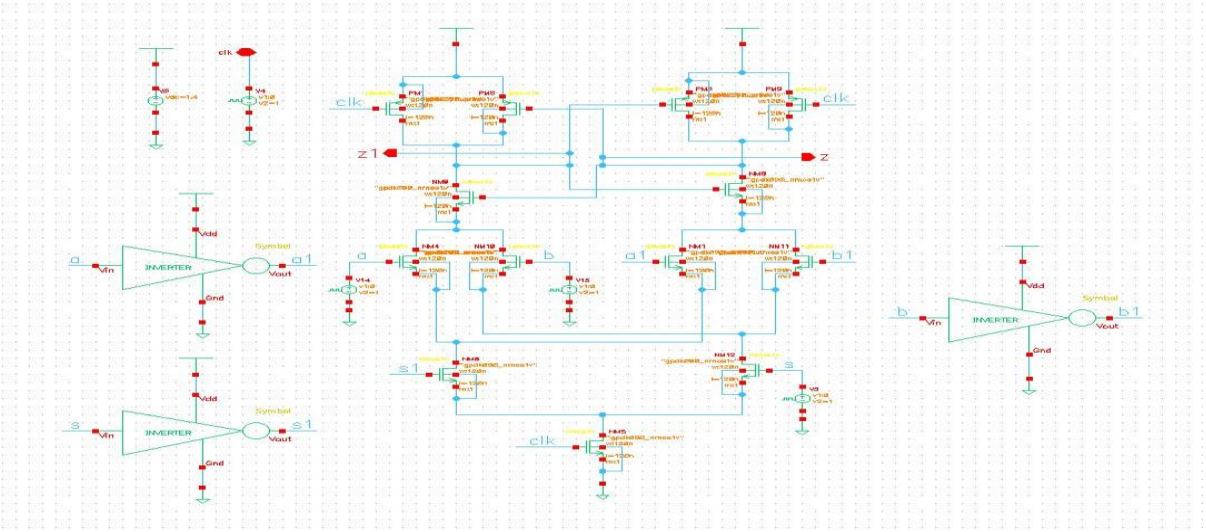

Figure 30(a) – Conventional Dynamic DCVSL

There are 4 PMOS and 9 NMOS in the DCVSL structure. Apart from that, there are 3 inverters, which constitutes 1 PMOS and 1 NMOS each, which accounts to 3 PMOS and 3 NMOS in total. The PMOS is having (W/L) = 240nm/120nm, i.e. 2:1 in ratio, whereas the NMOS is having (W/L) = 120nm/120nm, i.e. 1:1 in ratio. In Dynamic DCVSL, a clock is also used which is the input of the gate to 2 PMOS and 1 NMOS. The clock is having a delay of 10ns. Here, out of the 4 PMOS, 2 extreme PMOS having clock as the input to the gate is also having the bodies of both of them connected to the V_{DD}. The other two PMOS having their bodies connected to the drain of it and also internally connected to the gate of the intermediate NMOS and to the gate of the other two PMOS. Two intermediate NMOS are used here, which is having their body connected to the source of it, but the source is not connected to the ground but to the drains' of the other NMOS, two for each. Except the NMOS having clock as the input, the rest of the NMOS are having their body connected to the source which are not connected to the ground.

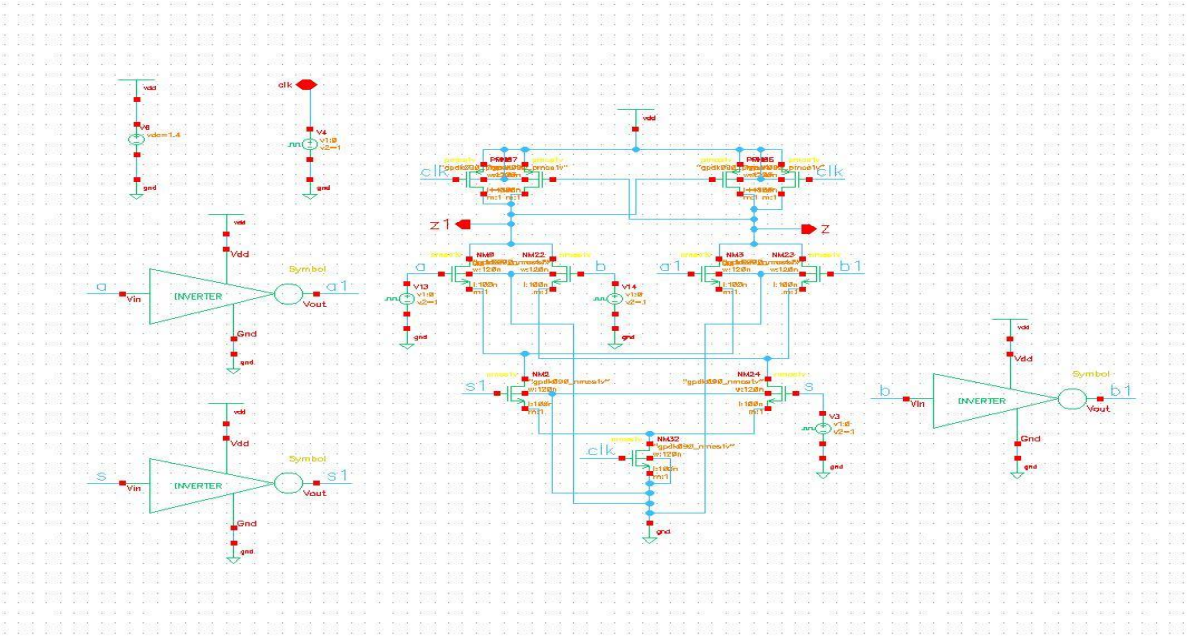

Figure 30(b) – Proposed Dynamic DCVSL

Here, the body of all the PMOS of the DCVSL structure are connected to the source which is ultimately connected to the V_{DD}. Same way, the body of all the NMOS of the DCVSL structure are connected to the source which in turn is connected to the ground. The (W/L) ratio of all the PMOS are 120nm/180nm, i.e. 1/1.5 whereas the (W/L) ratio for all the NMOS are 120nm/100nm, i.e. 1.2/1.

The transient responses for both the conventional and proposed circuits are shown below, with varying Voltages from 1V to 1.4V.

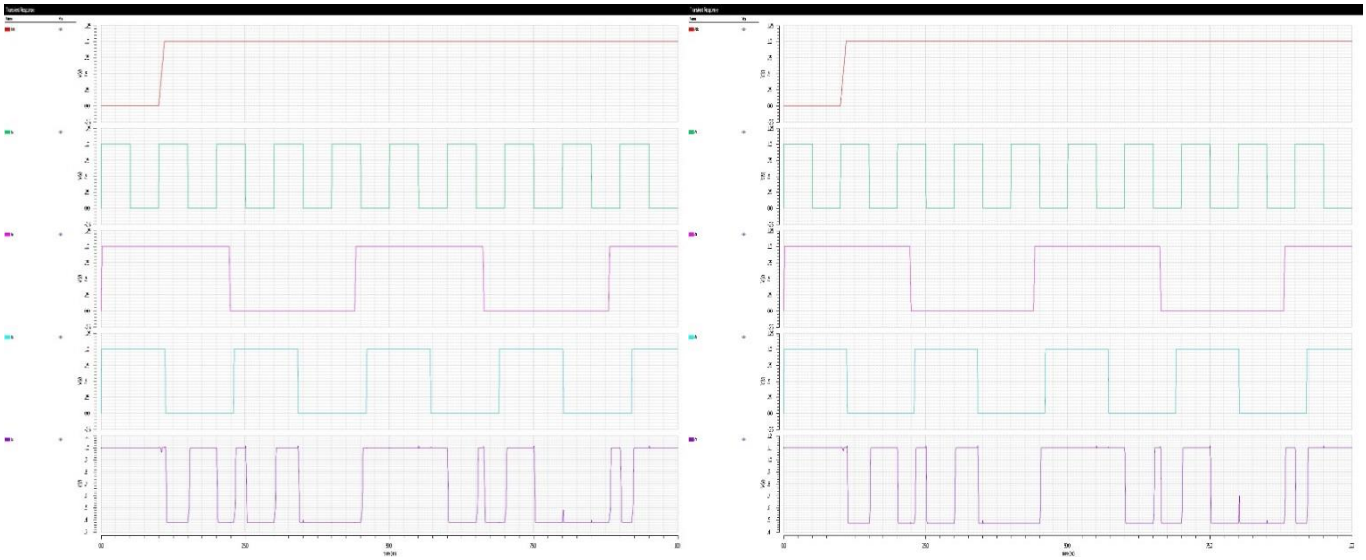

Figure 31(a) and 31(b) – Conventional (1V) and (1.1V)

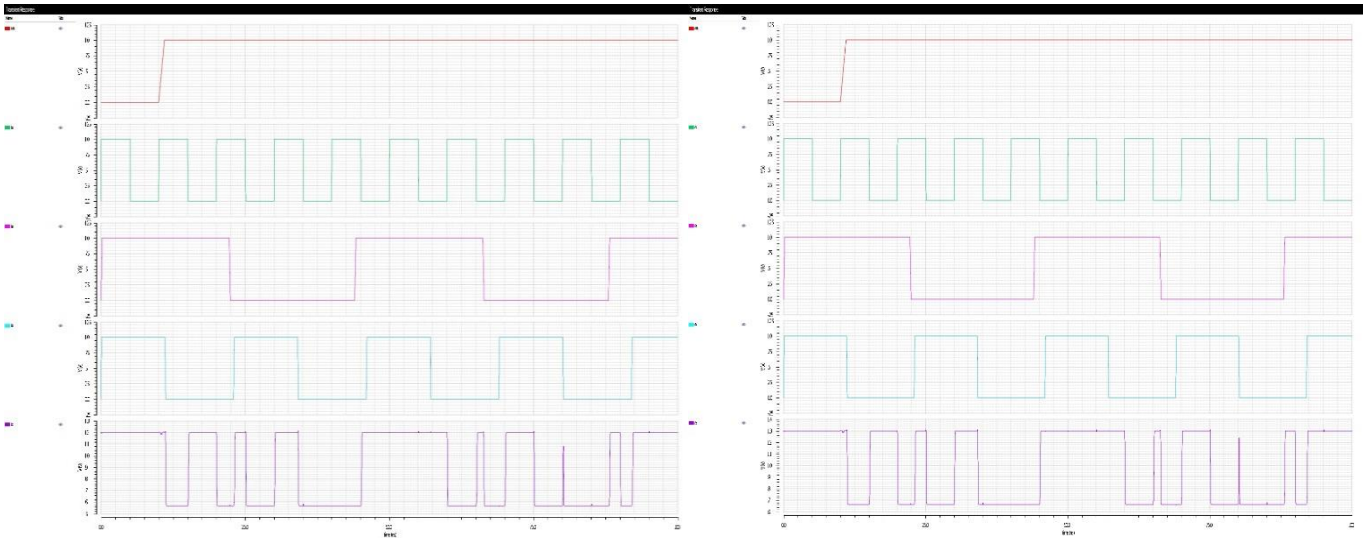

Figure 30(c) and 30(d) – Conventional (1.2V) and (1.3V)

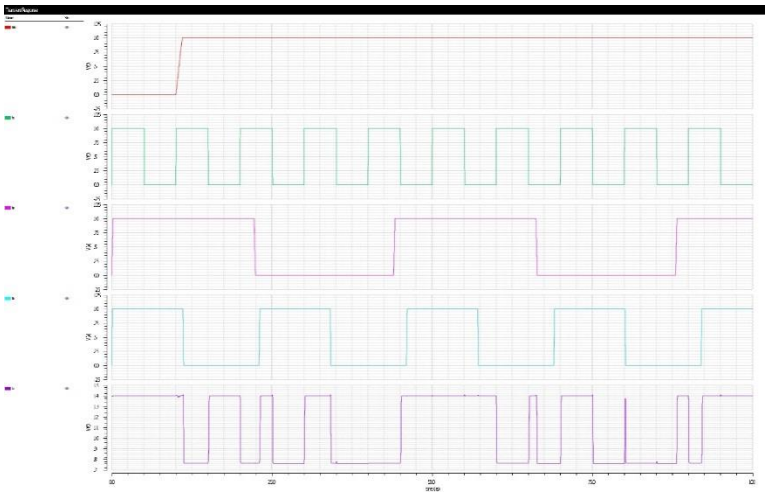

Figure 30(e) – Conventional (1.4V)

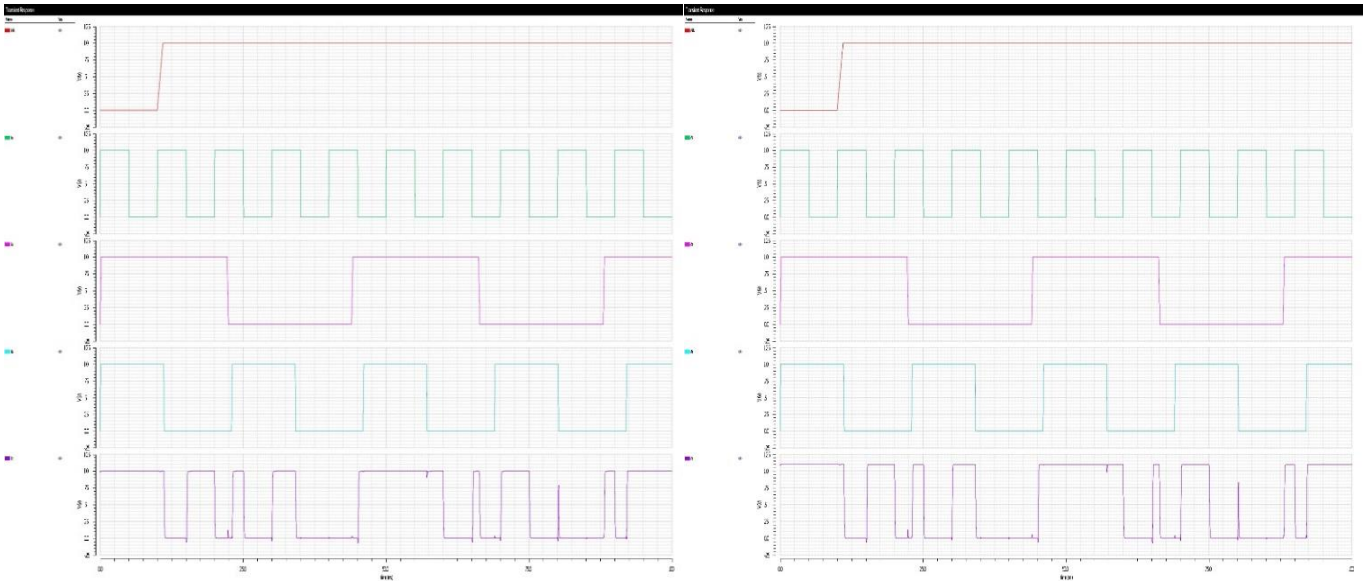

Figure 32(a) and 32(b) – Proposed (1V) and (1.1V)

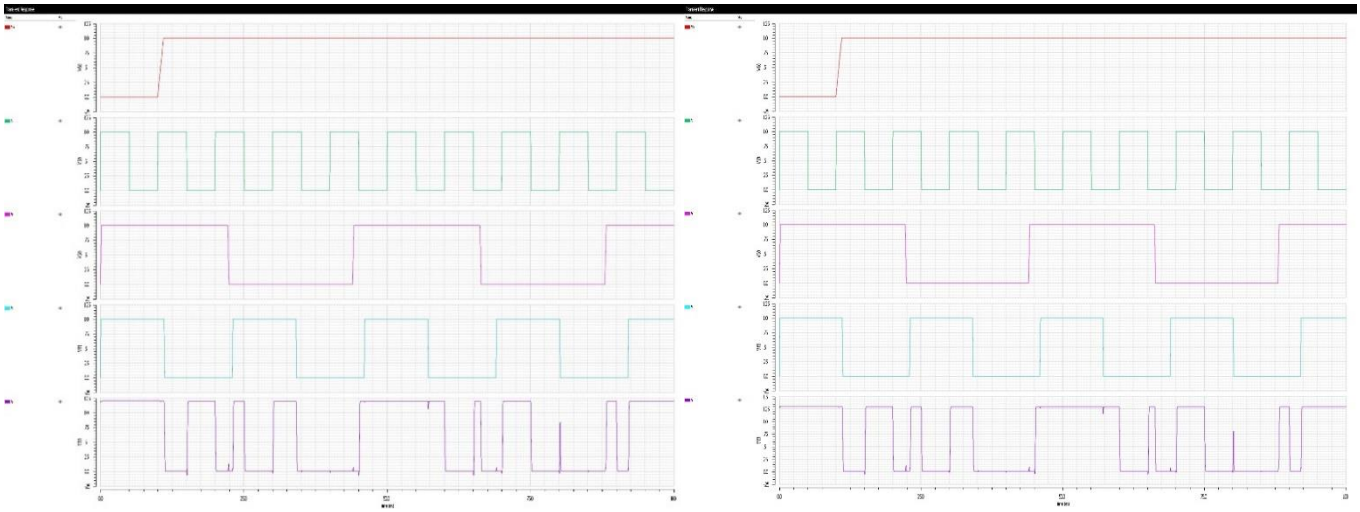

Figure 32(c) and 32(d) – Proposed (1,2V) and (1.3V)

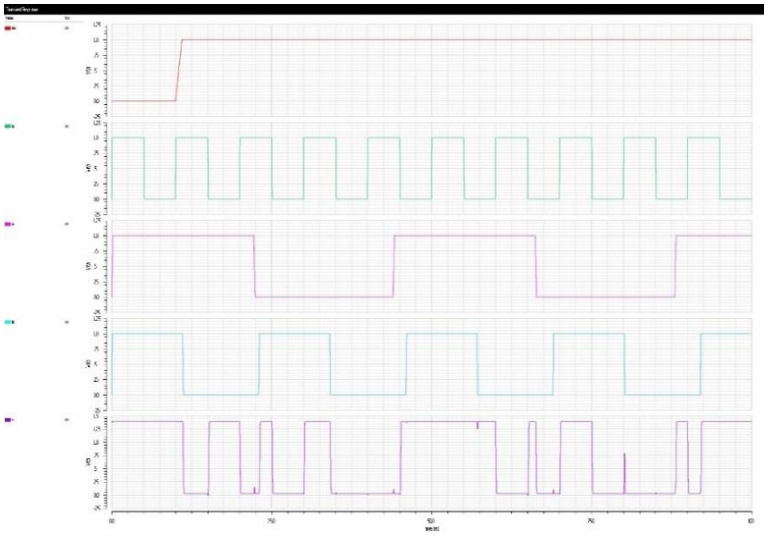

Figure 32(e) – Proposed (1.4V)

Between the conventional and proposed Dynamic DCVSL circuits, a comparison is done between the power consumption and temperature, varying the Voltage from 1V to 1.4V. The comparison shows better results for the proposed DCVSL circuit, in terms of power consumption.

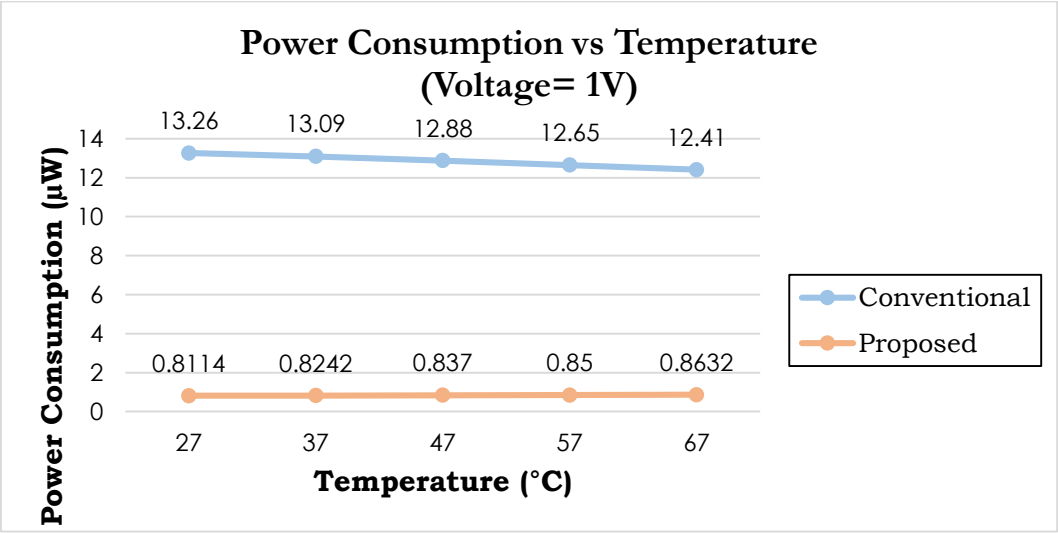

Figure 33(a) – Power Consumption vs Temperature (1V)

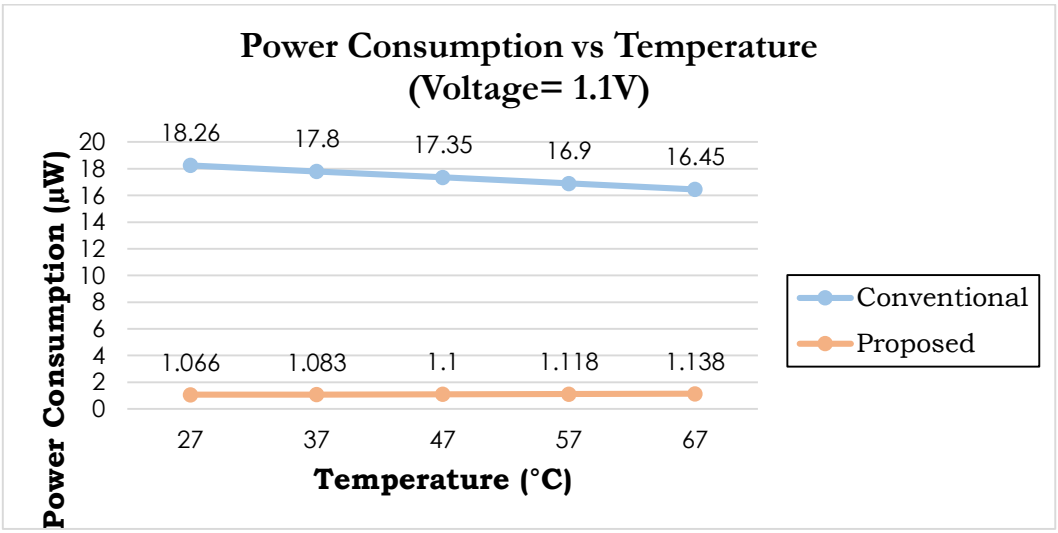

Figure 33(b) – Power Consumption vs Temperature (1.1V)

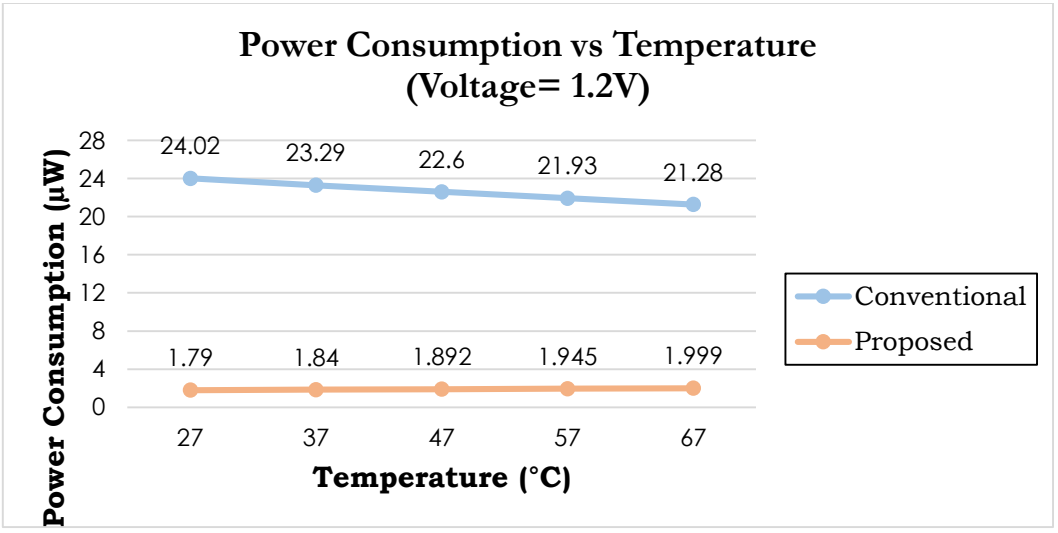

Figure 33(c) – Power Consumption vs Temperature (1.2V)

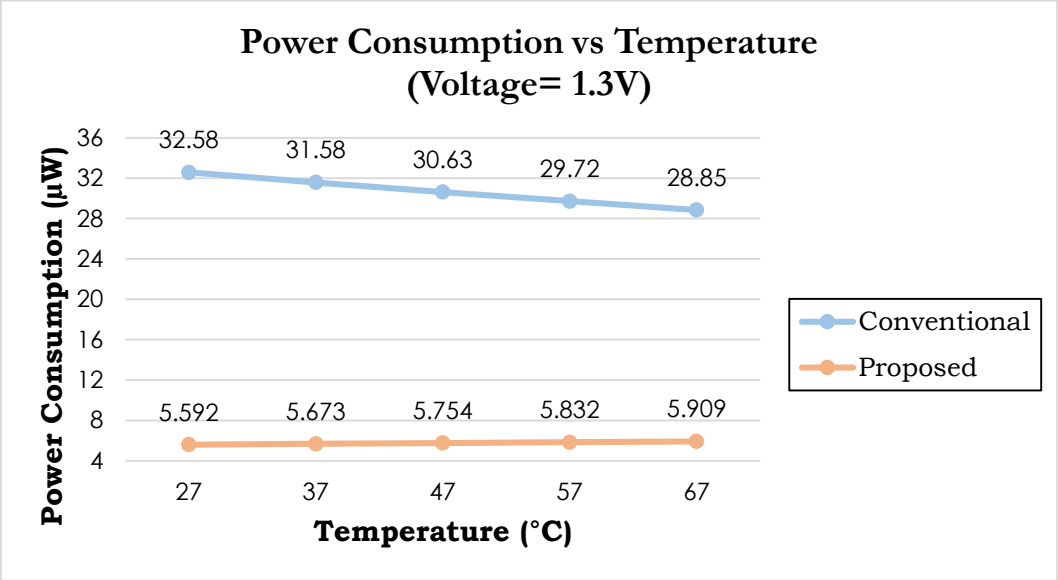

Figure 33(d) – Power Consumption vs Temperature (1.3V)

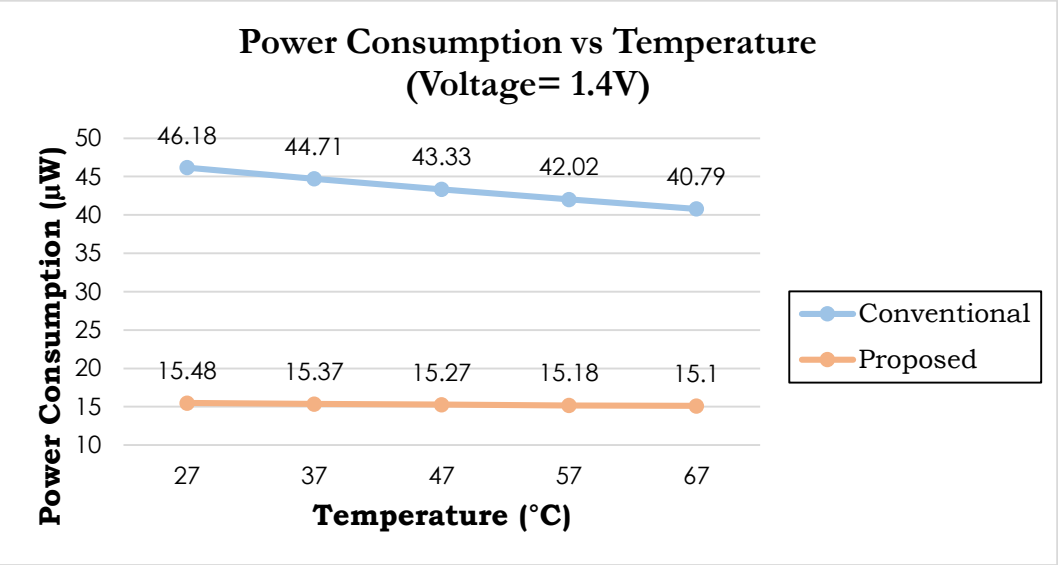

Figure 33(e) – Power Consumption vs Temperature (1.4V)

Now, we compare between the delay and temperature.

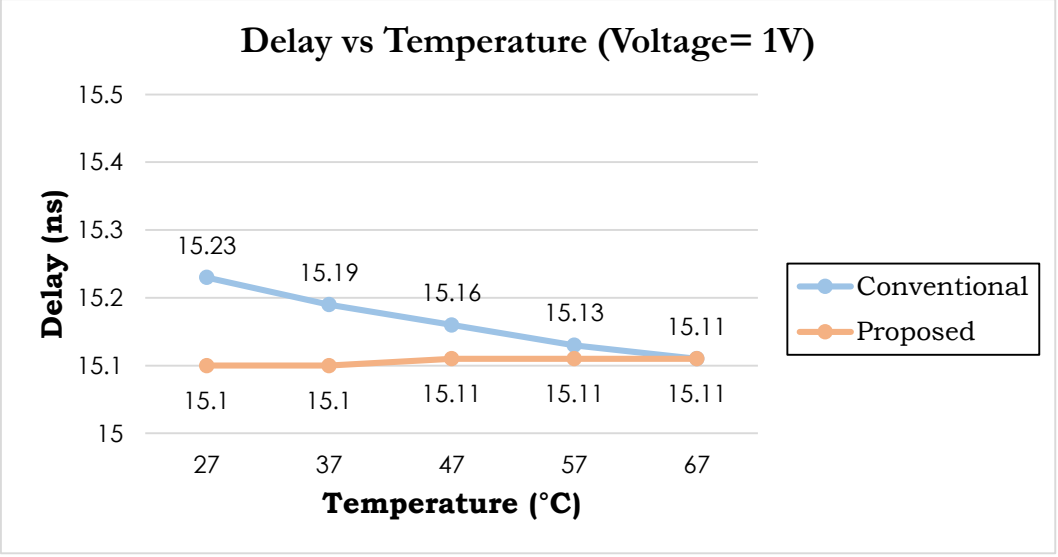

Figure 34(a) – Delay vs Temperature (1V)

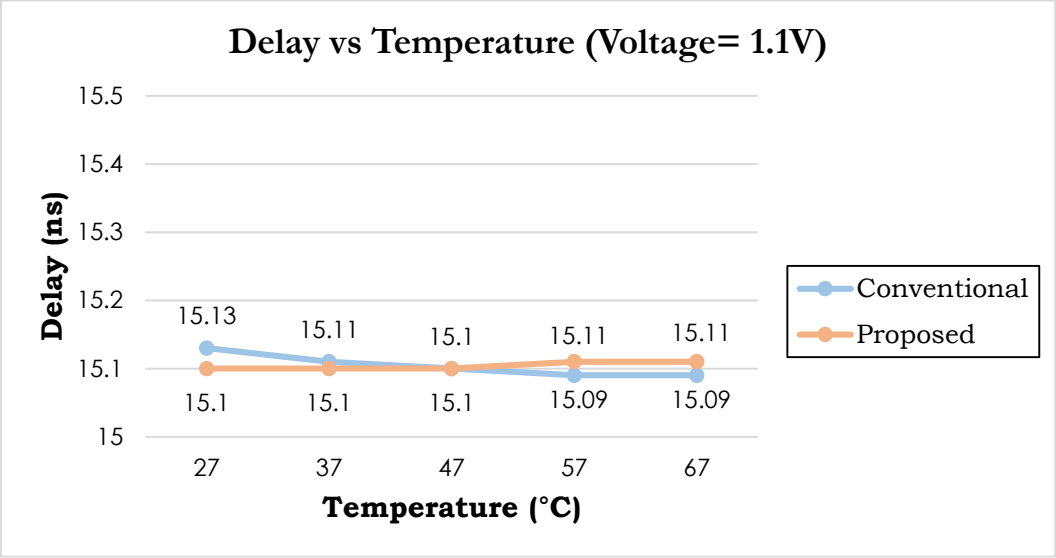

Figure 34(b) – Delay vs Temperature (1.1V)

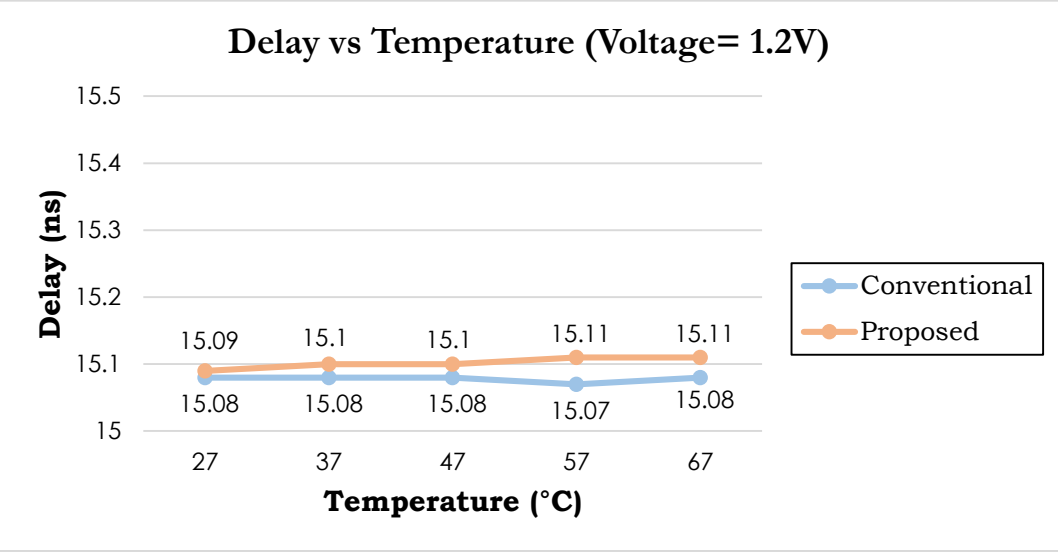

Figure 34(c) – Delay vs Temperature (1.2V)

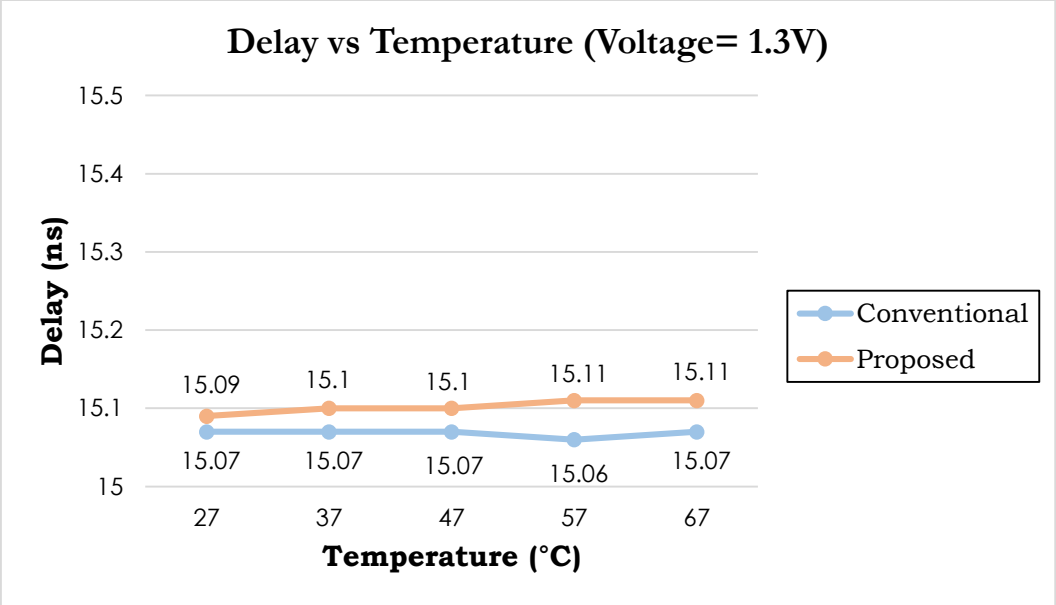

Figure 34(d) – Delay vs Temperature (1.3V)

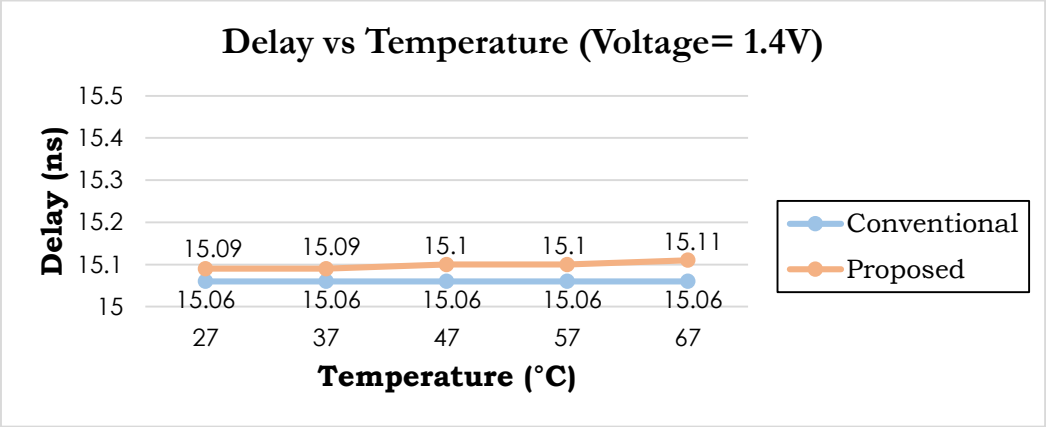

Figure 34(e) – Delay vs Temperature (1.4V)

Here, the delay is a bit more for the proposed DCVSL circuit, in most of the cases.

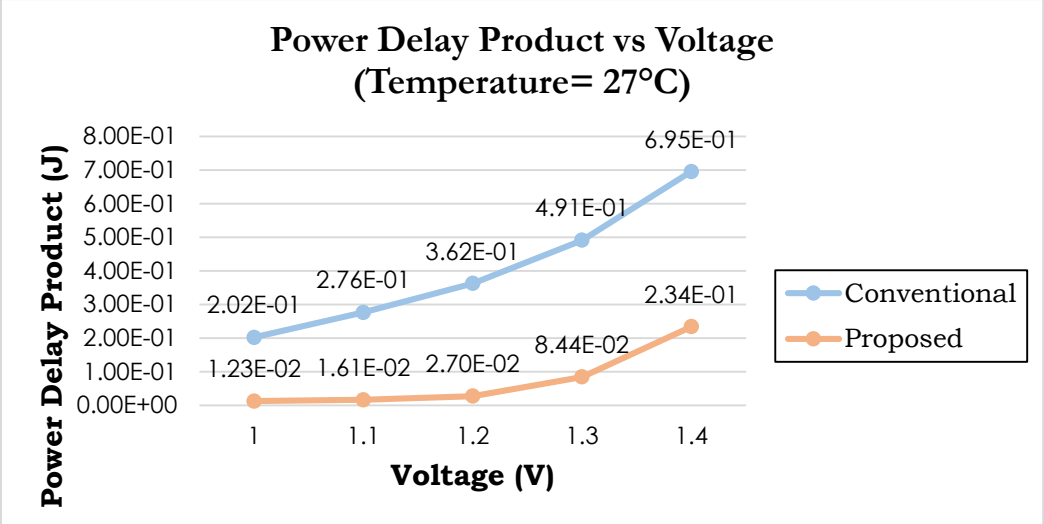

Figure 35 – Power Delay Product vs Voltage

➢ (5.3) Modified DCVSL –

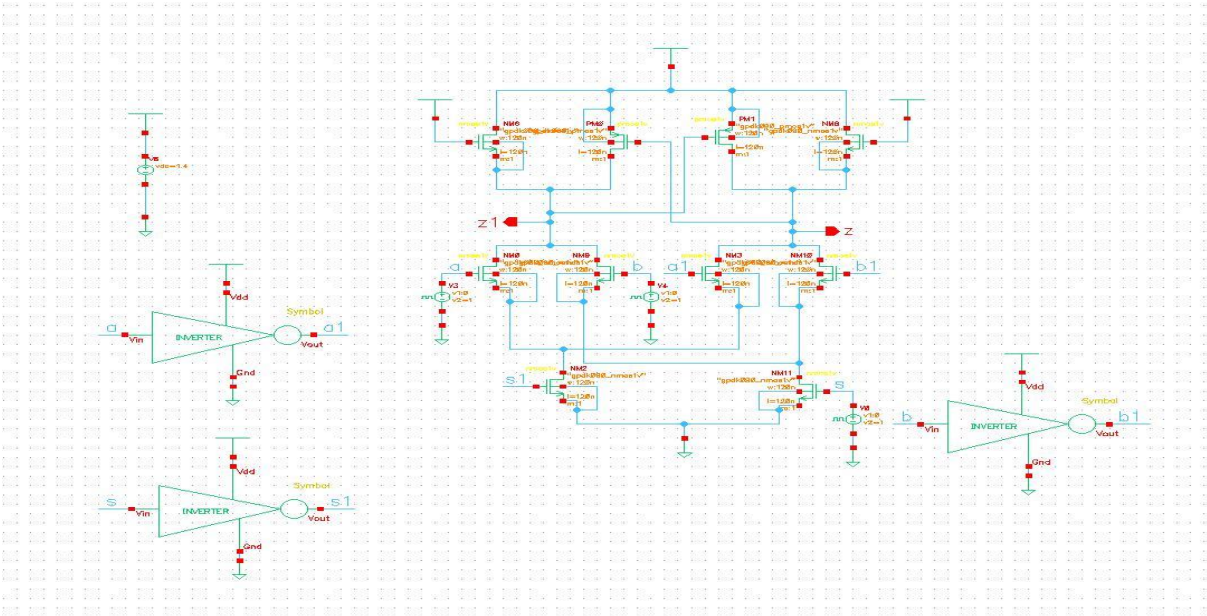

Figure 36(a) – Conventional Modified DCVSL

Here, there are 2 PMOS and 8 NMOS. 2 NMOS are equivalent to 2 PMOS. These 2 NMOS are having V_{DD} at the gates and the body connected to the source of it, which in turn is connected to the drain of its parallel PMOS. The gates of these 2 PMOS are connected to the outputs. The rest NMOS are same way connected to the source of it, without its connection being done with the ground. All the transistors within the DCVSL structure are having their (W/L) ratio as 1:1, since it is 120nm/120nm.

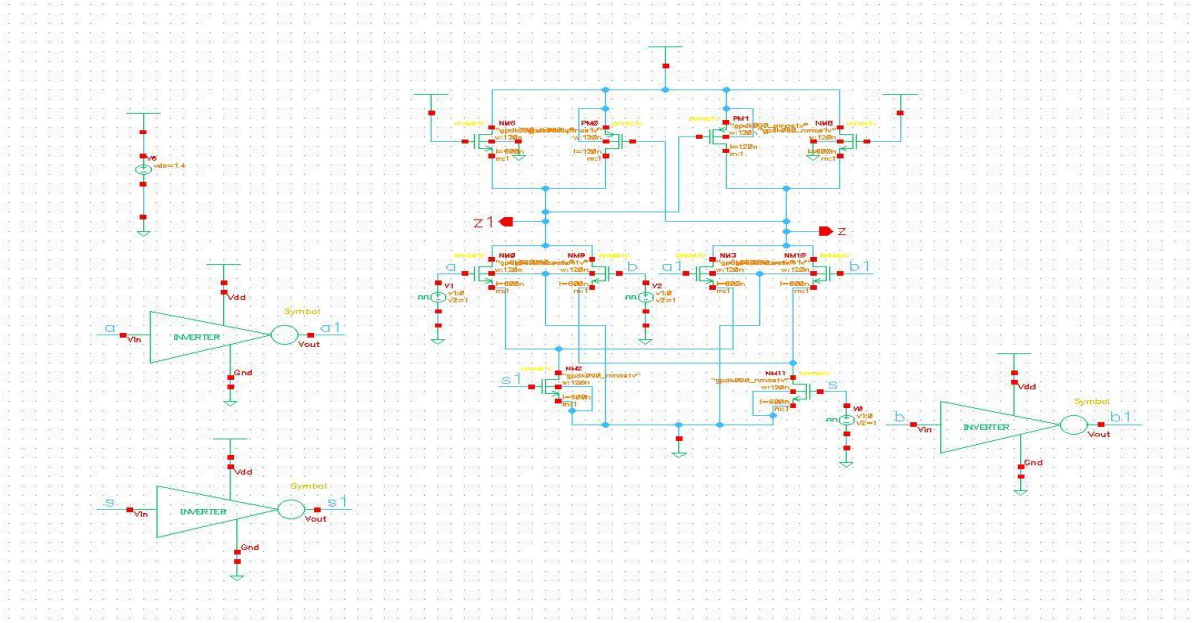

Figure 36(b) – Proposed Modified DCVSL

In the proposed circuit, the 2 NMOS equivalent to 2 PMOS are having their body grounded. And rest of the other 6 NMOS are having their body grounded too. Here, the (W/L) ratio is kept at 1:1.5, i.e. 120nm/180nm for the PMOS and for NMOS, it is 1:1, i.e. 120nm/120nm.

The transient responses for both the conventional and proposed circuits are shown below, with varying Voltages from 1V to 1.4V.

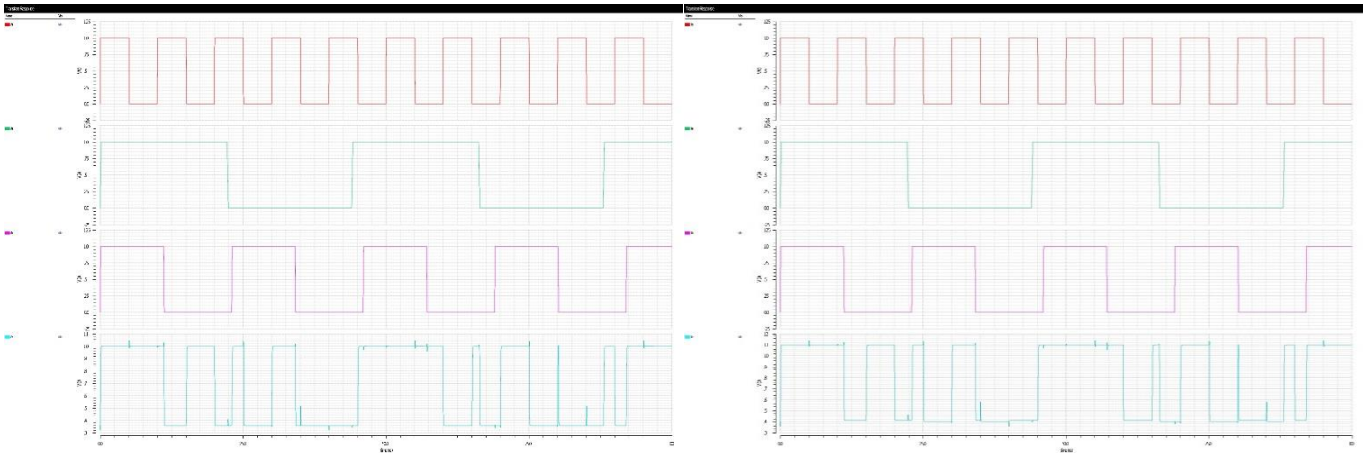

Figure 37(a) and 37(b) – Conventional (1V) and (1.1V)

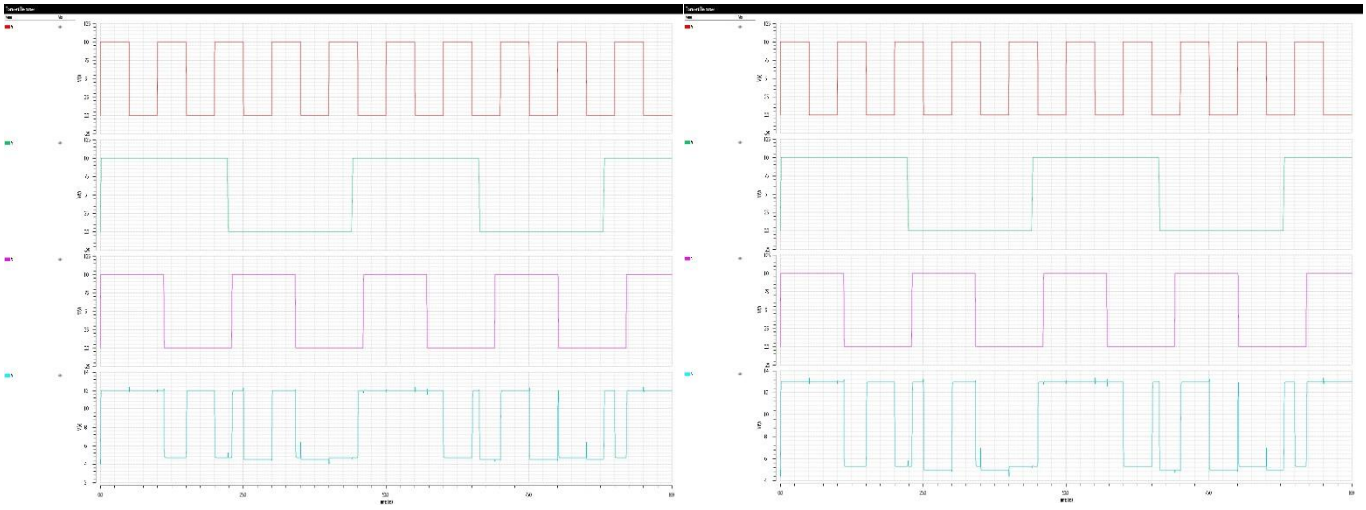

Figure 37(c) and 37(d) – Conventional (1.2V) and (1.3V)

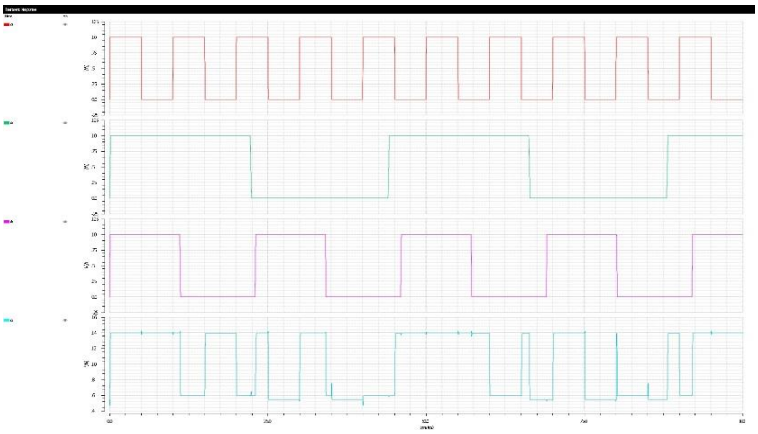

Figure 37(e) – Conventional (1.4V)

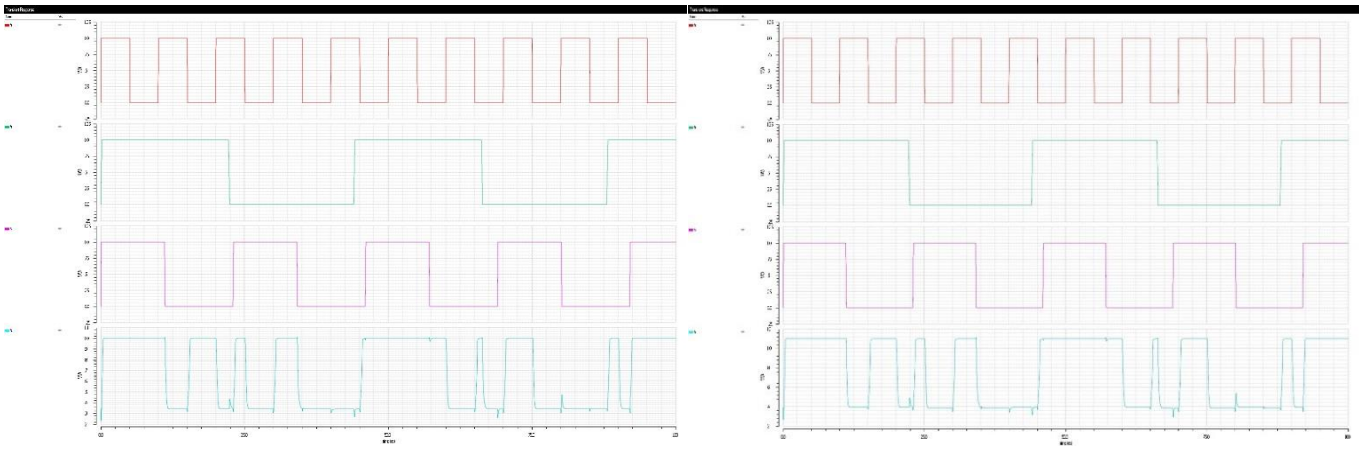

Figure 38(a) and 38(b) – Proposed (1V) and (1.1V)

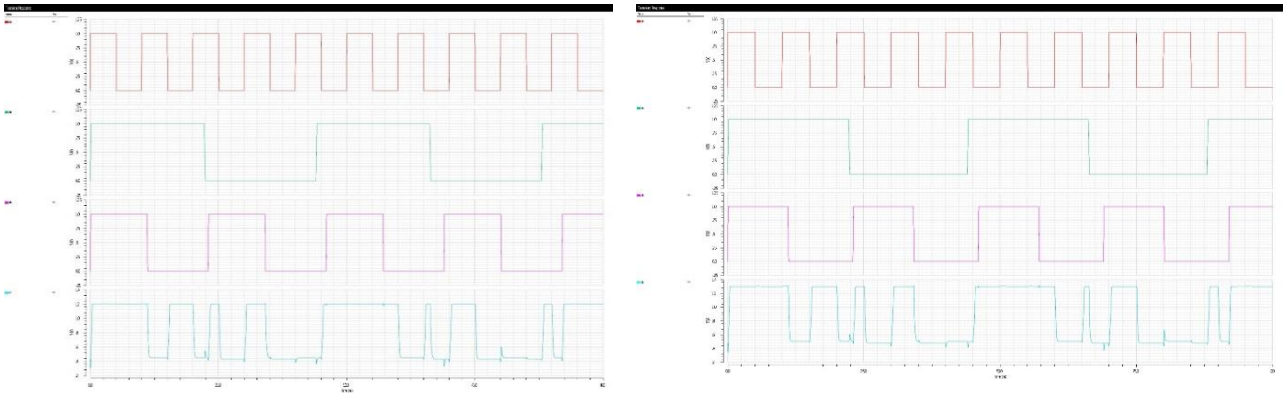

Figure 38(c) and 38(d) – Proposed (1.2V) and (1.3V)

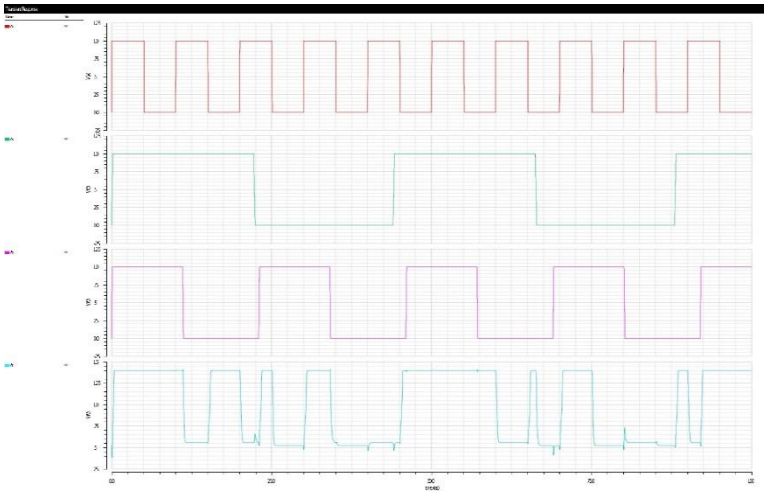

Figure 38(e) – Proposed (1.4V)

Between the conventional and proposed Modified DCVSL circuits, a comparison is done between the power consumption and temperature, varying the Voltage from 1V to 1.4V. The comparison shows better results for the proposed DCVSL circuit, in terms of power consumption.

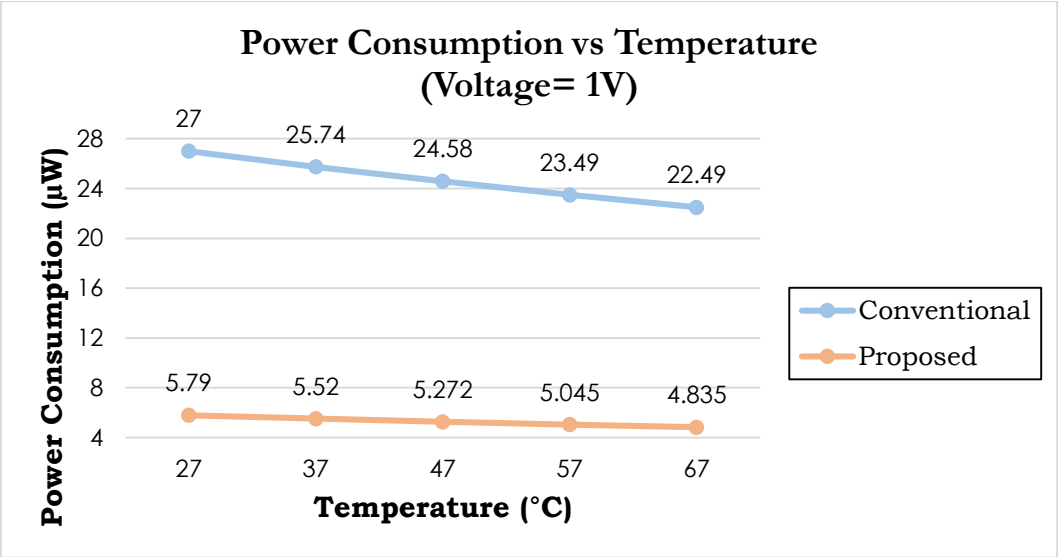

Figure 39(a) – Power Consumption vs Temperature (1V)

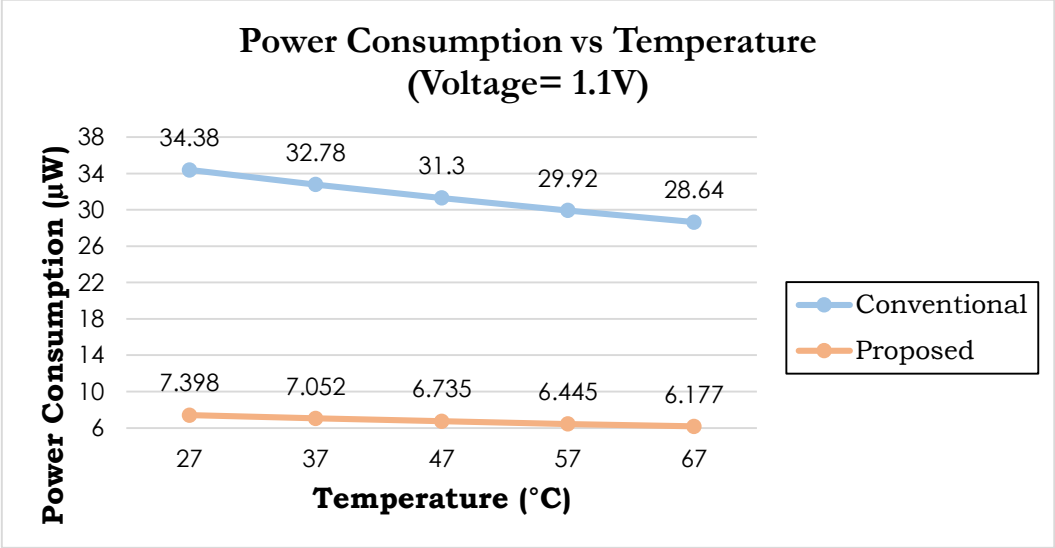

Figure 39(b) – Power Consumption vs Temperature (1.1V)

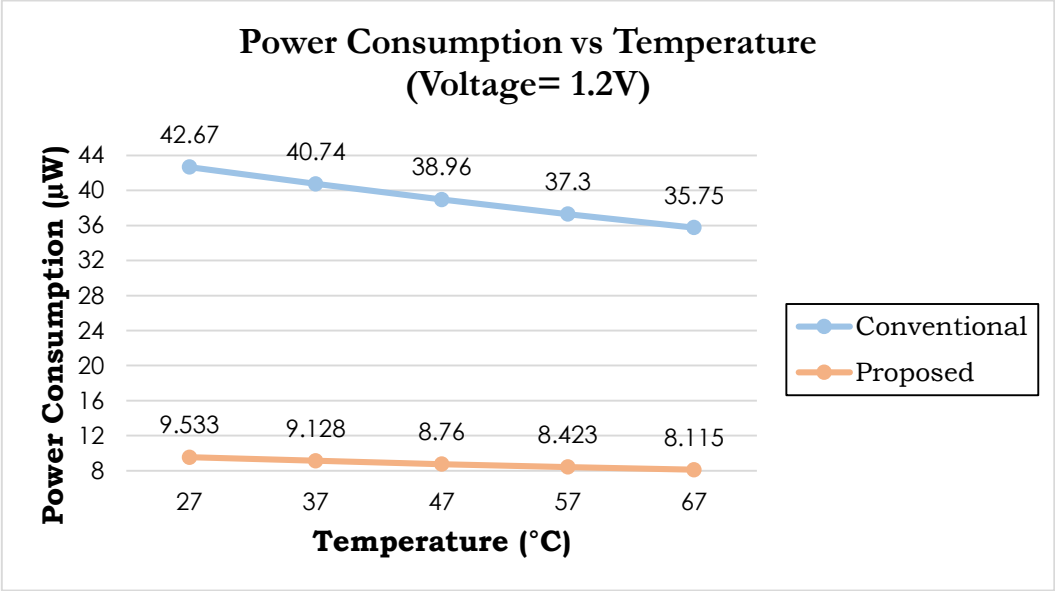

Figure 39(c) – Power Consumption vs Temperature (1.2V)

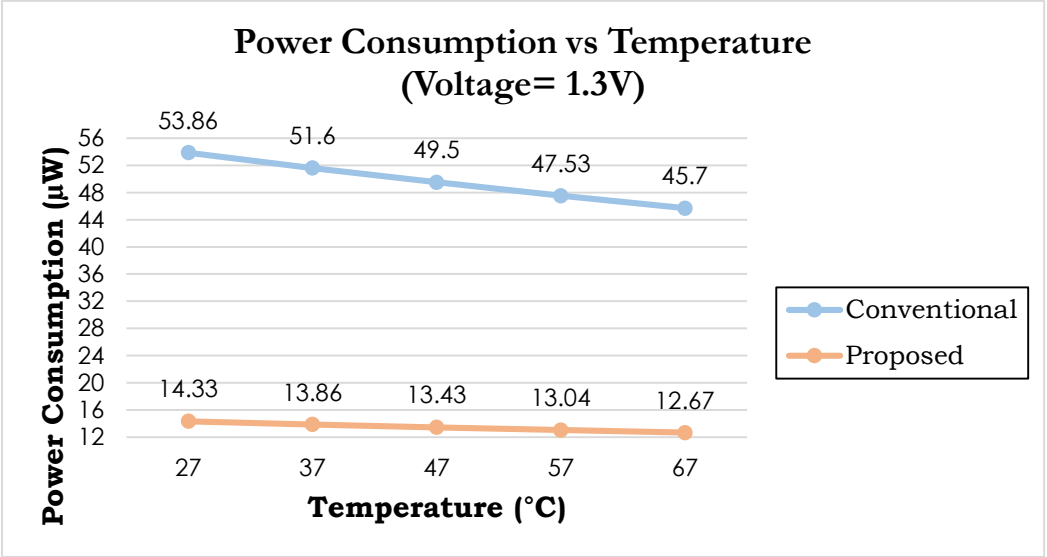

Figure 39(d) – Power Consumption vs Temperature (1.3V)

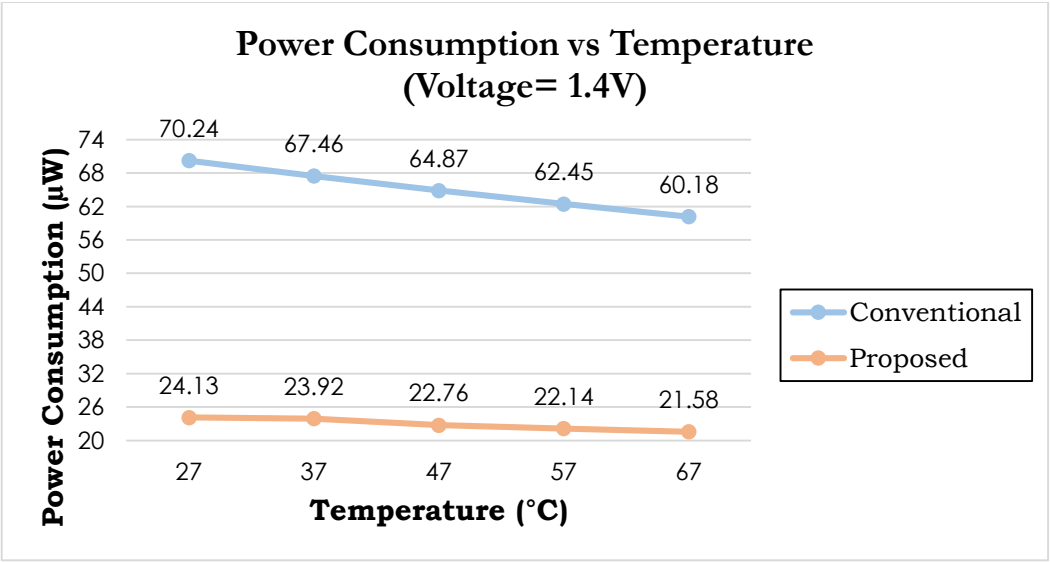

Figure 39(e) – Power Consumption vs Temperature (1.4V)

Now, we compare between the delay and temperature.

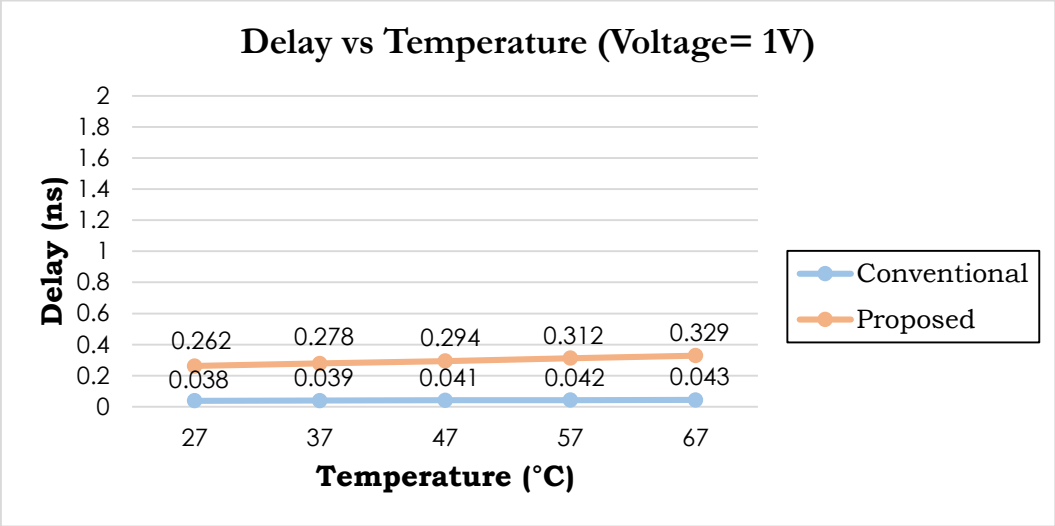

Figure 40(a) – Delay vs Temperature (1V)

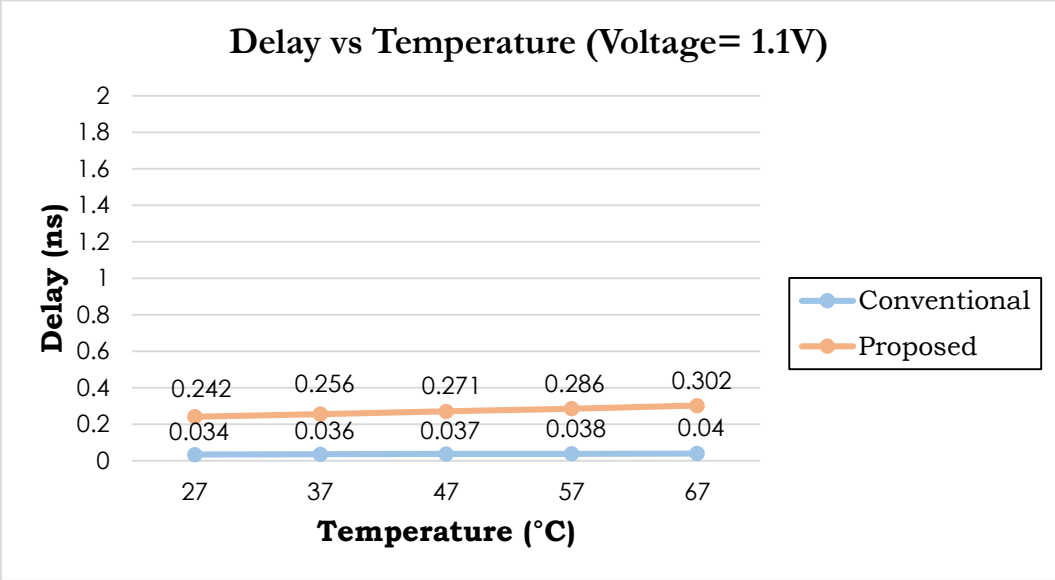

Figure 40(b) – Delay vs Temperature (1.1V)

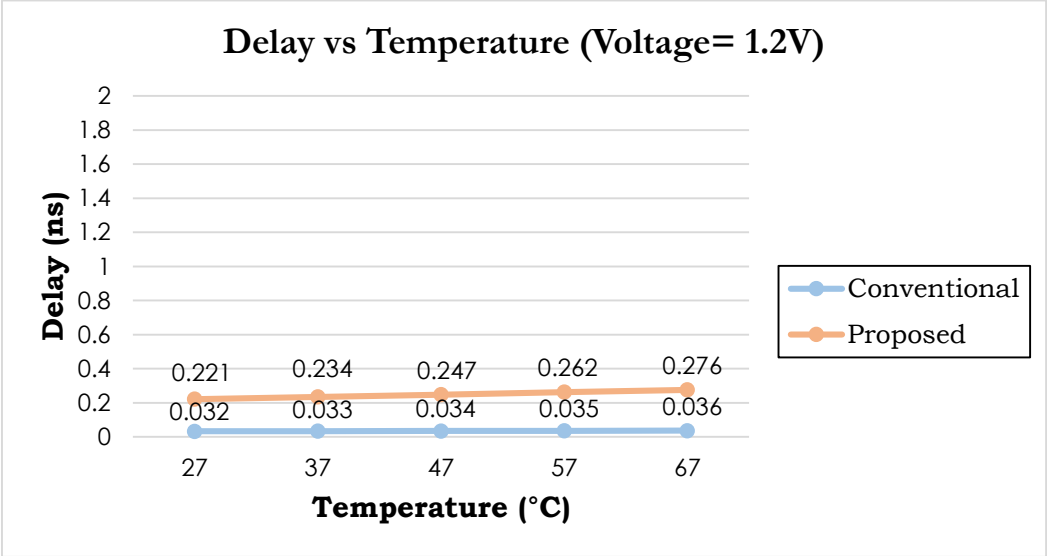

Figure 40(c) – Delay vs Temperature (1.2V)

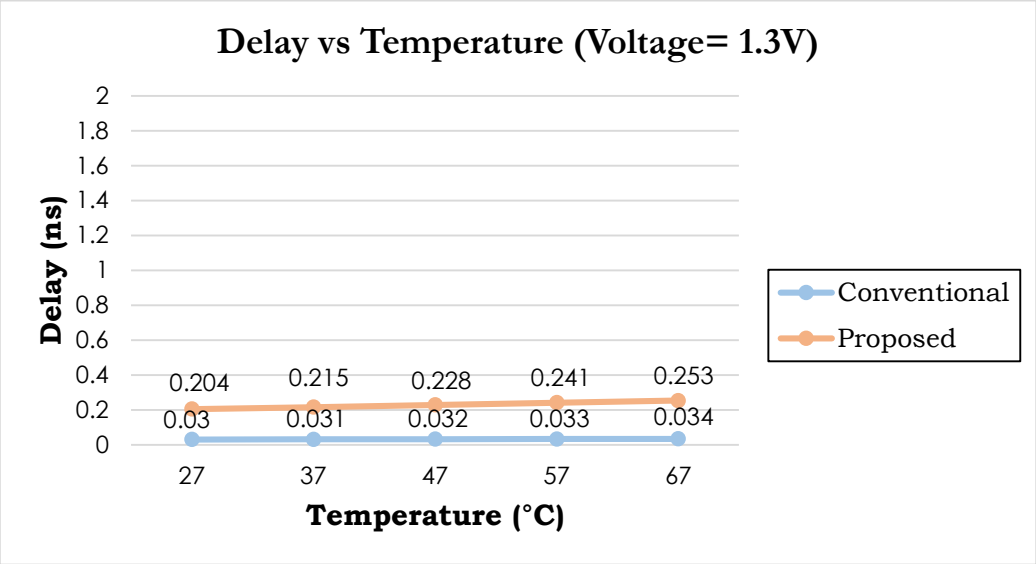

Figure 40(d) – Delay vs Temperature (1.3V)

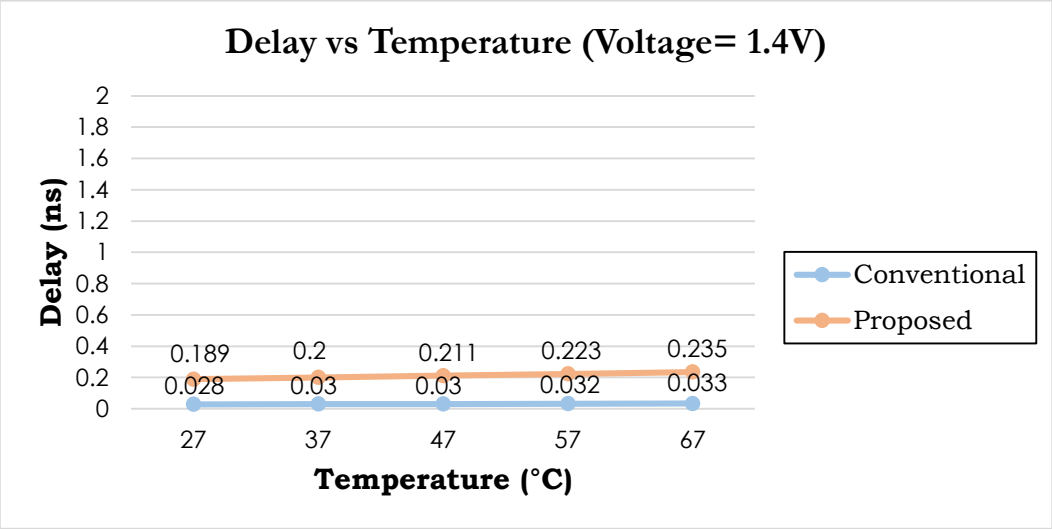

Figure 40(e) – Delay vs Temperature (1.4V)

The delay is a bit more for the proposed circuit.

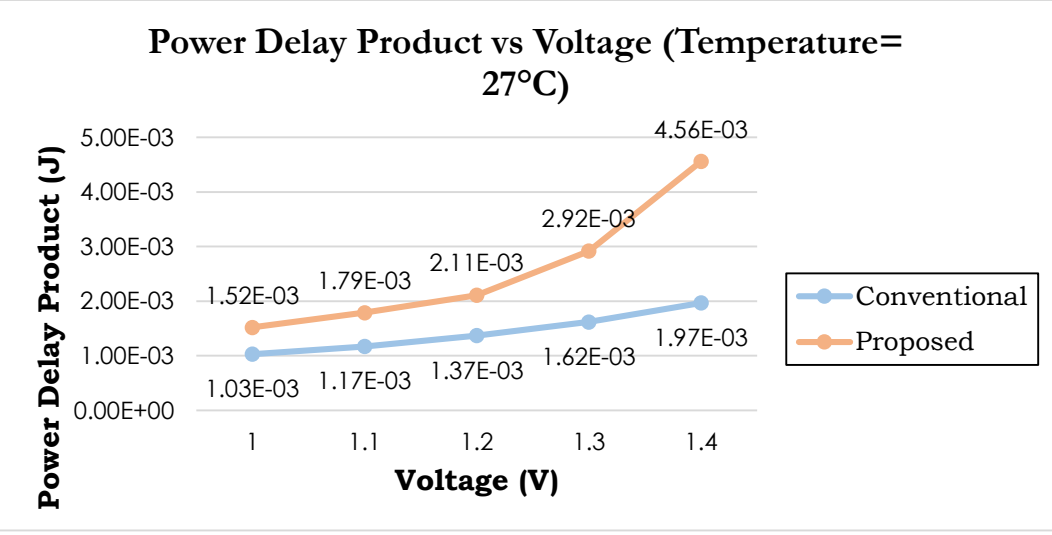

Figure 41 – Power Delay Product vs Voltage

	Temperature (°C)	Power Consumption (µW) (Conventional)					Power Consumption (µW) (Proposed)				
		1V	1.1V	1.2V	1.3V	1.4V	1V	1.1V	1.2V	1.3V	1.4V
Static DCVSL	27	2.76	4.03	5.98	10.7	20.6	0.76	0.99	1.62	4.81	12.9
	37	2.98	4.27	6.22	10.9	20.6	0.77	1.01	1.66	4.88	12.9
	47	3.18	4.50	6.46	11.2	20.7	0.78	1.02	1.70	4.95	12.8
	57	3.38	4.70	6.68	11.4	20.8	0.79	1.03	1.74	5.01	12.8
	67	3.56	4.89	6.89	11.6	20.8	0.81	1.05	1.79	5.07	12.7
Dynamic DCVSL		1V	1.1V	1.2V	1.3V	1.4V	1V	1.1V	1.2V	1.3V	1.4V
	27	13.3	18.3	24.0	32.6	46.2	0.81	1.06	1.79	5.59	15.5
	37	13.1	17.8	23.3	31.6	44.7	0.82	1.08	1.84	5.67	15.4
	47	12.9	17.6	22.6	30.6	43.3	0.84	1.10	1.89	5.75	15.3
	57	12.7	16.9	21.9	29.7	42.0	0.85	1.19	1.95	5.83	15.2
	67	12.4	16.5	21.3	28.9	40.8	0.86	1.14	1.99	5.91	15.1
Modified DCVSL		1V	1.1V	1.2V	1.3V	1.4V	1V	1.1V	1.2V	1.3V	1.4V
	27	27	34.4	42.7	58.9	70.2	5.79	7.40	9.54	14.3	24.1
	37	25.7	32.8	40.7	51.6	67.5	5.52	7.05	9.13	13.9	23.9
	47	24.6	31.3	38.9	49.5	64.9	5.27	6.74	8.76	13.4	22.8
	57	23.5	29.9	37.3	47.5	62.5	5.05	6.45	8.42	13.0	22.1
	67	22.5	28.6	35.8	45.7	60.2	4.84	6.18	8.12	12.8	21.6

Table 8 – Comparison between the three DCVSL structures (Power Consumption vs Temperature)

	Temperature (°C)	Delay (ns) (Conventional)					Delay (ns) (Proposed)				
		1V	1.1V	1.2V	1.3V	1.4V	1V	1.1V	1.2V	1.3V	1.4V
Static DCVSL	27	0.07	0.08	0.08	0.08	0.08	0.11	0.09	0.09	0.09	0.09
	37	0.08	0.08	0.08	0.08	0.09	0.11	0.10	0.10	0.09	0.09
	47	0.08	0.08	0.08	0.09	0.09	0.11	0.11	0.10	0.10	0.10
	57	0.09	0.09	0.09	0.09	0.10	0.12	0.11	0.11	0.10	0.10
	67	0.09	0.09	0.09	0.10	0.11	0.12	0.12	0.11	0.11	0.11
Dynamic DCVSL		1V	1.1V	1.2V	1.3V	1.4V	1V	1.1V	1.2V	1.3V	1.4V
	27	15.2	15.1	15.9	15.8	15.7	15.1	15.1	15.1	15.1	15.1
	37	15.2	15.1	15.9	15.8	15.7	15.1	15.1	15.1	15.1	15.1
	47	15.2	15.1	15.9	15.8	15.7	15.1	15.1	15.1	15.1	15.1
	57	15.1	15.1	15.8	15.7	15.7	15.1	15.1	15.1	15.1	15.1
	67	15.1	15.1	15.9	15.8	15.7	15.1	15.1	15.1	15.1	15.1
Modified DCVSL		1V	1.1V	1.2V	1.3V	1.4V	1V	1.1V	1.2V	1.3V	1.4V
	27	0.04	0.03	0.03	0.03	0.03	0.26	0.24	0.22	0.20	0.19
	37	0.04	0.04	0.03	0.03	0.03	0.28	0.26	0.23	0.22	0.20
	47	0.04	0.04	0.03	0.03	0.03	0.29	0.27	0.28	0.23	0.21
	57	0.04	0.04	0.04	0.03	0.03	0.31	0.29	0.26	0.24	0.22
	67	0.04	0.04	0.04	0.03	0.03	0.33	0.30	0.28	0.25	0.24

Table 9 – Comparison between the three DCVSL structures (Delay vs Temperature)

Voltage	Power Delay Product (Joules)					
	Static DCVSL		Dynamic DCVSL		Modified DCVSL	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
1	2.12E-04	8.03E-05	2.02E-01	1.23E-02	1.03E-03	1.52E-03
1.1	3.02E-04	9.81E-05	2.76E-01	1.61E-02	1.17E-03	1.79E-03
1.2	4.55E-04	1.52E-04	3.62E-01	2.70E-02	1.37E-03	2.11E-03
1.3	8.33E-04	4.38E-04	4.91E-01	8.44E-02	1.62E-03	2.92E-03
1.4	1.69E-03	1.16E-03	6.95E-01	2.34E-01	1.97E-03	4.56E-03

Table 10 – Power Delay Product vs Voltage (Temperature= 27°C)

The power consumption for all the DCVSL structures is considerably reduced in the proposed circuits, but the delay is comparatively a bit more for all the proposed circuits. As we can see, for the Static DCVSL, the delay is more for the proposed one than the conventional one in negligible amount, whereas for the Dynamic one, it is more or less same. And for the Modified DCVSL, we can clearly point out the difference in delay between the conventional and proposed circuits.

In case of the Power Delay Product for all the three DCVSL circuits, it is seen that the values are less for both the Static and Dynamic DCVSL whereas for the Modified DCVSL, it is more in case of the proposed circuit.

Chapter 6

Performance Analysis of **DCVSL Adder Circuits**

The previous chapter ends with the analysis of all the DCVSL structures, starting with Static, then Dynamic and ending with Modified. The transient responses are shown, along with the power consumption and delay with varying temperature and voltages. The power consumption is better for the proposed structures whereas the delay is comparatively better for the conventional ones. Therefore, PDP which shows that Static and Dynamic DCVSL gives better PDP result in the proposed circuit than the conventional one, whereas for the Modified DCVSL, it's just the opposite.

Now, we study the full adder circuits, implemented using the XOR and the XNOR logic on the DCVSL structures or rather one can say 3 XOR gates implemented on the DCVSL structures.

$$Sum = A \oplus B \oplus Cin$$
$$Cout = Cin(A \oplus B) + A \cdot (A \odot B)$$

Here, 3 inputs namely (a, b, C_{in}) are used and 2 outputs Sum and C_{out} are used. Complementary inputs a1 and $C_{in}1$ are also added using separate inverters. There is also a complementary output $C_{out}1$. Apart from the numerous transistors of DCVSL structures, there is also many more other transistors forming the Adder Circuit, including both PMOS and NMOS. The first XOR gate having the input 'b' connected to the source of one PMOS and to the gate of another PMOS, has the (W/L) ratio of both the PMOS as (2400nm/120nm) and (3600/120nm), i.e. 20:1 and 30:1, respectively and NMOS has 120nm/120nm, i.e. 1:1 in ratio. Same way, the input 'C_{in}' has both of its PMOS in the ratio 20:1 and 30:1 respectively and NMOS as 1:1. Rest of the other transistors are in default ratio, i.e. 1:1 including the separate inverters. This is for the conventional circuits of all the DCVSL circuits.

In the proposed circuits of all the DCVSL adders, Stacking technique is used to reduce the standby leakage, i.e. the sub-threshold leakage current which ultimately reduces the power dissipation. To understand the concept, a two-input NAND gate is shown in Figure 38.

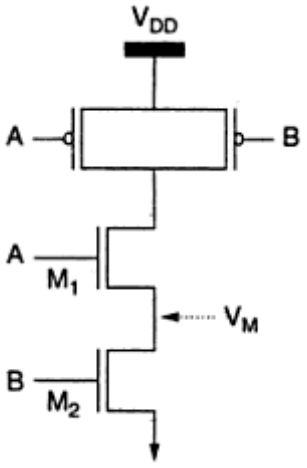

Figure 42 – Stacking Effect in 2-input NAND gate

Due to a small drain current, the voltage at the intermediate node (V_M) is positive when both M_1 and M_2 are turned off [27]. Therefore, it has following 3 effects due to this positive potential at the intermediate node :

- The V_{gs1} of M_1 becomes negative due to positive source potential V_M and therefore resulting in reduction of sub-threshold current.

- The V_{bs1} of M_1 decreases and resulting in reduction of sub-threshold current by increasing threshold voltage (more body effect), when $V_M > 0$.

- The V_{ds1} of M_1 decreases and resulting in reduction of sub-threshold current by increasing threshold voltage (less DIBL), when $V_M > 0$.

Therefore as proposed in [28], the leakage of stack circuit is fewer in amount that of a single transistor.

In this chapter, we deal with the same previously proposed parameters, but this time it's just the adder that is being added to all the DCVSL structures. Here also, we go for the power-delay-product (PDP) since the parameter power consumption is better for the proposed circuit, whereas the parameter delay is better for the conventional ones.

Now, coming back to the adder circuits, we have the following in all of the three –

For the input 'a', the 'Period' is kept at 42ns and the 'Pulse Width' is at 22ns.
For the input 'b', the 'Period' is kept at 23ns and the 'Pulse Width' is at 11ns.
For the input 'C_{in}', the 'Period' is kept at 15ns and the 'Pulse Width' is at 8ns.

➢ (6.1) Static DCVSL Adder –

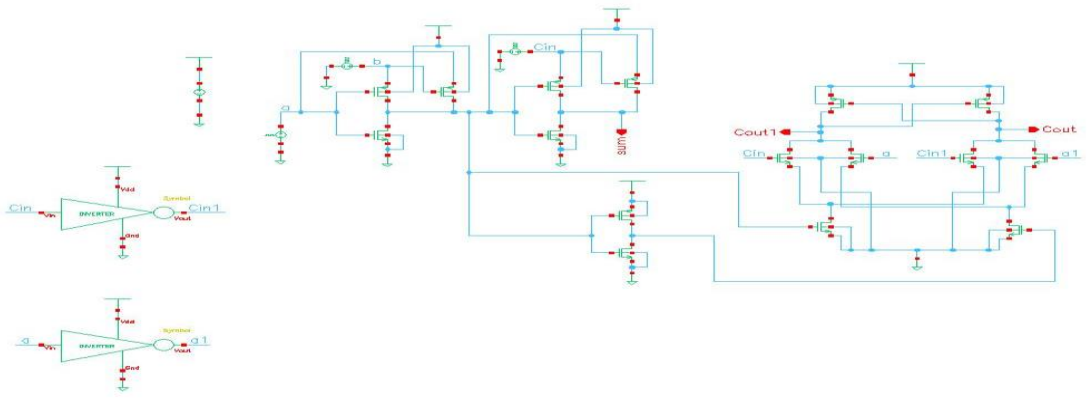

Figure 43(a) – Conventional Static DCVSL Adder

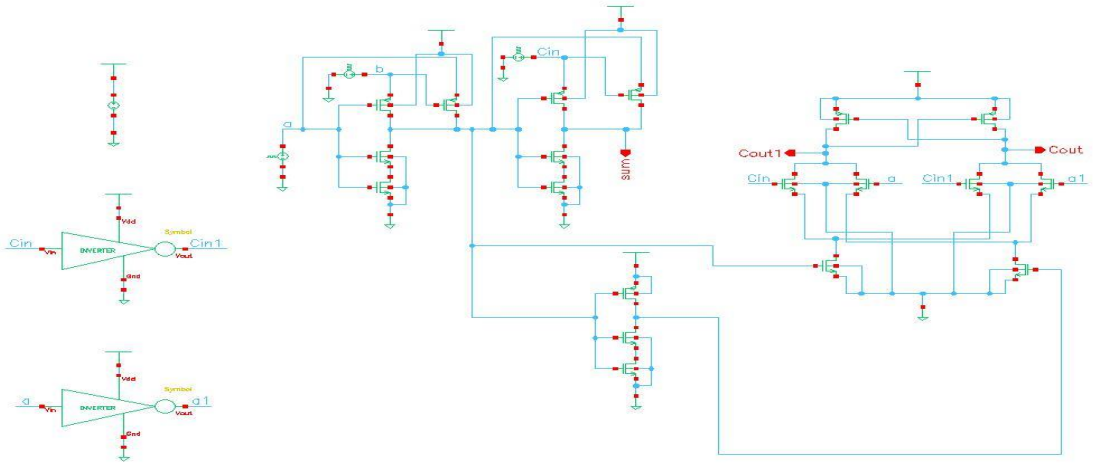

Figure 43(b) – Proposed Static DCVSL Adder

The proposed circuit differs from the conventional one, by the stacking technique. Stacking is used here, which includes an extra NMOS with the (W/L) ratio as 120nm/120nm, i.e. 1:1 in the 3 XOR gates. The rest of the configuration remains the same.

The transient responses are shown below –

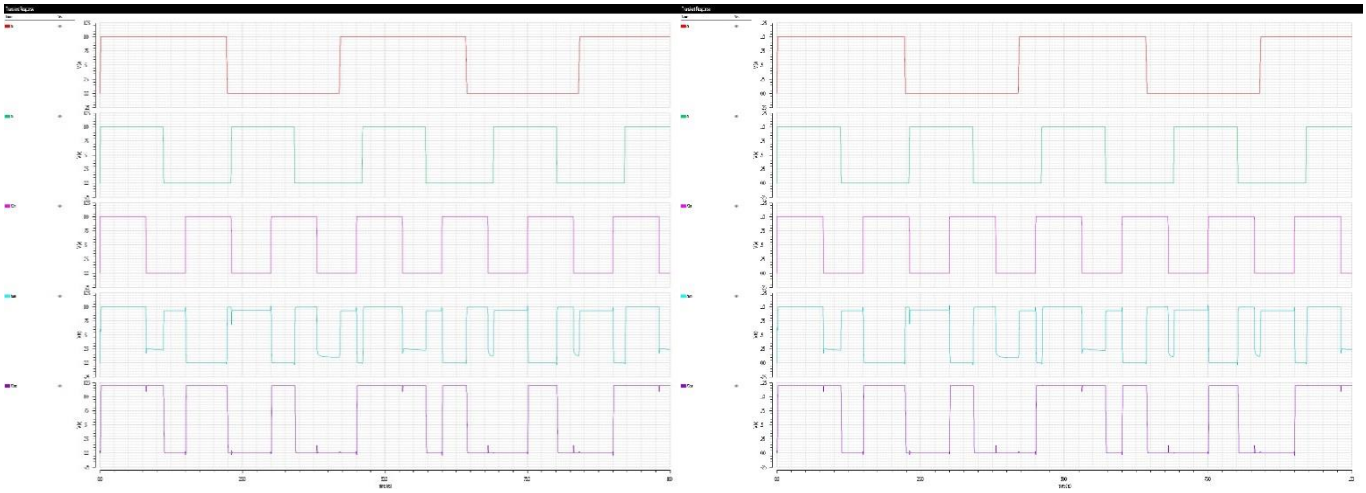

Figure 44(a) and 44(b) – Conventional (1V) and (1.1V)

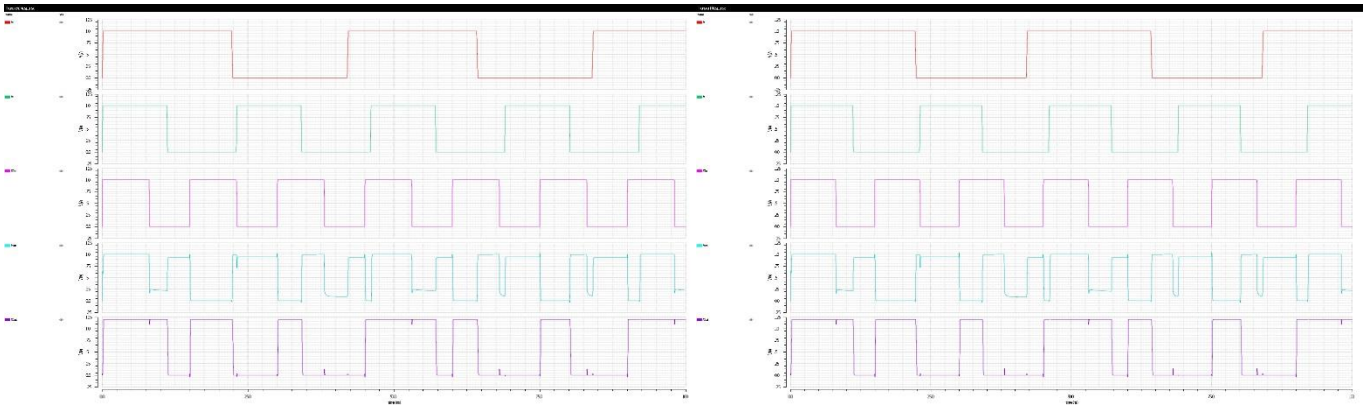

Figure 44(c) and 44(d) – Conventional (1.2V) and (1.3V)

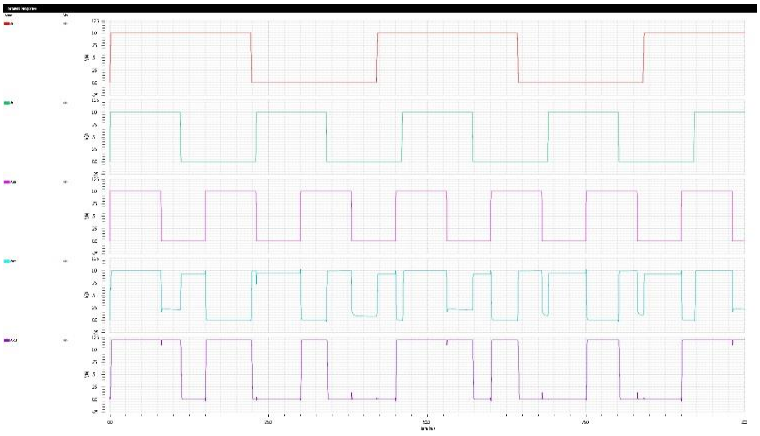

Figure 44(e) – Conventional (1.4V)

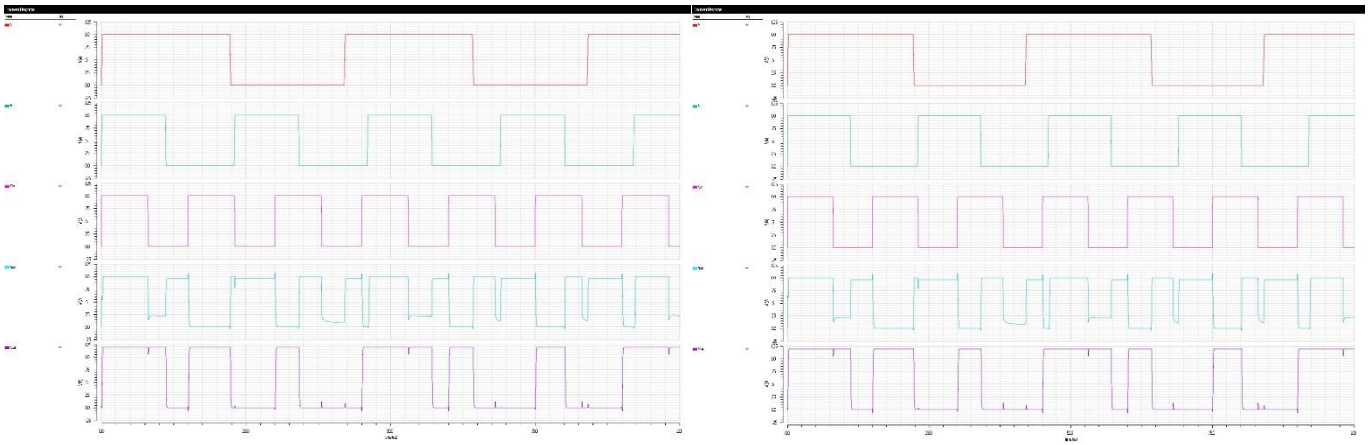

Figure 45(a) and 45(b) – Proposed (1V) and (1.1V)

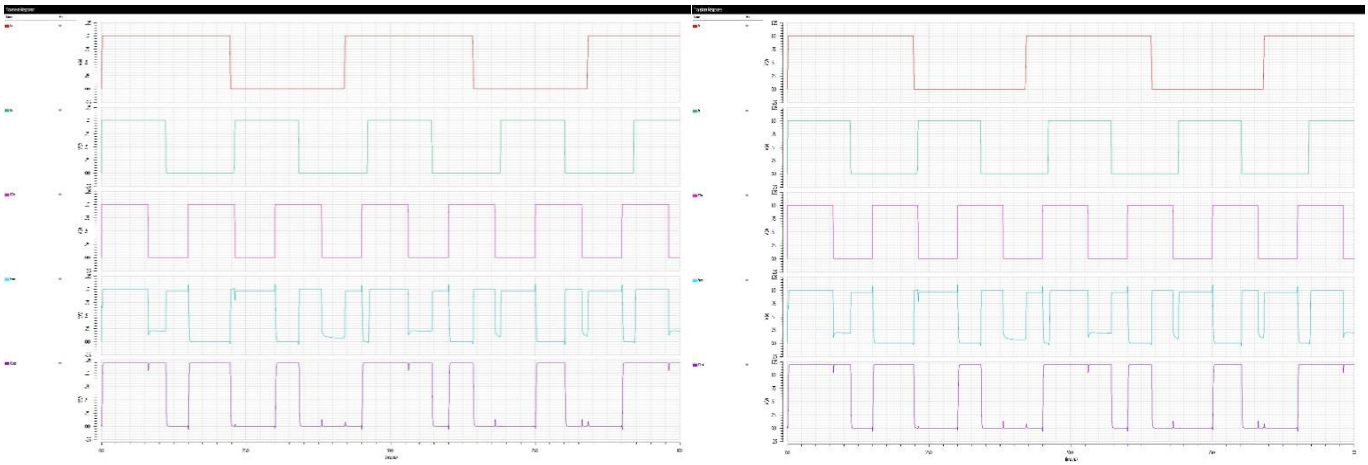

Figure 45(c) and 45(d) – Proposed (1.2V) and (1.3V)

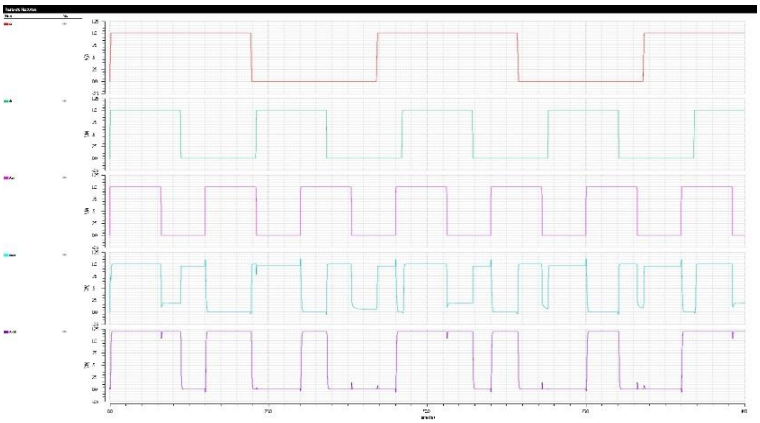

Figure 45(e) – Proposed (1.4V)

Now, we do the analysis keeping parameters such as power consumption, temperature, delay, voltage. First, there is the analysis between power consumption and temperature, keeping the voltage constant at 1.2V. We here can see that the proposed circuit of all the DCSVL structures produces better results in terms of power consumption by lowering its value.

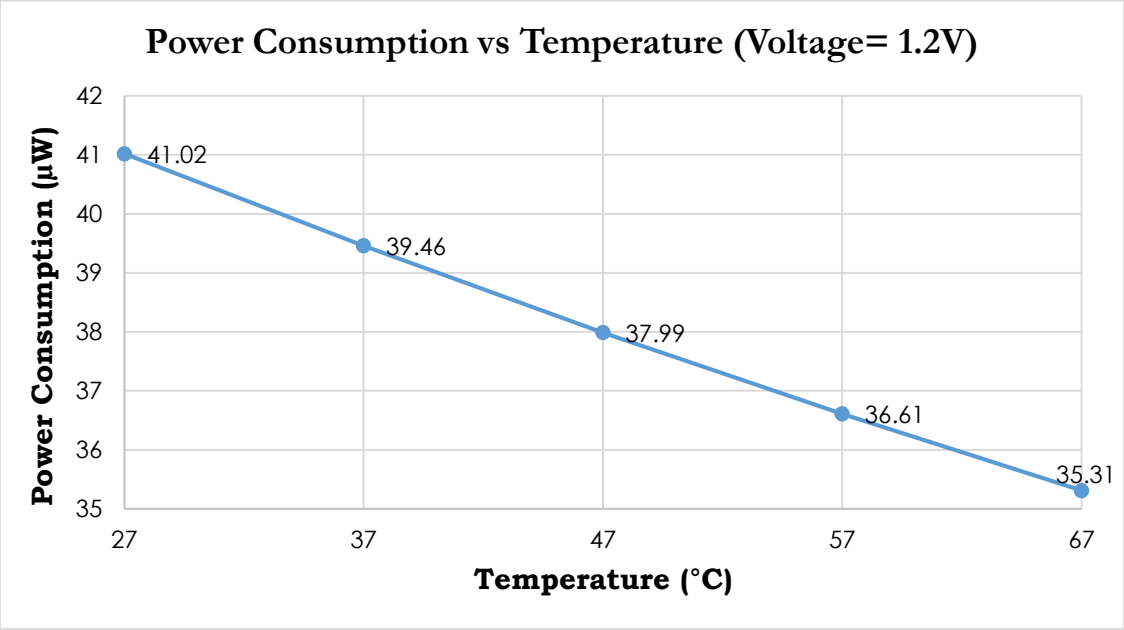

Figure 46(a) – Conventional Power Consumption vs Temperature

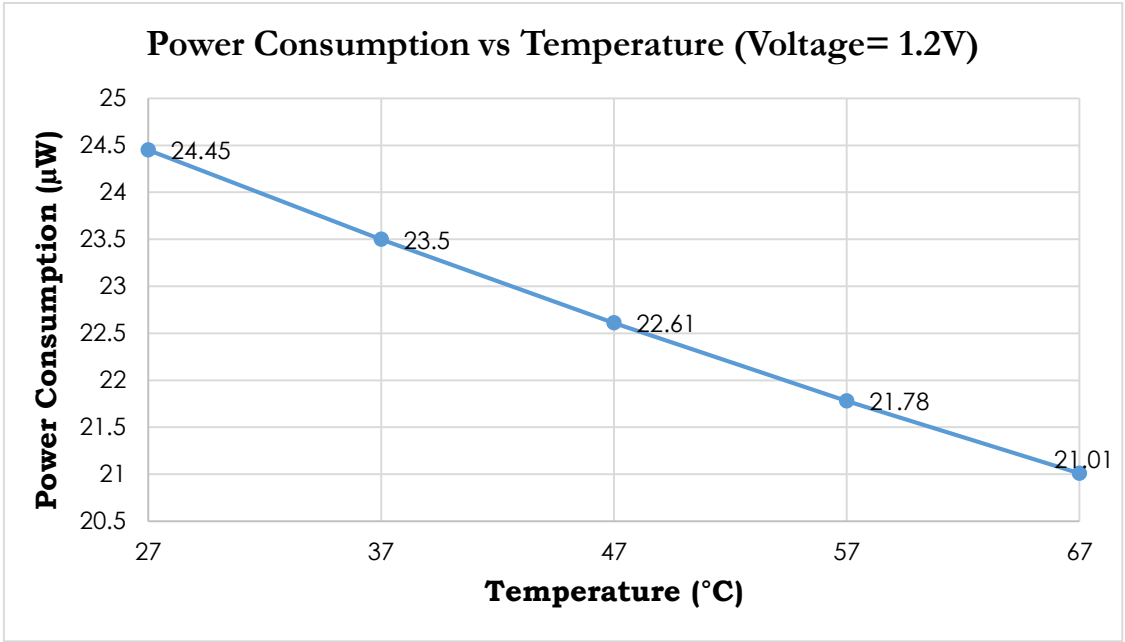

Figure 46(b) – Proposed Power Consumption vs Temperature

Now, we do the analysis of delay versus temperature, keeping the voltage same at 1.2V. Here, we see that the conventional circuit has better result than the proposed one, in terms of delay.

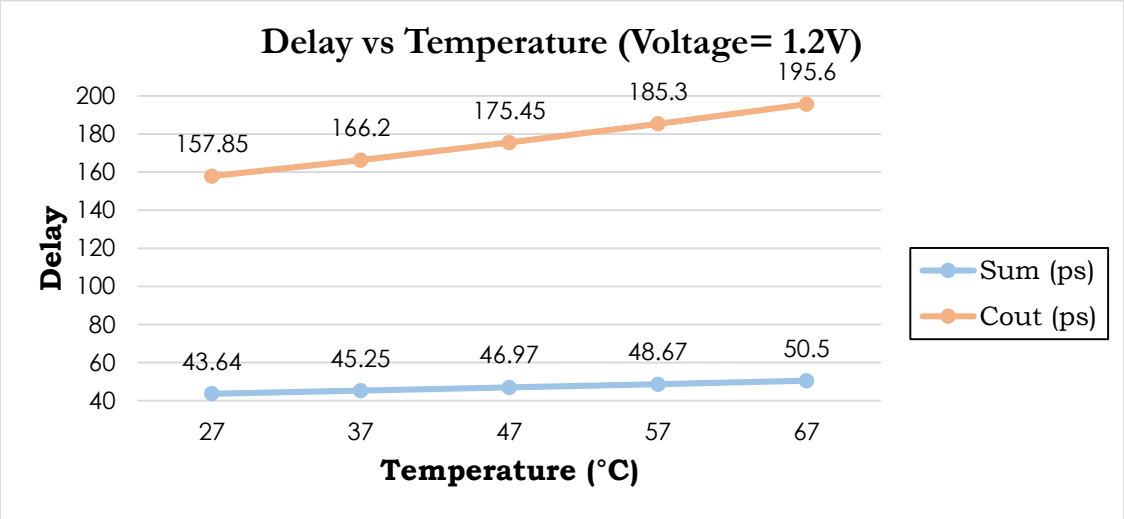

Figure 47(a) – Conventional Delay vs Temperature

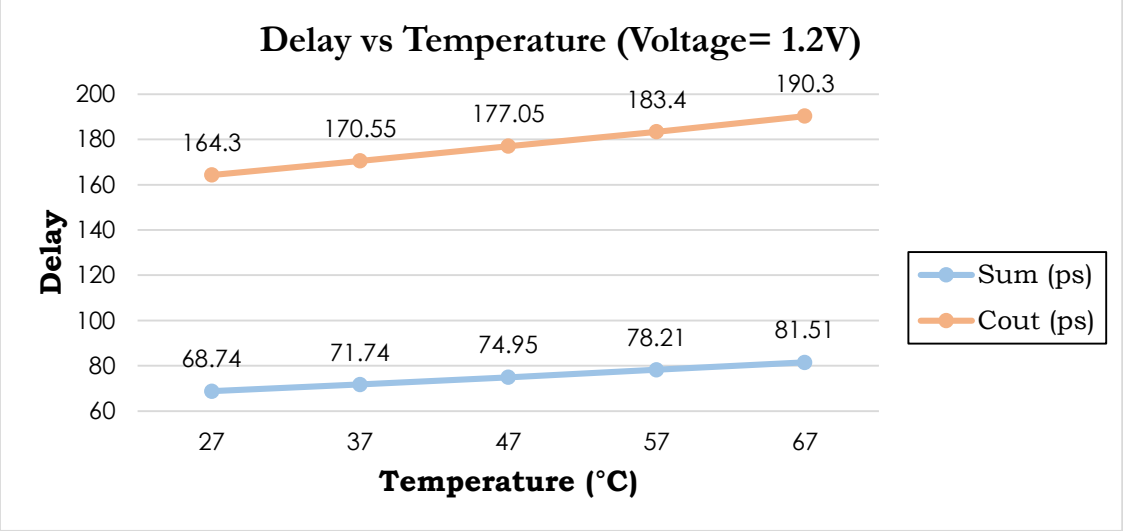

Figure 47(b) – Proposed Delay vs Temperature

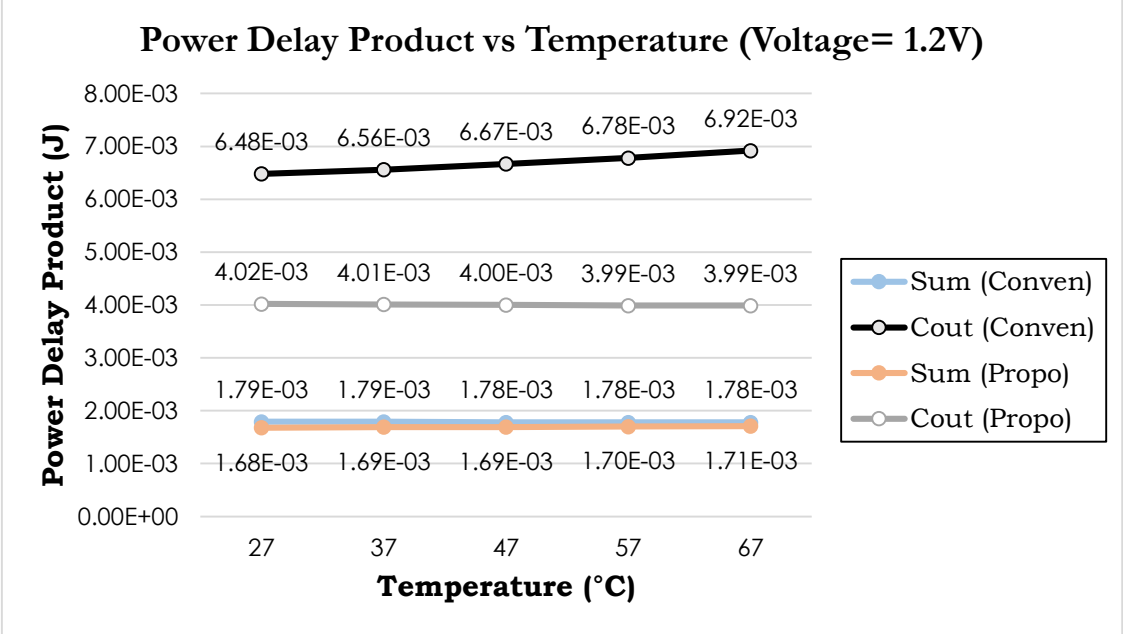

Figure 48– Power Delay Product vs Temperature

➢ (6.2) Dynamic DCVSL Adder –

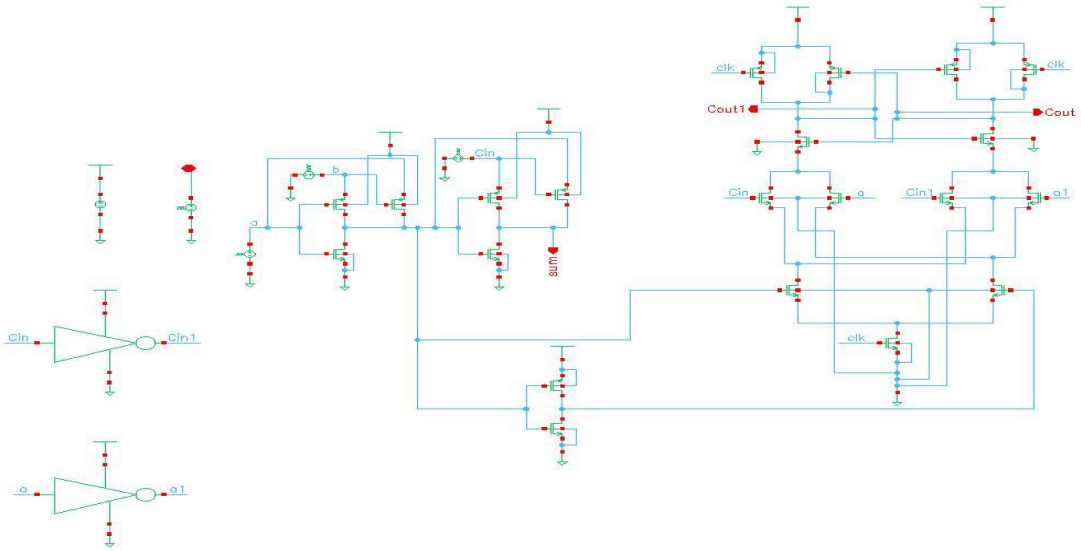

Figure 49(a) – Conventional Dynamic DCVSL Adder

In Conventional Dynamic DCVSL Adder, a clock is used whose Delay value is 10ns. Rest apart from the DCVSL structure, the remaining portion of the schematic are exact and also the values of the transistors are the same.

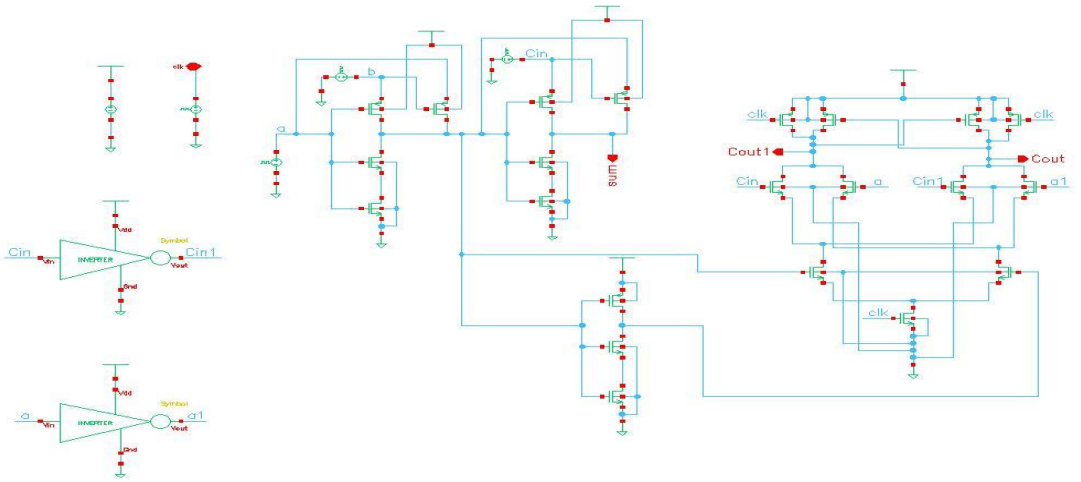

Figure 49(b) – Proposed Dynamic DCVSL Adder

In this circuit, stacking is used. The first XOR gate having the input ‘b’ connected to the source of one PMOS and to the gate of another PMOS, has the (W/L) ratio of both the PMOS as (1200nm/120nm) and (1800/120nm), i.e. 10:1 and 15:1, respectively and NMOS has 120nm/120nm, i.e. 1:1 in ratio. Same way, the input ‘C_{in}’ has both of its PMOS in the ratio 10:1 and 15:1 respectively and NMOS as 1:1. The below

stacking unit has PMOS in the ratio (W/L) 180nm/120nm, i.e. 1.5/1 and the NMOS's (W/L) ratio is 120nm/120nm, i.e. 1:1.

The transient responses are shown below –

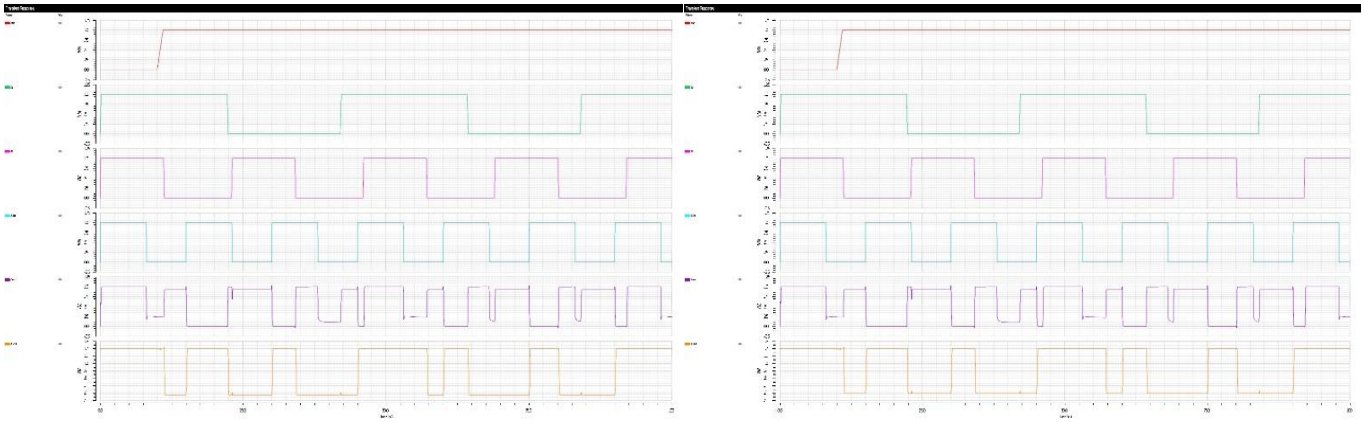

Figure 50(a) and 50(b) – Conventional (1V) and (1.1V)

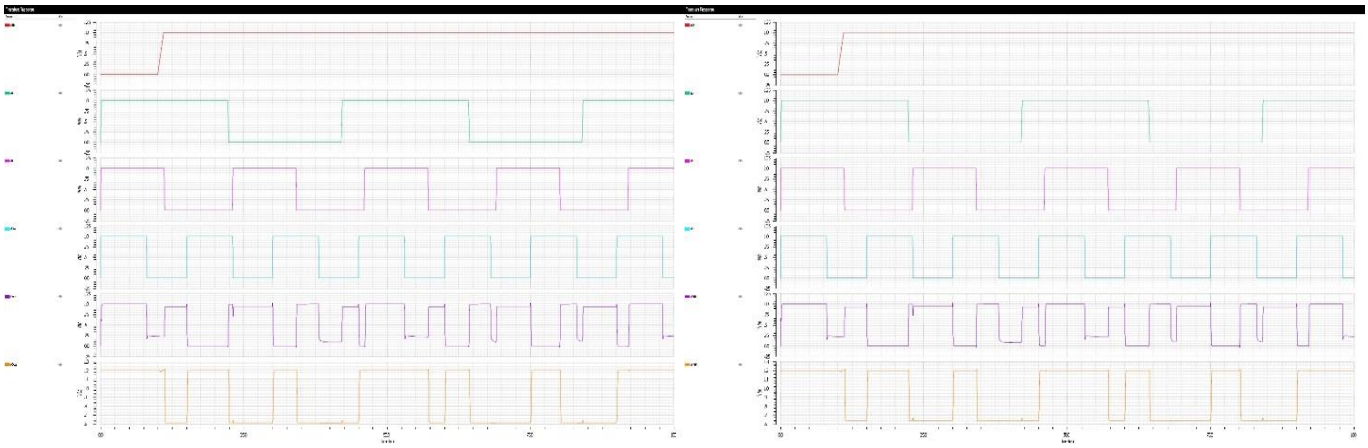

Figure 50(c) and 50(d) – Conventional (1.2V) and (1.3V)

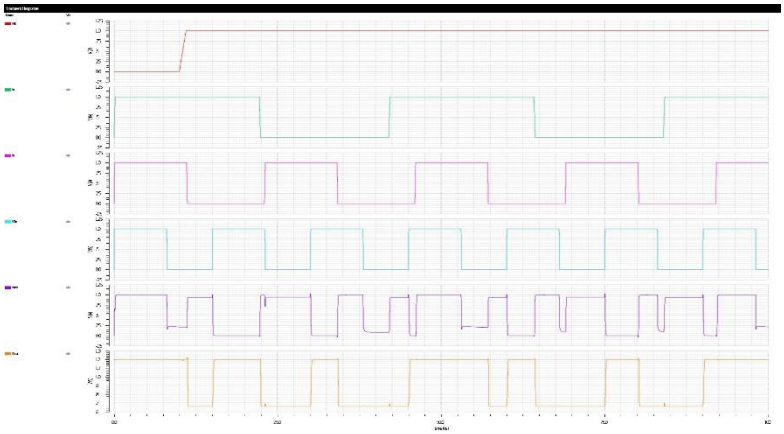

Figure 50(e) – Conventional (1.4V)

Now, we show the transient response of the proposed circuit.

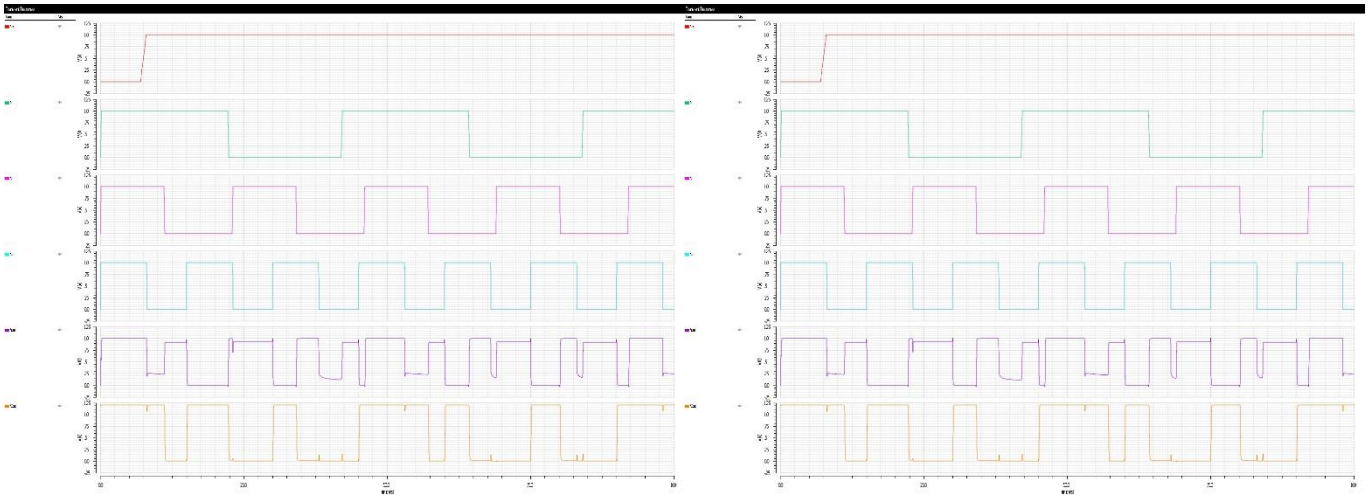

Figure 51(a) and 51(b) – Proposed (1V) and (1.1V)

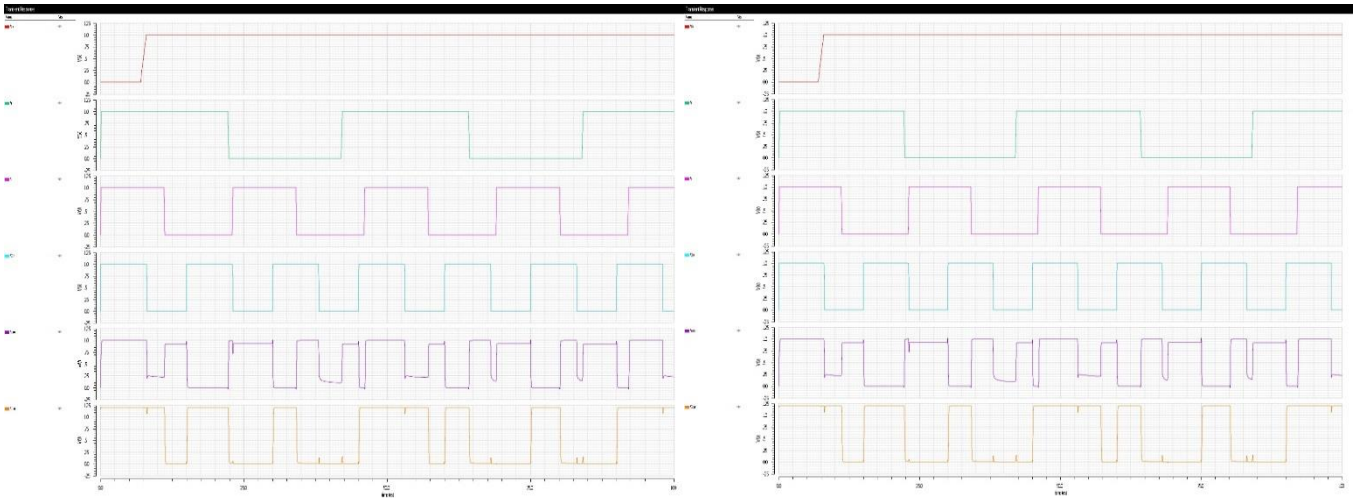

Figure 51(c) and 51(d) – (1.2V) and (1.3V)

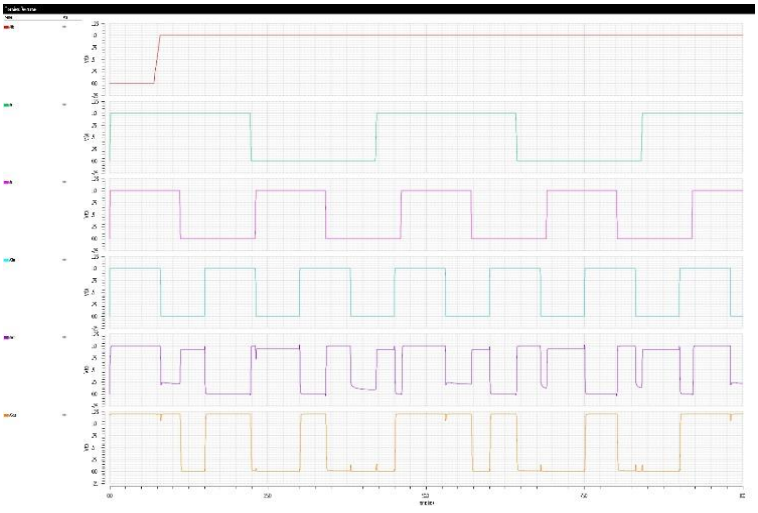

Figure 51(e) – Proposed (1.4V)

Now, the power consumption is calculated keeping the voltage constant at 1.2V.

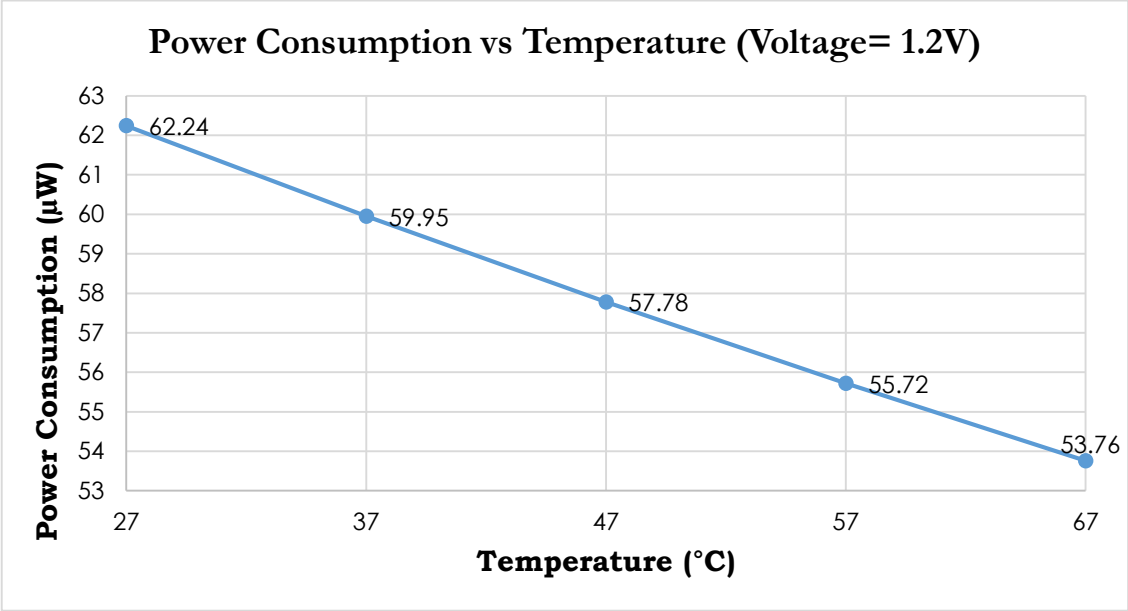

Figure 52(a) – Conventional Power Consumption vs Temperature

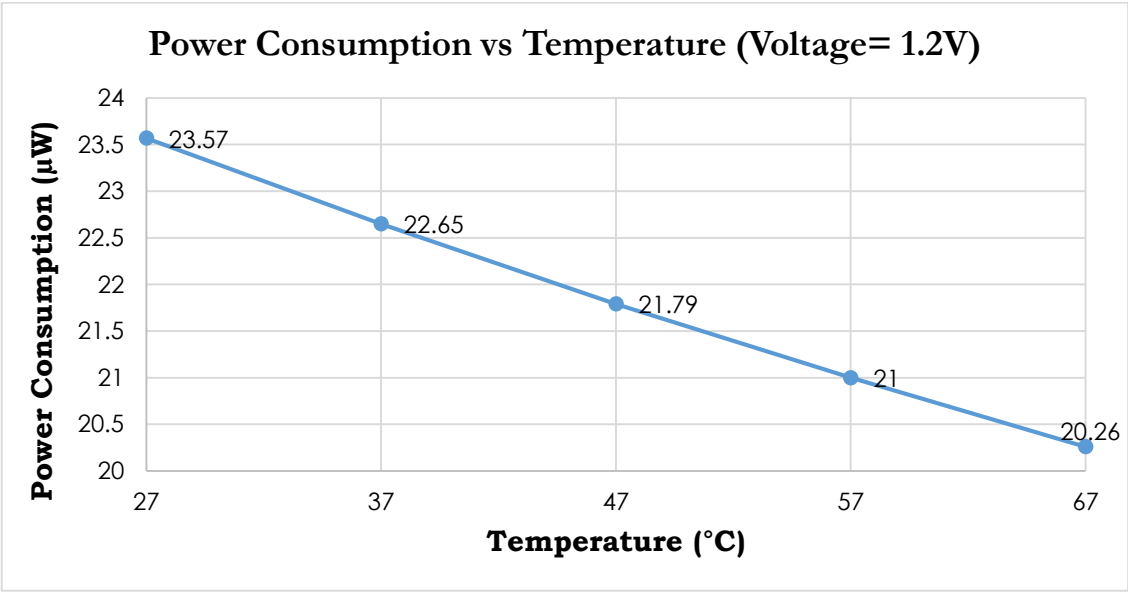

Figure 52(b) – Proposed Power Consumption vs Temperature

The delay is shown below –

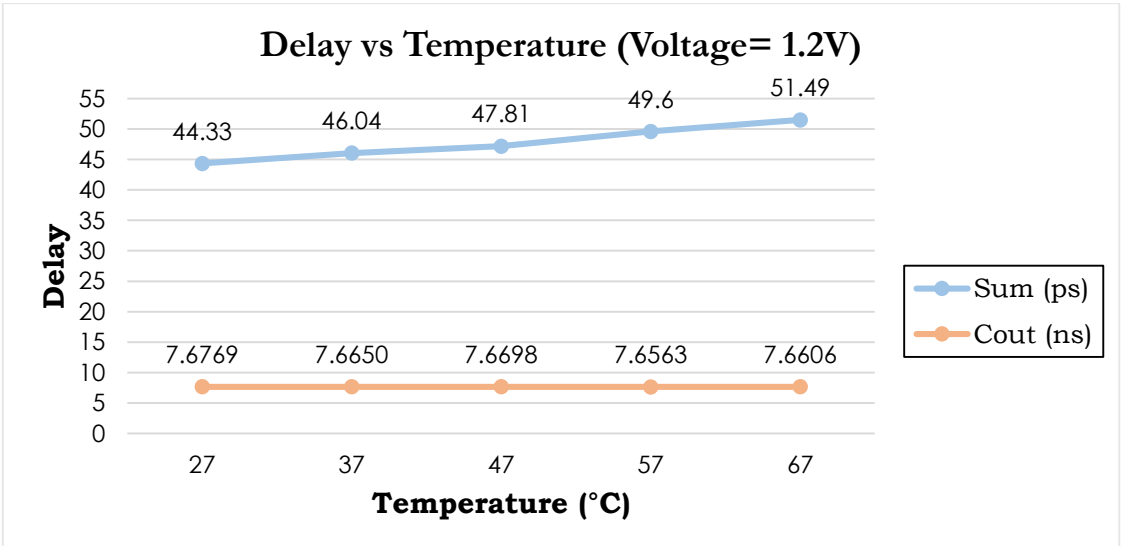

Figure 53(a) – Conventional Delay vs Temperature

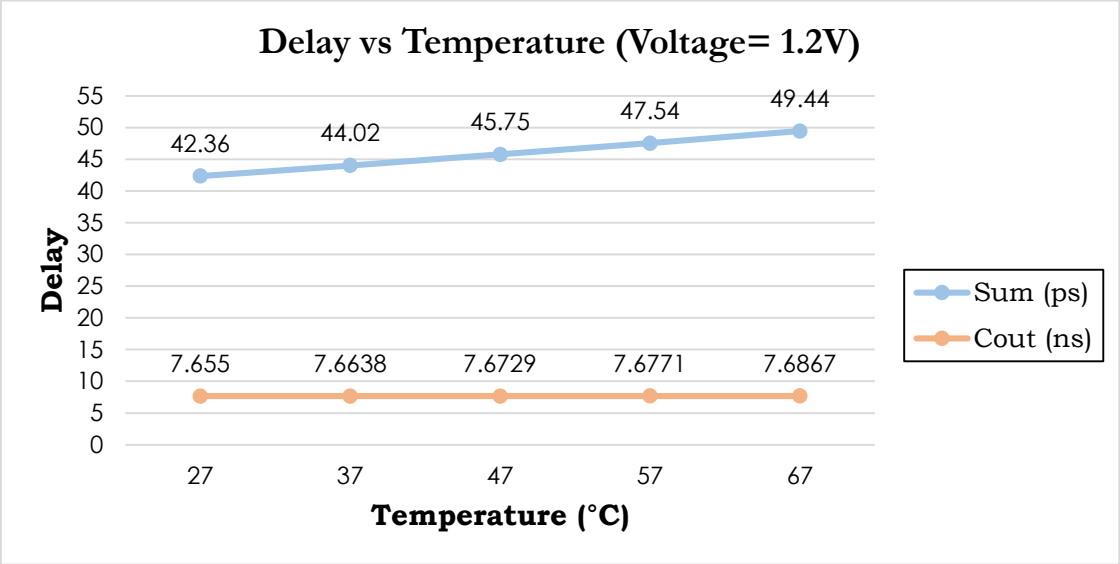

Figure 53(b) – Proposed Delay vs Temperature

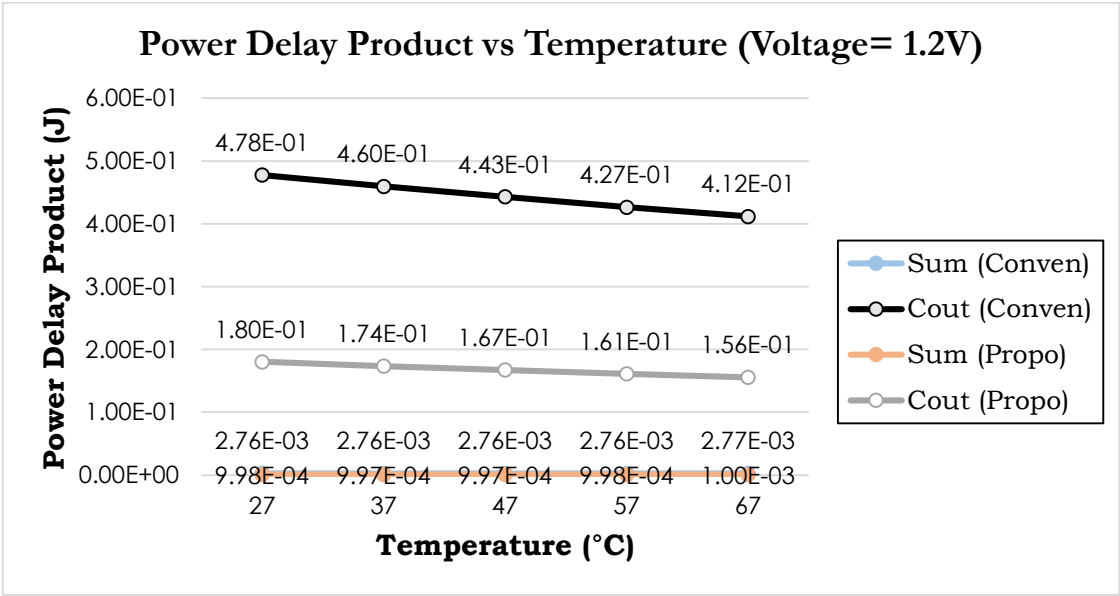

Figure 54 - Power Delay Product vs Voltage (Temperature= 27C)

➢ (6.3) Modified DCVSL Adder –

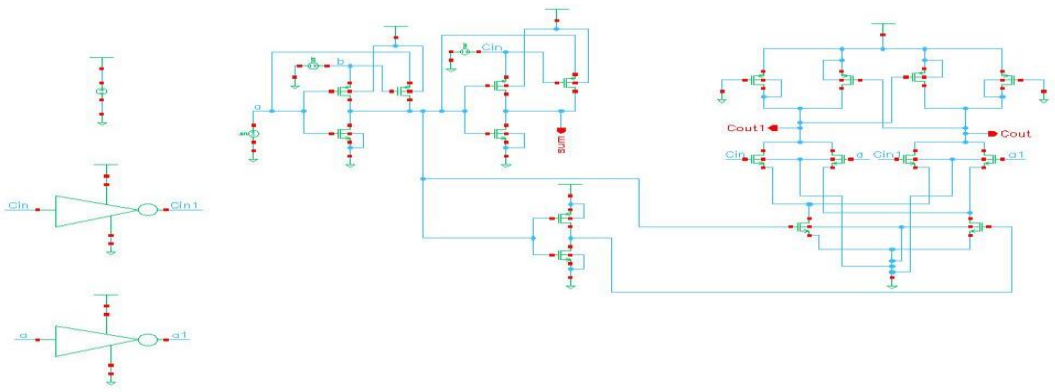

Figure 55(a) – Conventional Modified DCVSL Adder

The values of all the transistors are in the same format as the Dynamic DCVSL circuit.

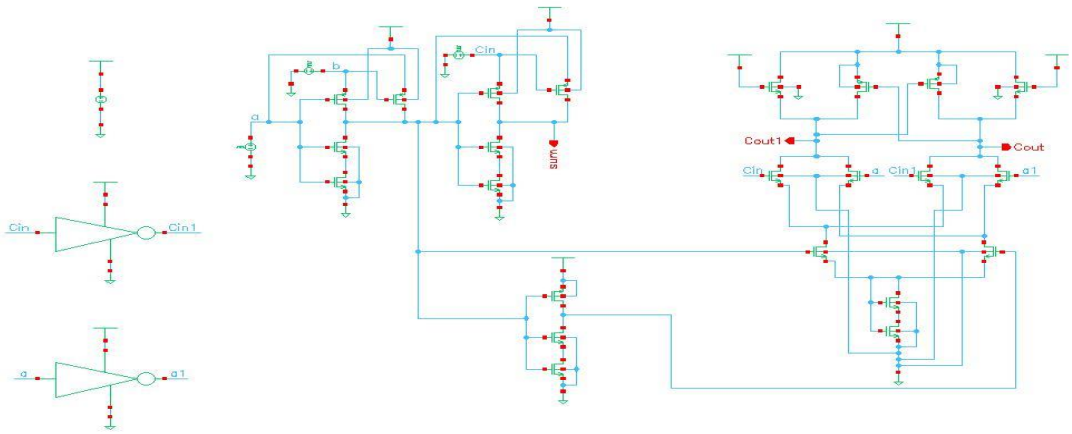

Figure 55(b) – Proposed Modified DCVSL Adder

Here, stacking is used. The extra NMOS transistor added to the stacking effect has its (W/L) ratio as 1:1 since it is 120nm/120nm. The connection along the input 'b' and 'C_{in}' has its transistor's value same as the conventional circuit. But, the proposed DCVSL portion of this Adder circuit has PMOS (W/L) ratio as 1:1 since it is 120nm/120nm and for NMOS it is 1:5 since it is 120nm/600nm.

The transient responses are –

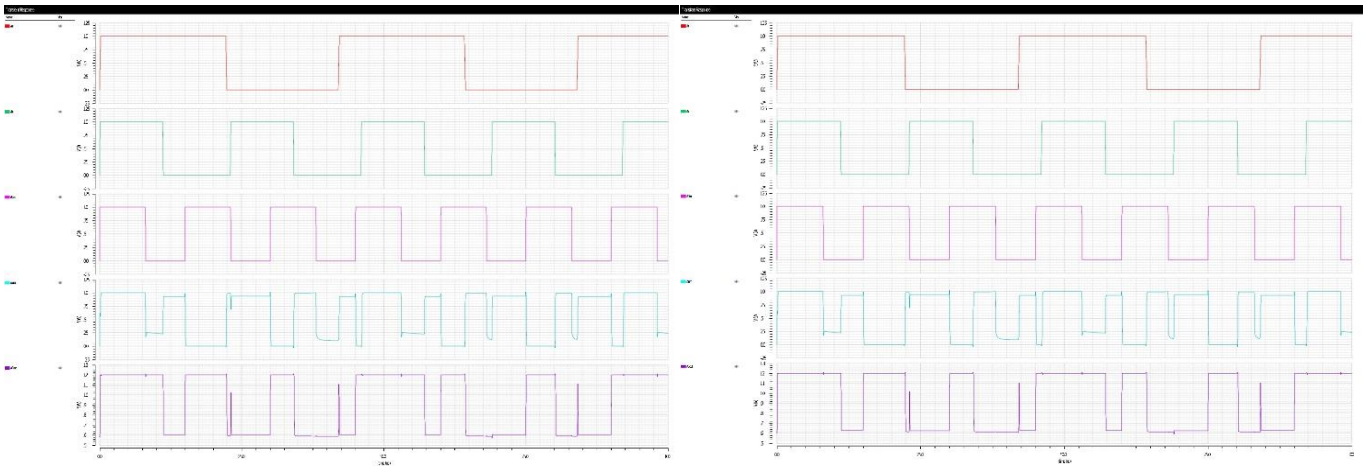

Figure 56(a) and 56(b) – Conventional (1V) and (1.1V)

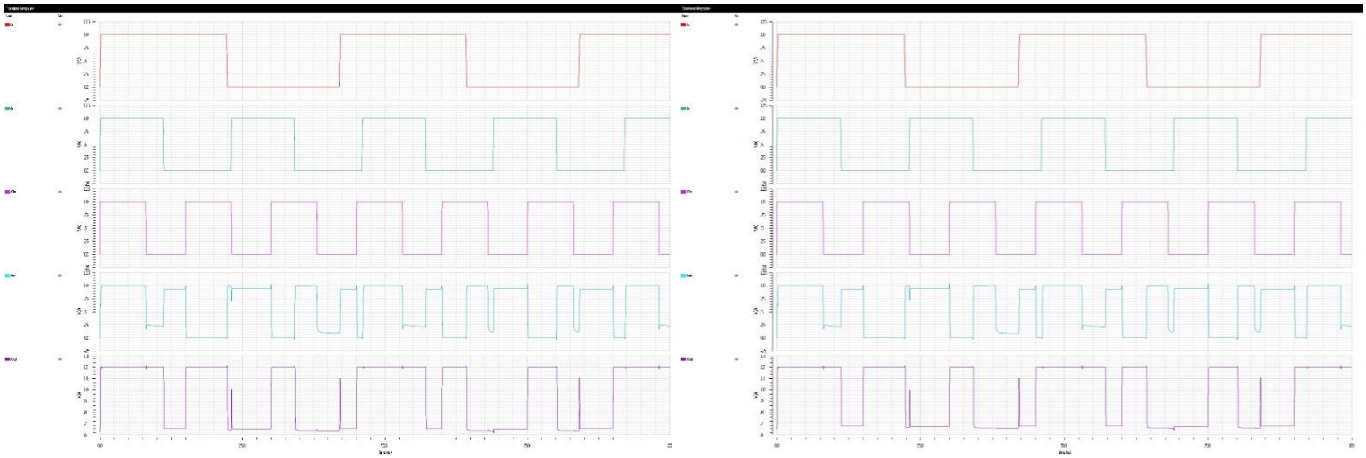

Figure 56(c) and 56(d) – Conventional (1.2V) and (1.3V)

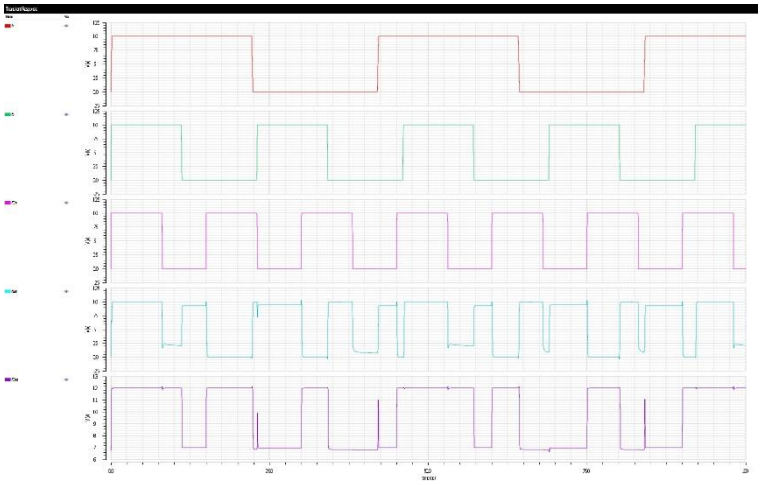

Figure 56(e) – Conventional (1.4V)

Here goes the proposed transient responses –

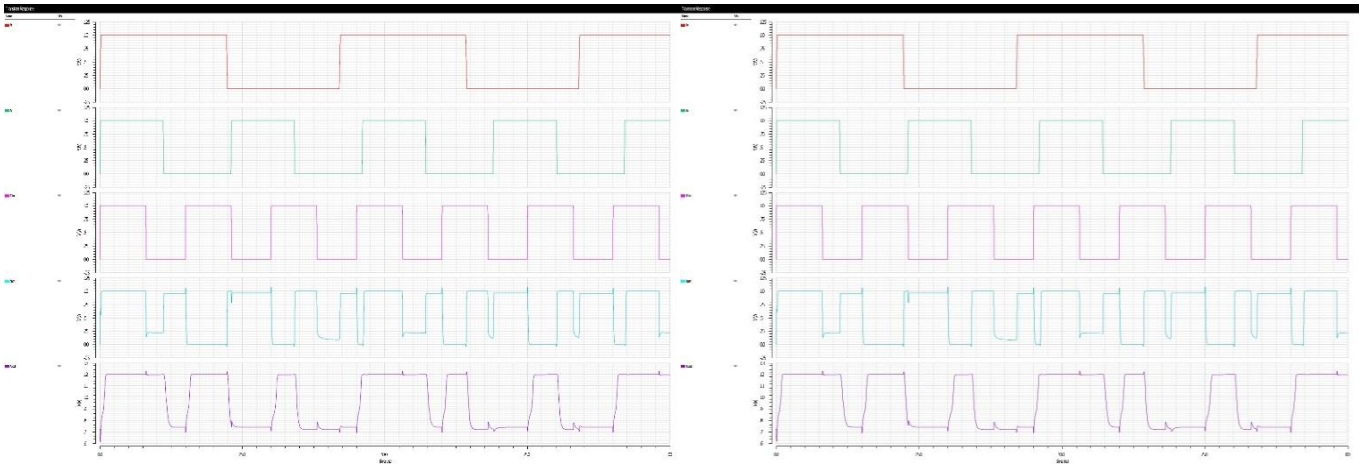

Figure 57(a) and 57(b) – Proposed (1V) and (1.1V)

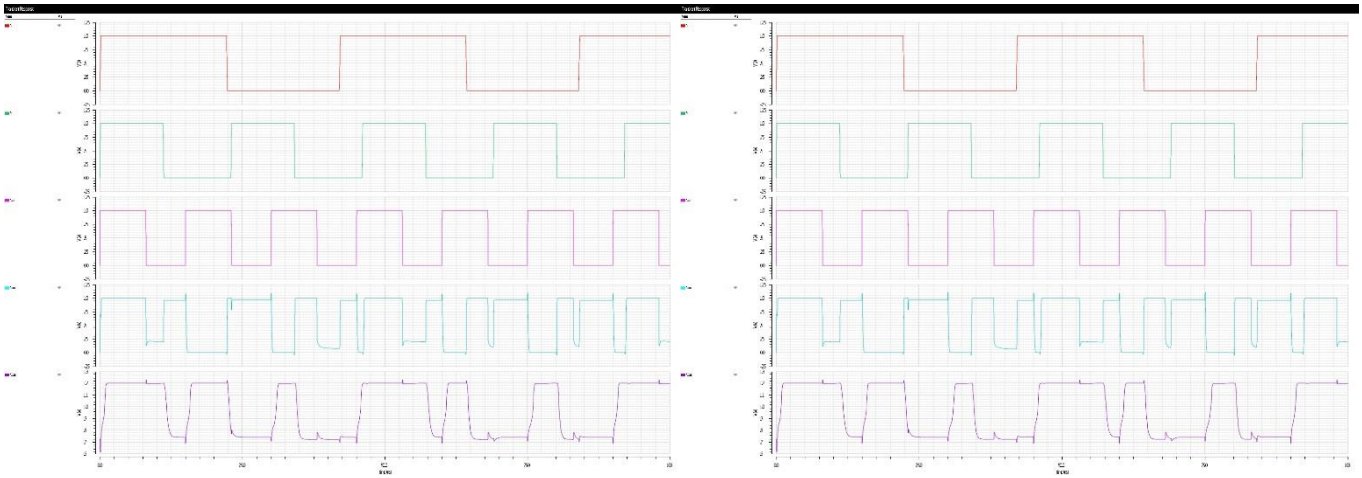

Figure 57(c) and 57(d) – Proposed (1.2V) and (1.3V)

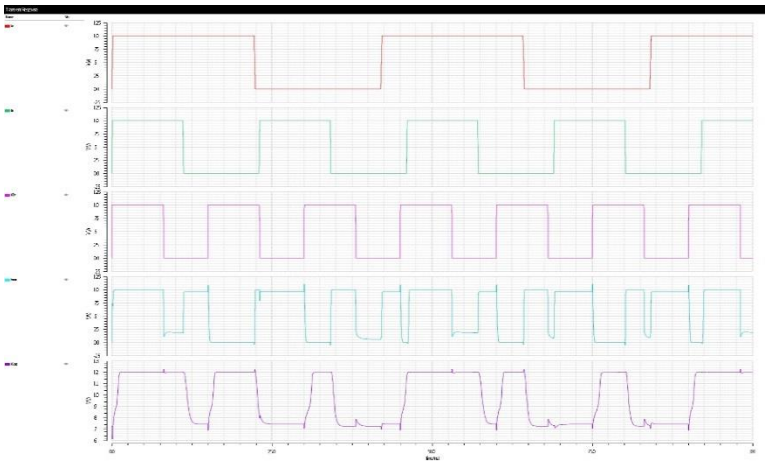

Figure 57(e) – Proposed (1.4V)

Here are the results of power consumption versus temperature keeping the voltage constant at 1.2V.

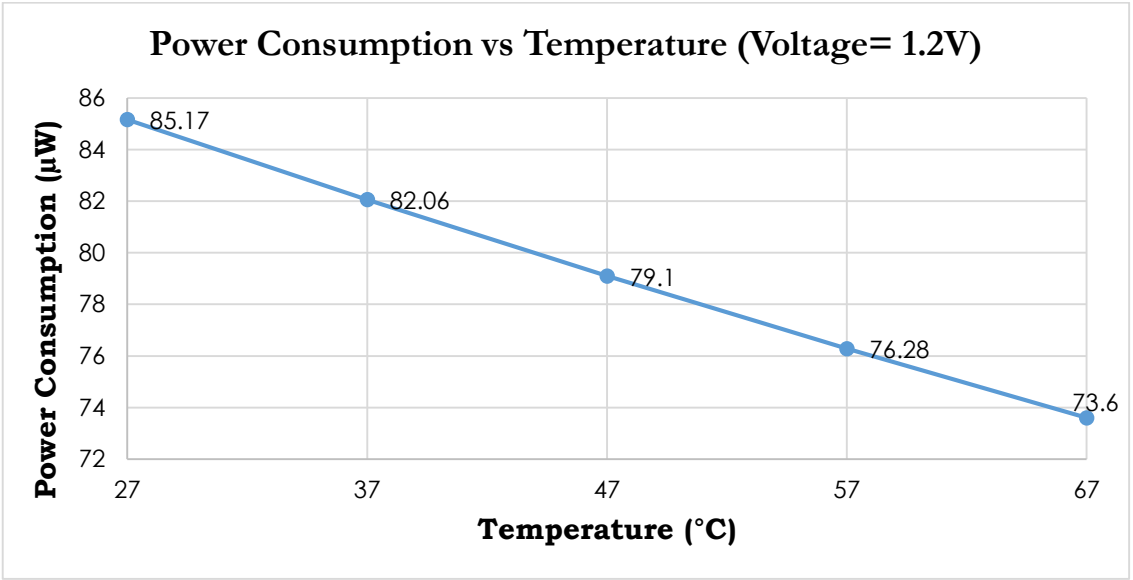

Figure 58(a) – Conventional Power Consumption vs Temperature

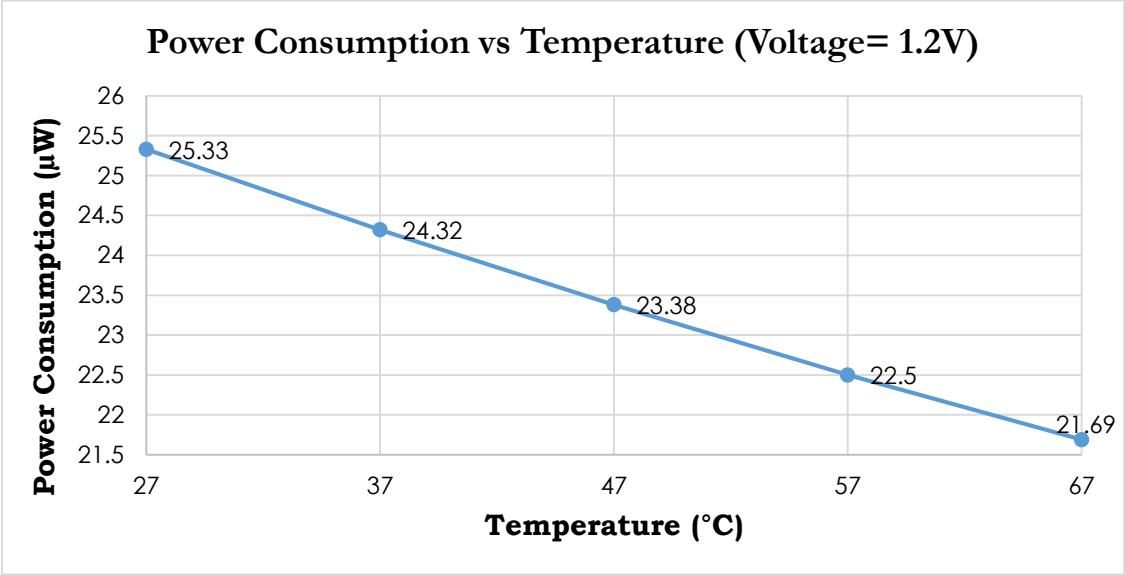

Figure 58(b) – Proposed Power Consumption vs Temperature

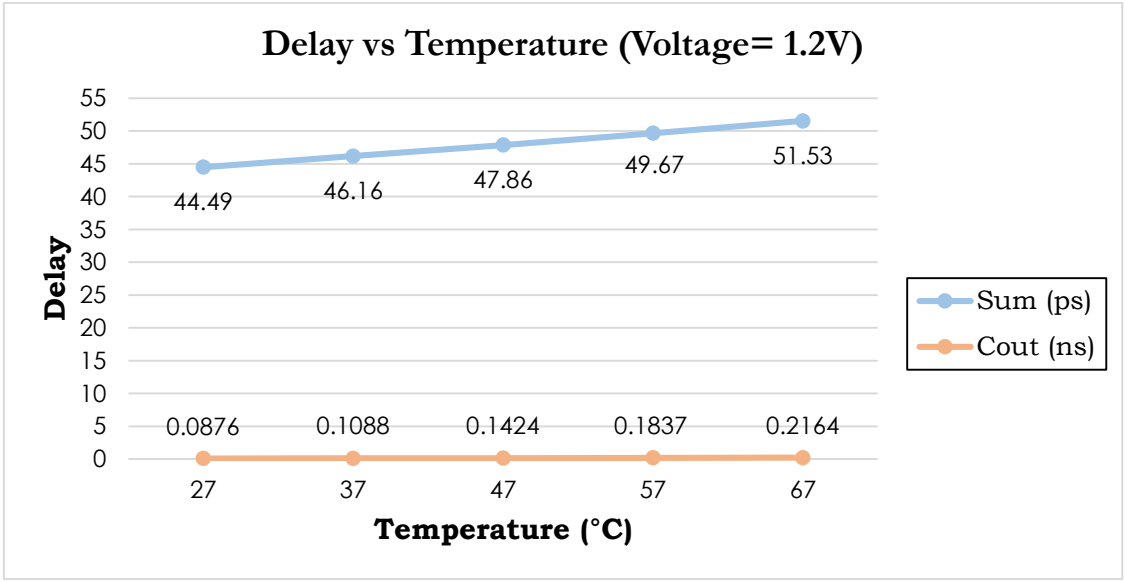

Figure 59(a) – Conventional Delay vs Temperature

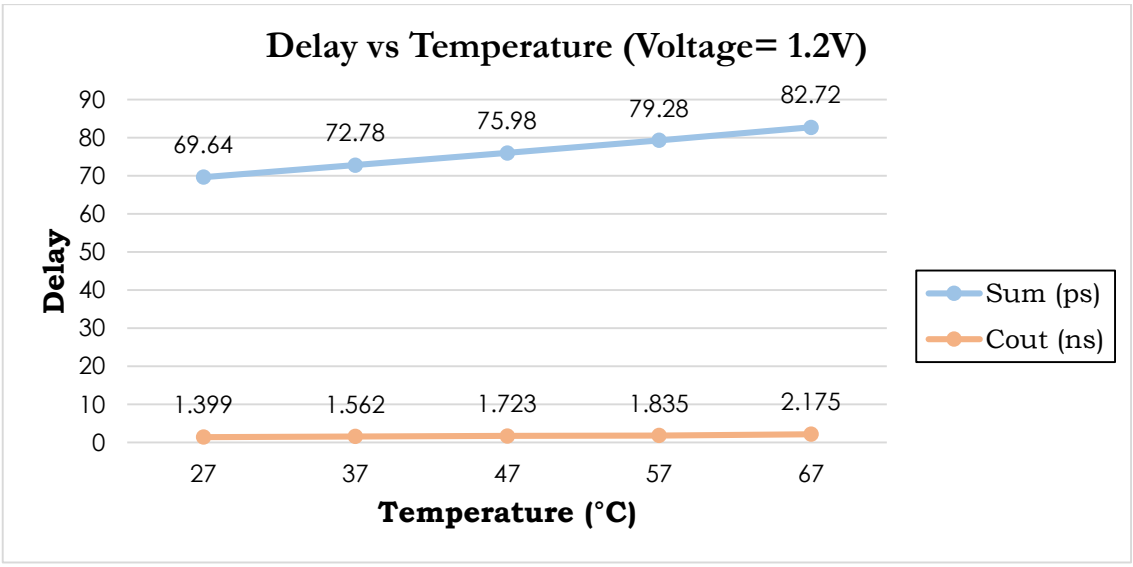

Figure 59(b) – Proposed Delay vs Temperature

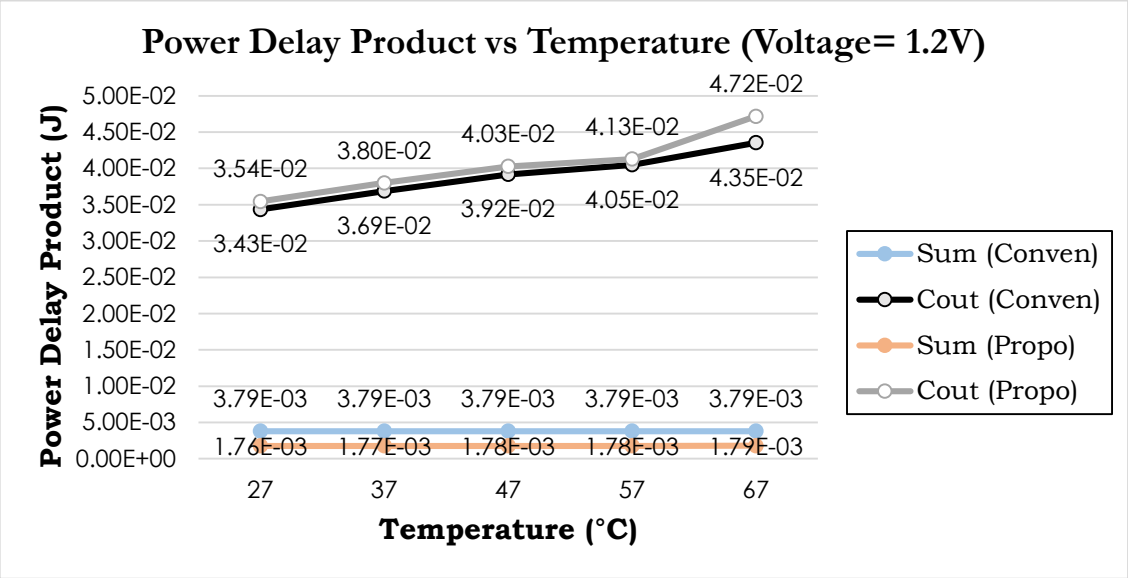

Figure 60 – Power Delay Product vs Temperature

Temperature	Power Consumption (µW)					
	Static DCVSL		Dynamic DCVSL		Modified DCVSL	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
27	41.02	24.45	62.24	23.57	85.17	25.33
37	39.46	23.5	59.95	22.65	82.06	24.32
47	37.99	22.61	57.78	21.79	79.10	23.38
57	36.61	21.78	55.72	21.00	76.28	22.50
67	35.31	21.01	53.76	20.26	73.60	21.69

Table 11 – Power Consumption vs Temperature (1.2V)

Temperature	Delay (ns) for Sum					
	Static DCVSL		Dynamic DCVSL		Modified DCVSL	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
27	0.04364	0.06874	0.04433	0.04236	0.04449	0.04821
37	0.04525	0.07174	0.04604	0.04402	0.04616	0.05028
47	0.04697	0.07495	0.04781	0.04575	0.04786	0.05188
57	0.04867	0.07821	0.04960	0.04754	0.04967	0.05379
67	0.05050	0.08151	0.05196	0.04944	0.05153	0.05565

Table 12 – Delay vs Temperature for Sum (Voltage= 1.2V)

Temperature	Delay (ns) for Cout					
	Static DCVSL		Dynamic DCVSL		Modified DCVSL	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
27	0.15785	0.16430	7.6769	7.6550	0.0876	1.399
37	0.16620	0.17055	7.6650	7.6638	0.1088	1.562
47	0.17545	0.17705	7.6698	7.6729	0.1424	1.723
57	0.18530	0.18340	7.6563	7.6771	0.1837	1.835
67	0.19560	0.19030	7.6606	7.6867	0.2164	2.175

Table 13 – Delay vs Temperature for Cout (Voltage= 1.2V)

Temperature	Power Delay Product (Joules) for Sum					
	Static DCVSL		Dynamic DCVSL		Modified DCVSL	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
27	1.79E-03	1.68E-03	2.76E-03	9.98E-04	3.79E-03	1.76E-03
37	1.79E-03	1.69E-03	2.76E-03	9.97E-04	3.79E-03	1.77E-03
47	1.78E-03	1.69E-03	2.76E-03	9.97E-04	3.79E-03	1.78E-03
57	1.78E-03	1.70E-03	2.76E-03	9.98E-04	3.79E-03	1.78E-03
67	1.78E-03	1.71E-03	2.77E-03	1.00E-04	3.79E-03	1.79E-03

Table 14 – Power Delay Product vs Temperature for Sum (Voltage= 1.2V)

Temperature	Power Delay Product (Joules) for Cout					
	Static DCVSL		Dynamic DCVSL		Modified DCVSL	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
27	6.48E-03	4.02E-03	4.78E-01	1.80E-01	3.43E-02	3.54E-02
37	6.56E-03	4.01E-03	4.60E-01	1.74E-01	3.69E-02	3.80E-02
47	6.67E-03	4.00E-03	4.43E-01	1.67E-01	3.92E-02	4.03E-02
57	6.78E-03	3.99E-03	4.27E-01	1.61E-01	4.05E-02	4.13E-02
67	6.92E-03	3.99E-03	4.12E-01	1.56E-01	4.35E-02	4.72E-02

Table 15 – Power Delay Product vs Temperature for Cout (Voltage= 1.2V)

The power consumption for all the three DCVSL Adder circuits are less in case of the proposed circuits than the conventional one.

For the delay of the output Sum, the Static DCVSL and the Modified DCVSL are having their values more in case of the proposed circuit than the conventional circuit. And, for the delay of the output Cout, the delays of all the proposed circuits are having more value than the conventional one.

In case of Power Delay Product, the PDPs of all the tree DCVSL structures for the output Sum, are having less value for the proposed circuit than the conventional one. And for the output Cout, except for the value of the Modified DCVSL, the rest two DCVSL structures are having less value in case of the proposed circuit.

Next, the layout of all the DCVSL adder structures are shown, which includes both the conventional and the proposed ones –

(6.4) Layouts –

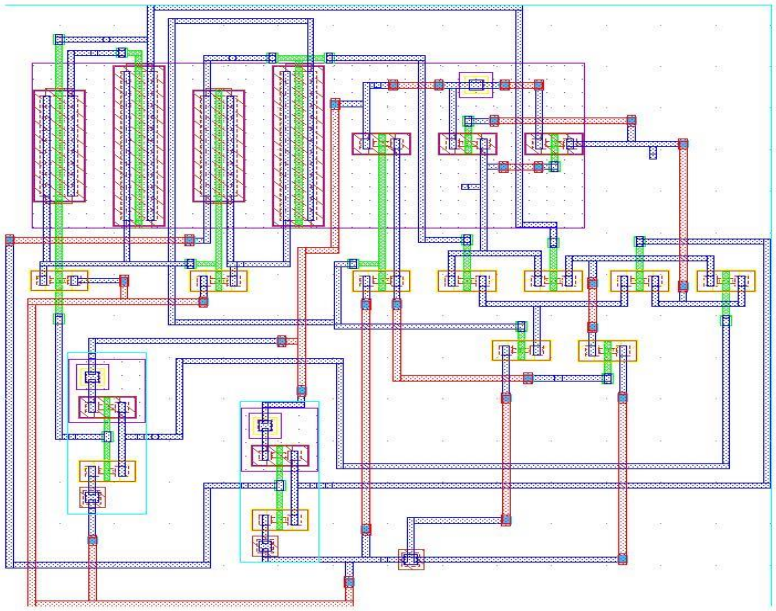

Figure 61 – Conventional Static DCVSL Adder

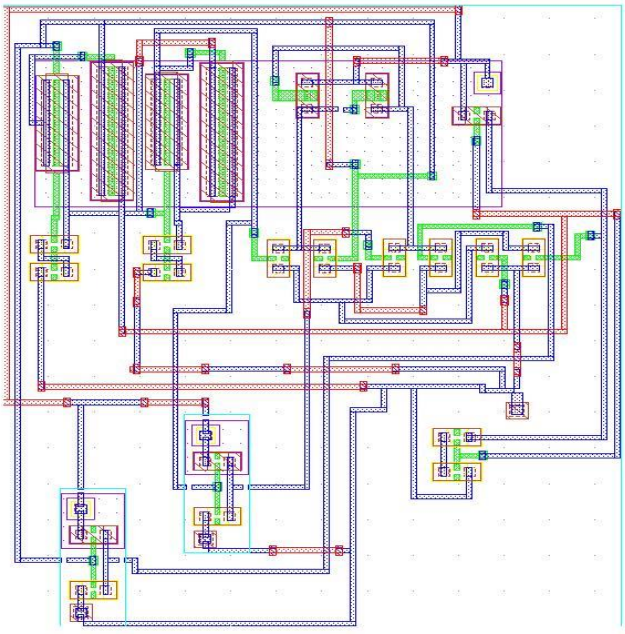

Figure 62 – Proposed Static DCVSL Adder

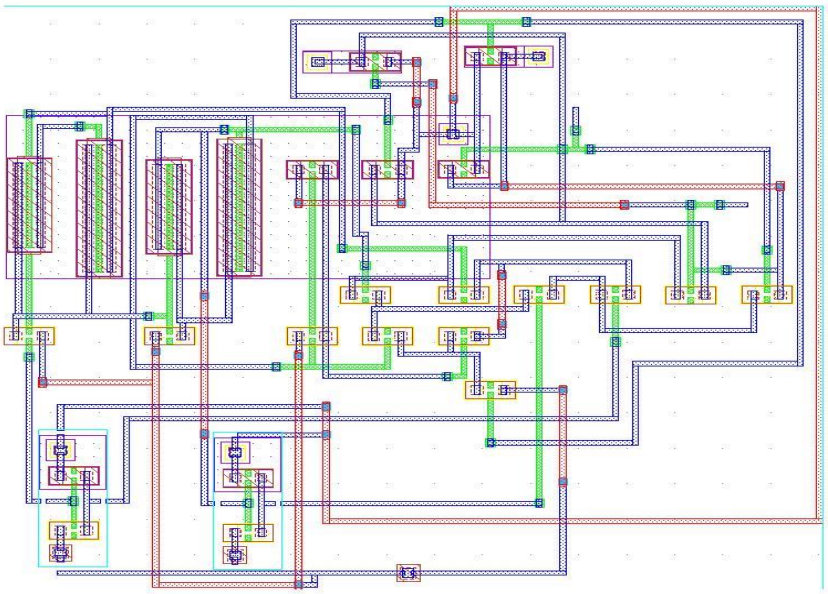

Figure 63 – Conventional Dynamic DCVSL Adder

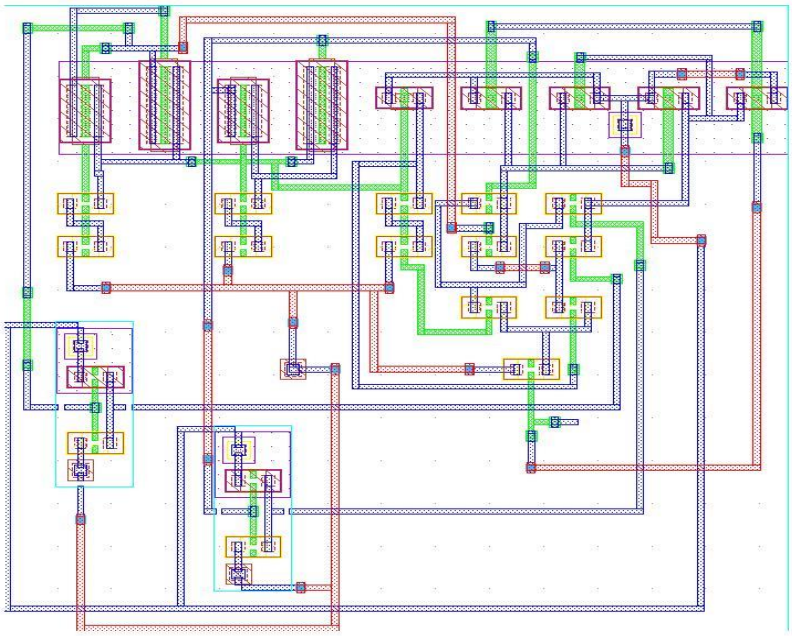

Figure 64 – Proposed Dynamic DCVSL Adder

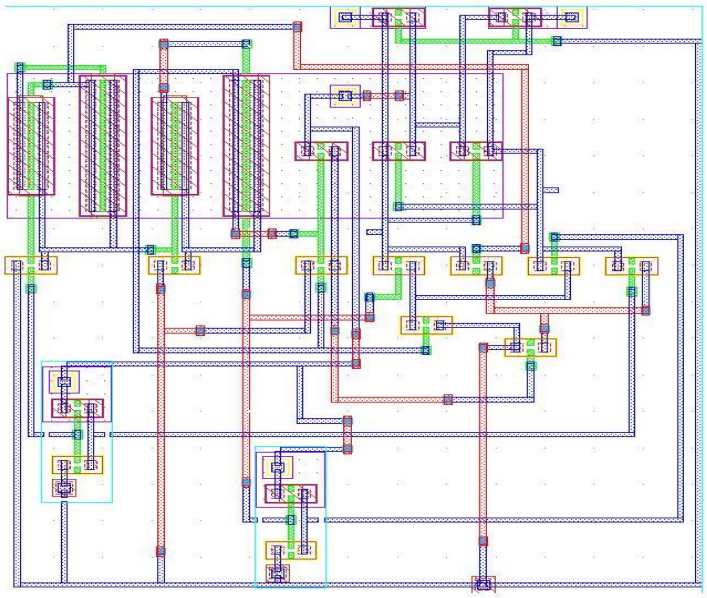

Figure 65 – Conventional Modified DCVSL Adder

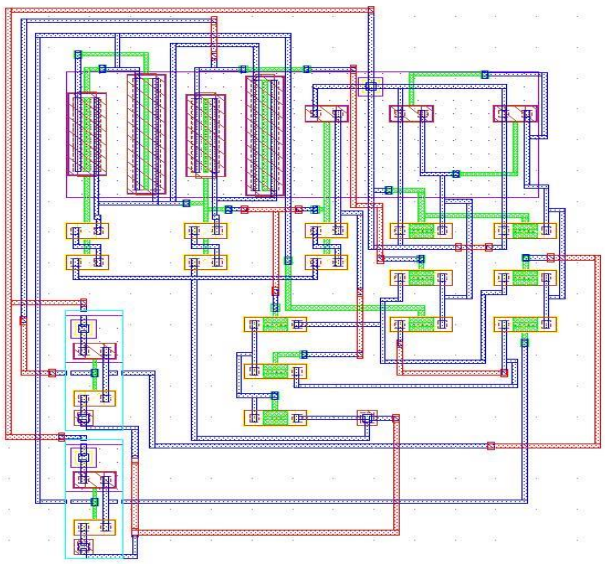

Figure 66 – Proposed Modified DCVSL Adder

	Static DCVSL Adder		Dynamic DCVSL Adder		Modified DCVSL Adder	
	Conventional	Proposed	Conventional	Proposed	Conventional	Proposed
Area (μm^2)	203.01	235.85	190.44	213.01	228.53	263.13
No. of Transistors	20	23	25	26	22	27

Table 16 – Comparison of Area and No. of Transistors for the Adder Circuits

As you can see that the number of transistors are increased a bit for the proposed circuits, so the area which is achieved from the layout is also a bit more for the proposed circuits.

Conclusion

For low-leakage and high-speed circuit, the important two factors are speed and power. However, the main trade-off is that; when someone goes for speed, the power is degraded. And in the next case, when the power consumption is improved, the delay is more in that case. Therefore, we go for the power delay product, which best determines the efficient circuit combining the two parameters, keeping other factors such as voltage and temperature.

When the power consumption is considered, along the temperature, we find out that all the three DCVSL structures produces better result in case of the proposed circuit than the conventional one and the best among them being the Static DCVSL; whereas when the delay is measured along the various temperatures, the Dynamic DCVSL alone produces better result in case of the proposed circuit than the conventional one, and the rest of the other two DCVSL structures' delay is a bit more for the proposed circuit than the conventional one (i.e., Static DCVSL and Modified DCVSL).

Now, considering the power delay product for these three structures, we find out that the PDP is less for the proposed circuits in case of the Static and Dynamic DCVSL, whereas for the Modified DCVSL, it is more for the proposed circuit.

Coming to the DCVSL Adder circuits which is implemented using the previous three DCVSL structures, we find out that the power consumption is less in case of the proposed circuit for all of these Adder structures. As this is an adder circuit, therefore it has two outputs, i.e. Sum and Cout. So, for delay, it is calculated separately. And from the analysis, we find out that the delay for Sum is less in case of the Dynamic DCSVL Adder than the other two DCVSL structures. For Cout, the values are more for all of the three DCVSL structures, where the Dynamic DCVSL is having the highest.

In calculation of the Power Delay Product (PDP) for all of these Adder circuits, it is found out that the proposed circuits of all of them are having less value than the conventional one and the Dynamic DCVSL adder is having the least among them, for the output Sum. For the output Cout, except for the Modified DCVSL adder, the rest of the other two DCVSL adders are having less value in the proposed circuit than the conventional one. And, among them, the Static DCVSL is having lesser value than the Dynamic one which determines better PDP for Cout, in this case.

The layouts of all the Adder circuits are also done considering both the conventional and the proposed ones. An analysis is done with the parameters area and number of transistors, which shows that the area is least for the Dynamic DCVSL Adder than the rest and the number of transistors is least for the Static DCVSL than the rest two.

So considering all the scenarios, we cannot specify a particular DCVSL structure to be the best as for different parameters, the result is indeed different, taking all the previous analysis. Depending upon the

requirement, we may use the particular DCVSL structure, which best suits the situation, i.e. the particular parameter providing the least value for it. For some cases, the Static DCVSL may be the best option and for other cases, one among the other two may be the best option.

This completes the thesis work.

Bibliography

[1] - L. G. Heller and W. R. Griffin, “Cascode voltage switch logic: A differential CMOS logic family,” in ISSCC Dig. Tech. Papers, 1984, pp. 16-17.

[2] - R. K. Montoye, “Testing scheme for differential cascode voltage switch circuits,” IBM Tech. Disc. Bull., vol. 27, pp. 6148-6152, 1985.

[3] - C. K. Erdelyi, “Random logic design utilizing single-ended cascode voltage switch circuits in NMOS,” IEEE J. Solid-State Circuits, vol. SC-20, pp. 591-594, Apr.1985.

[4] - K. M. Chu, D. L. Pulfrey, "A Comparison of CMOS Circuit Techniques: Differential Cascode Voltage Switch Logic Versus Conventional Logic", IEEE J. Solid-State Circuits, v01.22, pp.528-532, August 1987.

[5] - R. K. Brayton and C. McMullen, “The decomposition and factorization of Boolean expressions,” m Proc. IEEE Int. Symp. Circuits Syst. (Rome, Italy), 1982, pp. 49-54.

[6] - S. Muroga, Logic Design and Switching Theory. New York: Wiley, 1979, pp. 163-180.

[7] - J. M. Rabaey, A. Chandrakasan, B. Nicolic, “Digital Integrated Circuits,” 2nd Edition, Prentice-Hall, 2003.

[8] - M. Renaudin and B. E. Hassan, “The Design of Fast Asynchronous Adder Structures and Their Implementation Using DCVSL Logic,” Proceedings International Symposium on Circuits and Systems, 1994.

[9] - K. Chu and D. Pulfery, “Design Procedures for Differential Cascade Voltage Switch Circuits,” IEEE Journal of Solid-State Circuits, vol. 21, pp. 1082-1087, December 1986.

[10] - K. Chu and D. Pulfery, “A Comparison of CMOS Circuit Techniques: Differential Cascode Voltage Switch Logic versus Conventional Logic,” IEEE Journal of Solid-State Circuits, vol. 22, pp. 528-532, August 1987.

[11] - M. Shams, “Modelling and Optimization of CMOS Logic Circuits with Application to Asynchronous Design,” Ph.D. Thesis, University of Waterloo, 1999.

[12] - M. Shams, M. Elmasry, “A Formulation for Quick Evaluation and Optimization of Digital CMOS Circuits,” in IEEE International Symposium on Circuits and Systems, ISCAS-99, June 1999.

[13] - A. Bellaouar, Mohamed I. Elmasry, “Low-power digital VLSI design: circuits and systems”, 2nd Edition.

[14] - Kang, Sung-Mo, Leblebici, Yusuf, “CMOS Digital Integrated Circuits Analysis and Design”, McGraw-Hill International Editions, Boston, 2nd Edition, 1999.

[15] - T. Sakurai and R. Newton, “Alpha-power Law MOSFET Model and its Application to CMOS Inverter Delay and Other Formulas,” IEEE Journal of Solid-State Circuits, vol.25, pp. 584-594, April 1990.

[16] - Krambeck, R.H., Lee, C.M. and Law, H.S, “High-speed Compact Circuits with CMOS,” IEEE J. Solid State Circuits, Vol. SC-17, No. 3; June, 1982.

[17] - D. Z. Turker,, S. P. Khatri , “A DCVSL Delay Cell for Fast Low Power Frequency Synthesis Applications”, IEEE transactions on circuits and systems, June 2011, vol. 58, no. 6, pp. 1125-1138.

[18] - F-s Lai and W Hwang, “Design and Implementation of Differential Cascode Voltage Switch with Pass-Gate (DCVSPG) Logic for High-Performance Digital Systems” IEEE journal of solid-state circuits, vol. 32, no. 4, April 1997, pp. 563-573.

[19] - P. K. Lala and A. Walker, “A Fine Grain Configurable Logic Block for Self-checking FPGAs”, VLSI Design 2001, Vol. 12, No. 4, pp. 527-536.

[20] - Jan M. Rabaey, Digital Integrated Circuits; a design prospective, Upper Saddle River: Prentice-Hall, 1996.

[21] - Ila Gupta, Neha Arora, Prof. B. P. Singh, “Simulation and Analysis of 2:1 Multiplexer Circuits at 90nm Technology” in International Journal of Modern Engineering Research, Vol.1, Issue.2, pp-642-647, ISSN: 2249-6645, 2011.

[22] - Ila Gupta, Neha Arora, Prof. B. P. Singh, “Analysis of Several 2:1 Multiplexer Circuits at 90nm and 45nm Technologies” International Journal of Scientific and Research Publications, Vol.2, Issue.2, February 2012, ISSN: 2250-3153.

[23] - Ila Gupta, Neha Arora, Prof. B. P. Singh, “Design and Analysis of 2:1 Multiplexer for High Performance Digital Systems" International Journal on Electronics & Communication Technology (IJECT) Vol.3, Issue1, Jan- March 2012, pp-183-186, ISSN: 2230-7109 (Online) ISSN: 2230-9543 (Print).

[24] - M. Schlag, E. J. Yoffa, P. S. Hauge and C. K. Wong, "A Method for Improving Cascode-Switch Macro Wirability", IEEE Trans. on CAD, vol.CAD-4, 150-5, 1985.

[25] - T. C. Lo, "Regular Layout FET Carry Look-Ahead Circuit", IBM Technical Disclosure Bulletin, vol. 26, 6202-8, 1984.

[26] - T. C. Lo, "LSSD Implemented with DCVS Logic," IBM Technical Disclosure Bulletin, vol.26, 5805-10, 1984.

[27] - V. De, et al, "Techniques for leakage power reduction," in Design of High-Performance Microprocessor Circuits, ed. A. Chandrakasan, W. J. Bowhill, and F. Fox, IEEE Press, Piscataway NJ, pp. 46-62, 2000.

[28] - Y. Ye, S. Borkar and V. De, "New Technique for standby leakage reduction in high-performance circuits," IEEE Symposium on VLSI Circuits, pp. 40-41, 1998.