

PERFORMANCE ANALYSIS OF DIFFERENTIAL CMOS CIRCUITS



A thesis submitted to the
Department of Electrical and Electronic Engineering
BUET, Dhaka,
in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering
(Electrical and Electronic)

MD. ATAUR RAHMAN PATWARY



December 1995

The thesis titled "Performance Analysis of Differential CMOS Circuits" submitted by Md. Ataur Rahman Patwary, Roll No. 921309P, Session 90-91-92 to the Department of Electrical and Electronic Engineering, BUET has been accepted as satisfactory for partial fulfillment of the requirements for the degree of Master of Science in Engineering (Electrical and Electronic).

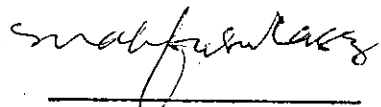
BOARD OF EXAMINERS

1. Mahfuzul Aziz 12 Dec 1995
(Dr. Syed Mahfuzul Aziz)
Associate Professor
Department of Electrical and
Electronic Engineering
BUET, Dhaka.
Chairman
(Supervisor)
2. A.B.M. Siddique Hossain 12/12/95
(Dr. A.B.M. Siddique Hossain)
Professor and Head,
Department of Electrical and
Electronic Engineering
BUET, Dhaka.
Member
(Ex-officio)
3. P. Kanai Saha 12/12/95
(Dr. Pran Kanai Saha)
Assistant Professor
Department of Electrical and
Electronic Engineering
BUET, Dhaka.
Member
4. M. Quamruzzaman 12/12/95
(Mr. M. Quamruzzaman)
Chief Engineer
Institute of Electronics
Atomic Energy Research Establishment
Savar, Dhaka.
Member
(External)

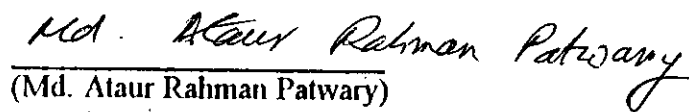
DECLARATION

I hereby declare that this work has been done by me and it has not been submitted elsewhere for the award of any degree or diploma.

Countersigned



(Dr. Syed Mahfuzul Aziz)
Supervisor



(Md. Ataur Rahman Patwary)

Acknowledgment

It is a matter of great pleasure on the part of the author to acknowledge his profound gratitude to his Supervisor, Dr. Syed Mahfuzul Aziz, Associate Professor of the Department of Electrical and Electronic Engineering, BUET for his valuable guidance, constant encouragement and supervision throughout the progress of the work.

The author also wishes to express his thanks and gratitude to Dr. A. B. M. Siddique Hossain, Professor and Head of the Department of Electrical and Electronic Engineering, BUET for his all-out support.

Finally the author would like to express thanks to all his friends and colleagues and staff of the Department of Electrical and Electronic Engineering for their constant support and assistance.

Abstract

CMOS is one of the leading VLSI technologies today. It is being used to implement high performance circuits in VLSI. Conventional static CMOS logic is attractive because of its extremely low quiescent power dissipation. This makes its power-delay product favorable compared to those of other technologies, viz., bipolar and nMOS technology. Although, the advancement of integrated circuit technology has now made it possible to fabricate devices with sub-micron dimensions thereby leading to very high speed CMOS circuits, the speed of CMOS devices is still lower than its nMOS counterpart.

Differential Cascode voltage Switch (DCVS) logic and Differential Split-Level (DSL) CMOS logic were introduced for speed improvement in CMOS circuits. However, these CMOS circuit techniques have not been used so far to design real VLSI chips owing to some inherent problems. This thesis examines the performance of the differential CMOS circuits compared to conventional static CMOS with a view to determine their suitability for VLSI implementation. The results obtained show that static DCVS circuits are slower than conventional static CMOS while DSL circuits are faster at optimum reference voltage only when short channel logic n-transistors are used.

Table of Contents

Acknowledgment	iv
Abstract	v
List of Figures	viii
List of Tables	x
CHAPTER 1 Introduction	1
1.1 Aims	1
1.2 Literature Review	2
1.3 Organization of the Thesis	3
CHAPTER 2 CMOS Circuit Techniques	4
2.1 Introduction	4
2.2 Conventional Static CMOS Logic	4
2.3 Differential CMOS Logic	9
2.3.1 Static Differential Cascode Voltage Switch (DCVS) Logic	12
2.3.2 Clocked Differential Cascode Voltage Switch (DCVS) Logic	17
2.3.3 Differential Split-Level (DSL) CMOS Logic	17

CHAPTER 3 Design of CMOS Circuits	24
3.1 Introduction	24
3.2 Design of Conventional Static CMOS Circuits	24
3.2.1 Conventional Static CMOS Network Derivation	25
3.2.2 Device Sizing for Conventional Static CMOS Circuits	29
3.3 Design of Differential CMOS Circuits	35
3.3.1 Differential Logic Tree Design	36
3.3.2 Device Sizing for DSL Circuits	44
CHAPTER 4 Performance Analysis of Various CMOS Logic Families	45
4.1 Introduction	45
4.2 Test arrangement	45
4.3 Analysis of Conventional Static CMOS Logic	47
4.4 Analysis of Static Differential Cascode Voltage Switch (DCVS) Logic	50
4.5 Analysis of Differential Split-Level (DSL) Logic	52
4.6 Comparison of the CMOS Logic Families	62
CHAPTER 5 Conclusions and Recommendations	64
5.1 Conclusions	64
5.2 Future work	65
References	66
Appendices	68

List of Figures

Fig. 2.1	Structure of conventional static CMOS logic gates.....	5
Fig. 2.2	Conventional static CMOS circuit with load and driver networks replaced by digital switches.....	6
Fig. 2.3	Conventional static CMOS inverter.....	8
Fig. 2.4	Block diagram of differential CMOS circuit.....	10
Fig. 2.5	Differential CMOS circuit with decision trees replaced by digital switches.....	11
Fig. 2.6	Block diagram of static DCVS circuit.....	13
Fig. 2.7	Static DCVS basic circuit.....	14
Fig. 2.8	Static DCVS two-input NAND gate.....	15
Fig. 2.9	Static DCVS two-input XOR gate.....	16
Fig. 2.10	Block diagram of clocked DCVS circuit.....	18
Fig. 2.11	Clocked DCVS two-input XOR gate.....	19
Fig. 2.12	Block diagram of DSL circuit.....	20
Fig. 2.13	DSL basic circuit.....	22
Fig. 2.14	Reconfiguration of DSL basic circuit.....	23
Fig. 3.1	Conventional Static CMOS circuit for $f = x + \bar{y}z$	26
Fig. 3.2	Conventional static CMOS two-input NOR gate.....	26
Fig. 3.3	Conventional static CMOS two-input NAND gate.....	28
Fig. 3.4	Conventional static CMOS AND-OR-INVERT gate.....	28
Fig. 3.5	Conventional static CMOS circuits for the carry-function.....	30
Fig. 3.6	Conventional static CMOS inverter with capacitive load, C_L	32
Fig. 3.7	Input and output voltage waveforms of the CMOS inverter.....	32
Fig. 3.8	A 3-input conventional static CMOS NAND gate.....	32
Fig. 3.9	Encirclement of the K-map for the carry-out function of a full adder.....	38
Fig. 3.10	DCVS/DSL implementation of the carry-out function of a full adder.....	38
Fig. 3.11	The K-map of Fig. 3.9, but with different encirclements.....	40
Fig. 3.12	The DCVS/DSL logic trees resulting from 10-loops of Fig. 3.11.....	40
Fig. 3.13	The complete DCVS/DSL logic trees resulting from Fig. 3.11.....	40
Fig. 3.14	The K-map for the function $Q = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1(x_2 + x_3 + x_4)$ showing the 10 and 01 encirclements.....	42
Fig. 3.15	DCVS/DSL logic tree corresponding to the 01-loops of K-map shown in Fig. 3.14.....	42
Fig. 3.16	The complete DCVS/DSL logic trees for the function $Q = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1(x_2 + x_3 + x_4)$	42
Fig. 3.17	An alternative encirclement of the K-map of Fig. 3.14.....	43
Fig. 3.18	The DCVS/DSL trees resulting from Fig. 3.17.....	43

Fig. 4.1	Cascaded chain of ten 2-input NAND gates.....	46
Fig. 4.2	2-input static CMOS NAND gate.....	48
Fig. 4.3	2-input static DCVS NAND gate in the cascaded chain of Fig. 4.1.....	51
Fig. 4.4	2-input DSL NAND gate.....	53
Fig. 4.5	Interface between conventional static CMOS and DSL logic.....	54
Fig. 4.6	Static power dissipation as a function of reference voltage.....	56
Fig. 4.7	Logic low voltage level at the internal nodes as a function of reference voltage.....	58
Fig. 4.8	Logic low voltage level at the I/O nodes as a function of reference voltage.....	59
Fig. 4.9	Logic high voltage level at the I/O nodes as a function of reference voltage	60
Fig. 4.10	Variation of propagation delay with reference voltage.....	61

List of Tables

Table 4.1:	Simulated performance of conventional static CMOS circuit.....	49
Table 4.2:	Simulated performance of static DCVS circuit.....	50
Table 4.3:	Simulated performance of DSL circuit.....	55

Chapter 1



Introduction

1.1 Aims

CMOS technology finds ubiquitous use in the majority of leading-edge commercial applications for its very low static power dissipation [1]. But CMOS technology demonstrates lower speed than other technologies such as silicon bipolar technology, Gallium Arsenide (GaAs) technology and Josephson junction technology [1], [2]. So, instead of conventional CMOS technique, other CMOS circuit techniques, for example, Differential Cascode Voltage Switch (DCVS) and Differential Split-Level (DSL) CMOS techniques have been proposed, but not yet implemented in VLSI chips. The objective of this research is to analyze and compare the performance of the differential CMOS circuits to that of conventional CMOS circuits with a view to determine the suitability of these circuits for VLSI design. The results of this work are also expected to optimize the design of VLSI circuits using these techniques in terms of speed, propagation delay etc.

1.2 Literature Review

Differential Cascode Voltage Switch (DCVS) logic was introduced with a view to improve the speed of CMOS circuits [3]. The speed advantage is mainly due to the reduction of input gate capacitance loading, typically by a factor of 2 to 3, compared to conventional static CMOS. Unlike conventional static CMOS, DCVS logic requires complementary inputs and generates complementary outputs. Since, the complementary p-transistor network (load network) of a CMOS gate is replaced by a second n-transistor logic tree in the corresponding DCVS gate, the later is more compact than the former [3], [4]. Moreover, since DCVS gates generate complementary signals, the necessity of signal inverters is eliminated. This helps in achieving compact and regular structure which is very suitable for VLSI implementation. Despite the advantages mentioned above, static DCVS circuits suffer from the disadvantage of skew between the complementary outputs and long output settling time [3], [5]. These problems can be eliminated by using clocked DCVS logic [3]-[5].

To increase the speed in CMOS circuits even further, DSL CMOS technique was introduced [6]. In fact, it is a modification of the static DCVS logic. Both DSL and DCVS logic use the same complementary logic tree structures. However, the cross-coupled pMOS loads of static DCVS logic are replaced by cross-coupled nMOS-pMOS loads in DSL logic. This results in maximum logic swing of only $V_{DD}/2$ at the I/O nodes of DSL gates compared to full V_{DD} swing in standard CMOS and DCVS logic gates. The speed advantage of DSL was claimed to be as high as 10 times compared to standard CMOS provided short-channel logic n-transistor can be used [6]. However, DSL circuits have high static power dissipation and low noise immunity compared to conventional static CMOS. A method of reducing the quiescent power dissipation was proposed in [7]. If this technique is employed, however, the channel lengths of the logic n-transistors can not be

reduced below the process minimum . owing to larger than half V_{DD} swing at the I/O nodes. Moreover, it was shown in [5] that the speed advantage of DSL circuits over standard CMOS is not as high as claimed in the original paper [6].

Neither DCVS nor DSL logic has been used so far to design VLSI chips. Some of the problems associated with both these logic families as mentioned above have deterred VLSI designers from using them in chip design. This thesis undertakes to carry out detailed investigations into the performance of the various CMOS logic families, i.e., conventional static CMOS, DCVS and DSL logic. The goal is to determine their comparative suitability for VLSI implementation.

1.3 Organization of the Thesis

The principles and operation of various CMOS circuits are presented in Chapter 2. This includes conventional static CMOS and two different types of differential CMOS, i.e., DCVS logic and DSL logic. Chapter 3 presents the design procedures of the various CMOS logic circuits. Both the designs of logic networks as well as device sizing are considered. Chapter 4 presents the results obtained throughout the course of this work. It also contains analysis of the results. Detailed simulation results on the performance of various designs of logic gates using different CMOS logic families are described. Chapter 5 concludes the work done with some recommendations for future work.

Chapter 2

CMOS Circuit Techniques

2.1 Introduction

In this chapter, various CMOS circuit techniques are presented. The principles of operation of conventional static CMOS and various differential CMOS circuits are described in detail. Two different types of differential CMOS circuits: DCVS logic and DSL logic are discussed.

2.2 Conventional Static CMOS Logic

The structure of a conventional static CMOS logic gate consists of a driver network comprising only nMOS transistors and a load network comprising only pMOS transistors, where all MOSFETs are of enhancement mode [1] as shown in Fig. 2.1. The load network is connected between the power supply voltage V_{DD} and the output node V_{out} . The driver network is connected between the V_{out} and GND. To explain the operation of conventional static CMOS circuits, the load and driver networks are replaced by two digital switches as shown in Fig. 2.2. The load and driver switches are never closed simultaneously. The load and driver switches operate in antiphase, i.e., when one is open, then the other one is close. A high input causes the driver switch to close and load switch to remain open, so V_{out} is connected to ground. Thus, the output is logic LOW. A low input closes the load switch and opens the driver switch, so V_{out} is connected to

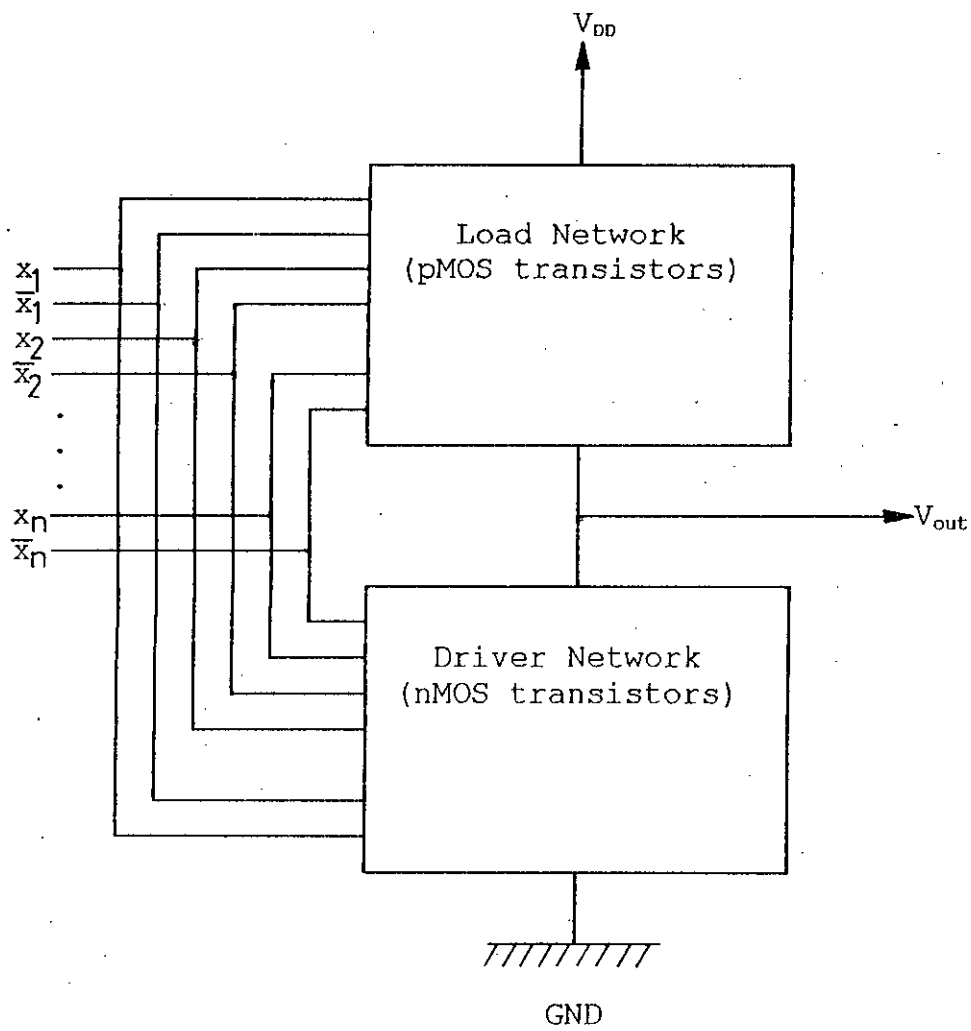


Fig. 2.1 Structure of conventional static CMOS logic gates.

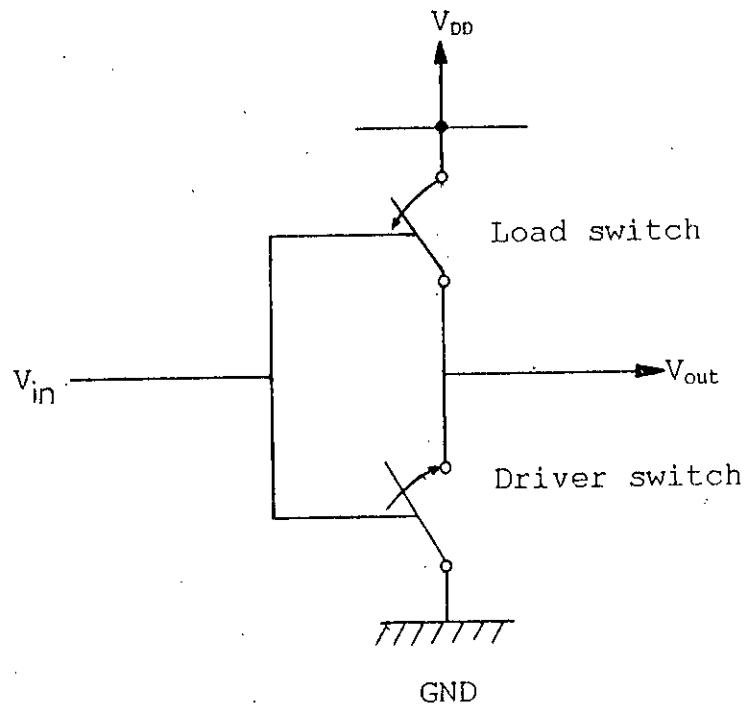


Fig. 2.2 Conventional static CMOS circuit with load and driver networks replaced by digital switches.

the supply voltage V_{DD} . Thus, the output is logic HIGH. The output logic levels are independent of the sizes of load and driver networks. This is why CMOS circuits are referred to as "ratioless" circuits [8]. When no input changes, then ideally there should be no current through the load and driver network.

The circuit diagram of a conventional static CMOS inverter is shown in Fig. 2.3. It consists of an enhancement nMOS transistor $n1$ as the driver and an enhancement pMOS transistor $p1$ as the load. The threshold voltage of $n1$, V_{thn} is the minimum gate-to-source voltage at which $n1$ conducts, while $p1$'s threshold voltage, V_{thp} is the minimum gate-to-source voltage at which it turns on. Obviously, the threshold voltages of nMOS and pMOS enhancement devices are positive and negative respectively. A low input voltage, i.e., $V_{in} = 0$ volt causes $n1$ to be OFF and $p1$ to be ON; the output equals V_{DD} . For a high input voltage of V_{DD} , $n1$ is ON and $p1$ is OFF. The output is then grounded via $n1$.

When no input changes, there is no conducting path from V_{DD} to GND via the load and driver network. Therefore, the static power dissipation (P_{SS}) in conventional CMOS circuits is very low. However, a small "leakage current" flows through the diodes formed between the source/drain of $n1/p1$ and substrate or well [1]. To reduce the amount of leakage current, these diodes are reverse biased by connecting the substrate/well of the pMOS and nMOS devices to the most positive and most negative voltage in the circuit respectively [1]. Thus, the static power dissipation in static CMOS circuits is mainly due to flow of reverse saturation current of the above mentioned diodes which is very small.

Whenever the inputs of conventional static CMOS circuits change in such a manner that the load network is conductive, then the load capacitance C_L at the output terminal (including parasitic capacitances at the inputs of the succeeding CMOS gates) charges up to the full power supply voltage V_{DD} .

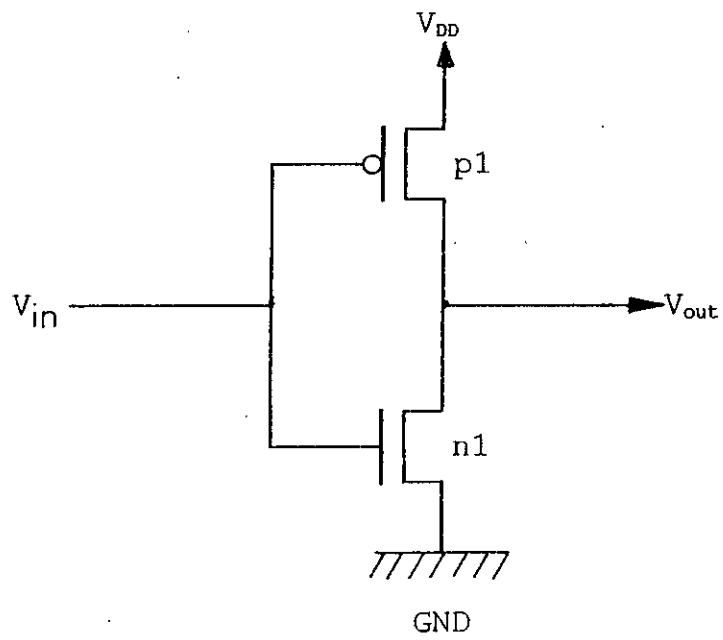


Fig. 2.3 Conventional static CMOS inverter.

Then if the inputs change so that the driver network is conductive, the charge stored in the load capacitance discharges. Therefore, current flows as long as the output discharges. This duration is dependent upon the input signal. The resulting power dissipation is called ac or dynamic power dissipation and is given by [9]:

$$P_{\text{dynamic}} = f C_L V_{DD}^2$$

where,

f = switching frequency of the input

C_L = parasitic capacitance

V_{DD} = supply voltage

2.3 Differential CMOS Logic

Differential CMOS logic was introduced for speed improvement in CMOS logic circuits [3], [4]. The basic differential CMOS circuit comprises two parts: complementary binary decision trees and a load as shown in Fig. 2.4. It requires complementary inputs and produces complementary outputs Q and \bar{Q} . For easier understanding, the binary decision trees of Fig. 2.4 are replaced by two anti-phase switches in Fig. 2.5. The tree is specified such that

- 1) when the input vector $x = (x_1, x_2, \dots, x_n)$ is the true vector of the switching function $Q(x)$, then the output Q is disconnected from GND and the node \bar{Q} is connected to GND; and
- 2) when $x = (x_1, x_2, \dots, x_n)$ is the false vector of $Q(x)$, then the reverse holds.

The tree network is constructed with nMOS transistors only. The differential CMOS logic family can also be divided in more classes depending on the variations of the load circuit.

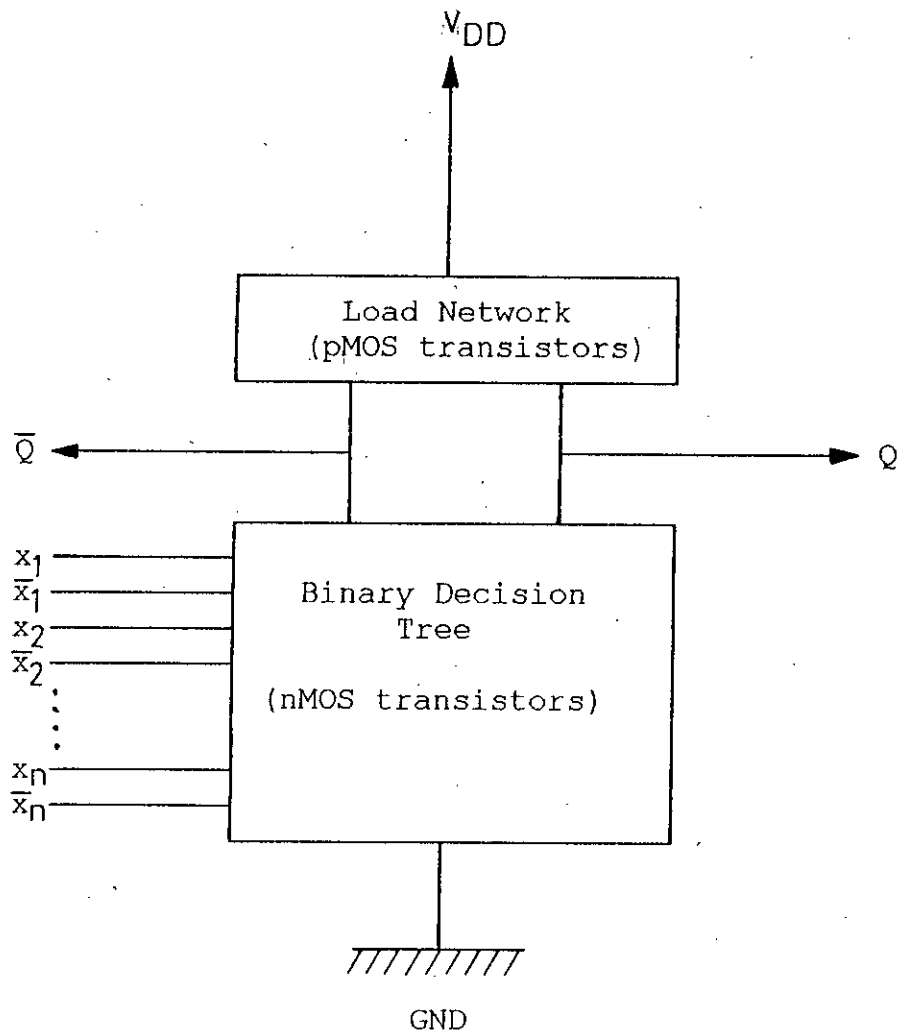


Fig. 2.4 Block diagram of differential CMOS circuits.

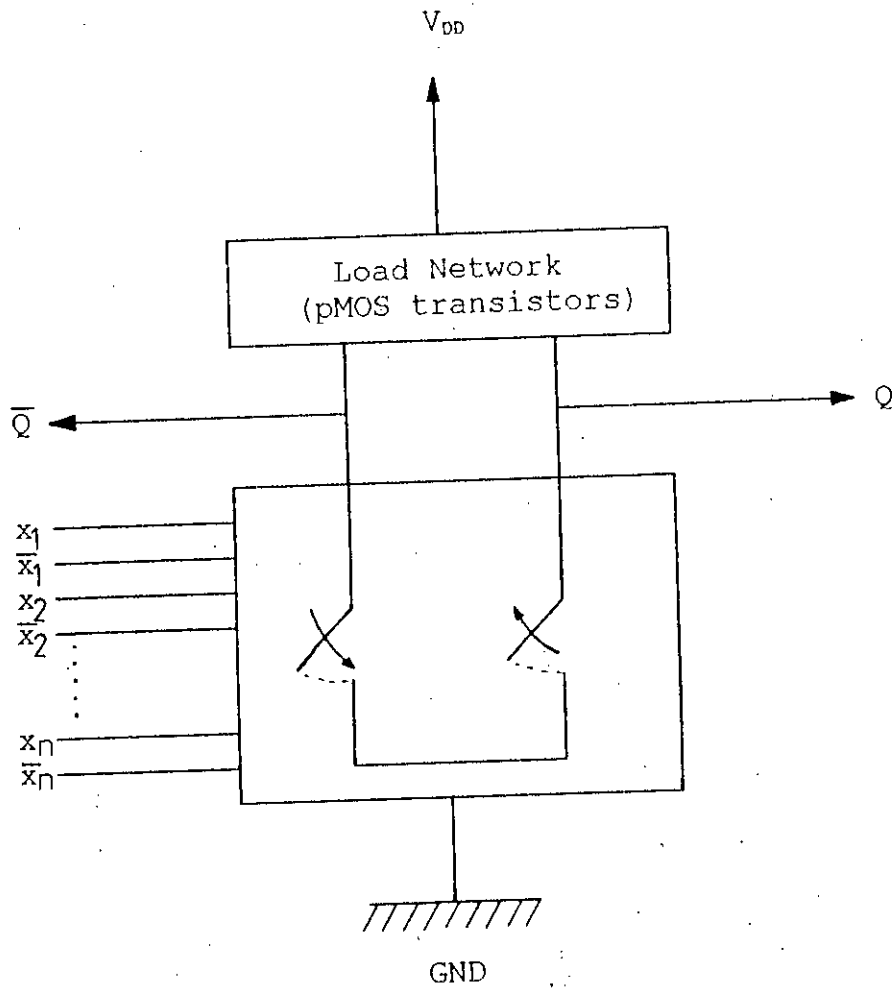


Fig. 2.5 Differential CMOS circuit with decision trees replaced by digital switches.

2.3.1 Static Differential Cascode Voltage Switch (DCVS) Logic

The basic form of this style of differential CMOS logic is depicted in Fig. 2.6. The load for this circuit is a simple latch made up of a pair of cross-coupled p-type pull-up transistors. To explain the switching behavior of the static DCVS circuit technique, the differential nMOS trees are replaced by two nMOS transistors as shown in Fig. 2.7.

Now let input D be switched from a low to a high level, starting with input D low and input DN high. The node \bar{Q} is at a high level of V_{DD} and node Q at a low level of zero volt, so pMOS p1 is ON and p2 is OFF. If we now switch the inputs D and DN then nMOS n1 turns ON and n2 turns OFF. This is ratioed logic because transistor n1 has to discharge node \bar{Q} , while p1 is still ON. Transistor p1 switches OFF, after p2 has switched ON and node Q has reached a high level. So during switching both n1 and p1 (or n2 and p2 depending on the input transition) conduct, causing relatively large current spikes and additional delay. However, the logic trees do not pass any direct current after the latch sets. Since the inputs drive only the nMOS tree devices, input gate capacitance loading is typically a factor of three times smaller than CMOS circuits that require complementary n-channel and p-channel devices to be driven. An obvious disadvantage is the need for two inputs (true and complement) for each variable. Figures 2.8 and 2.9 show implementations of NAND gate and XOR gate respectively in static DCVS logic

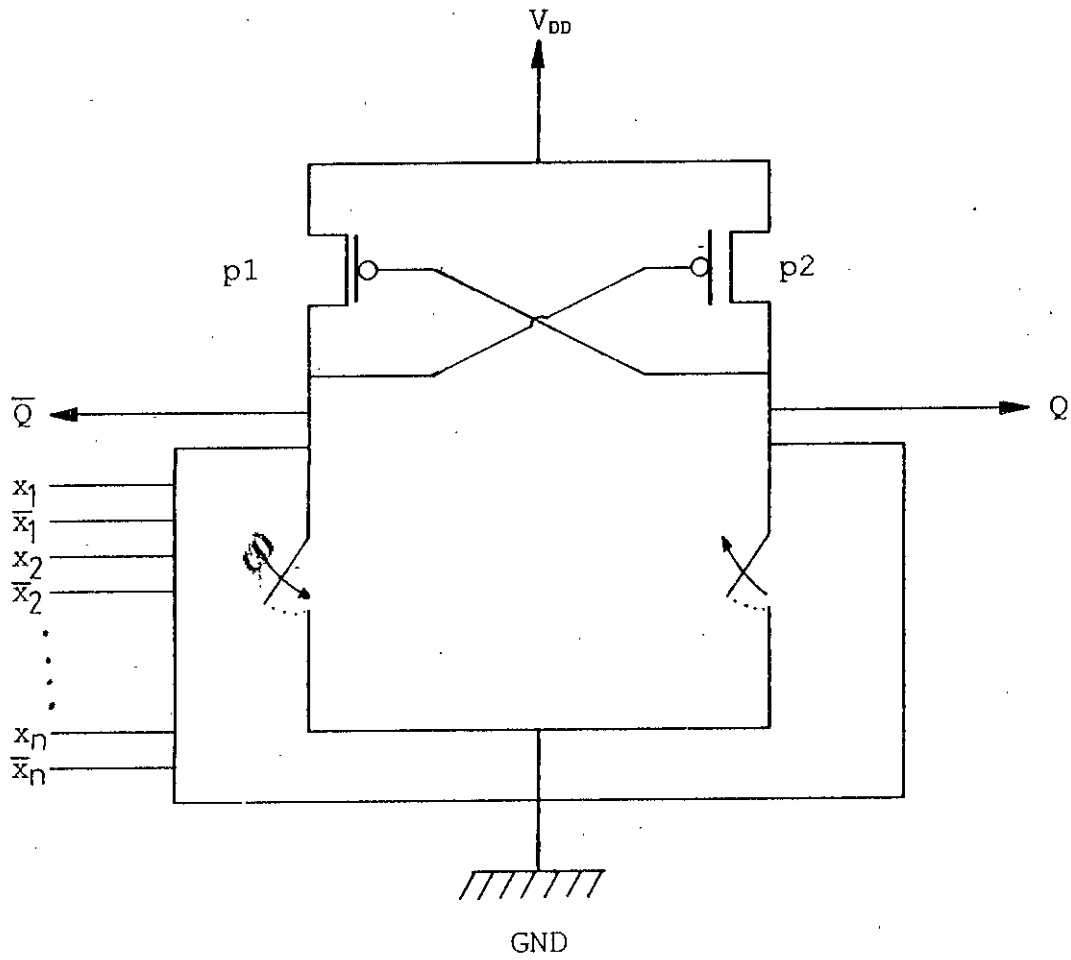


Fig. 2.6 Block diagram of static DCVS circuit.

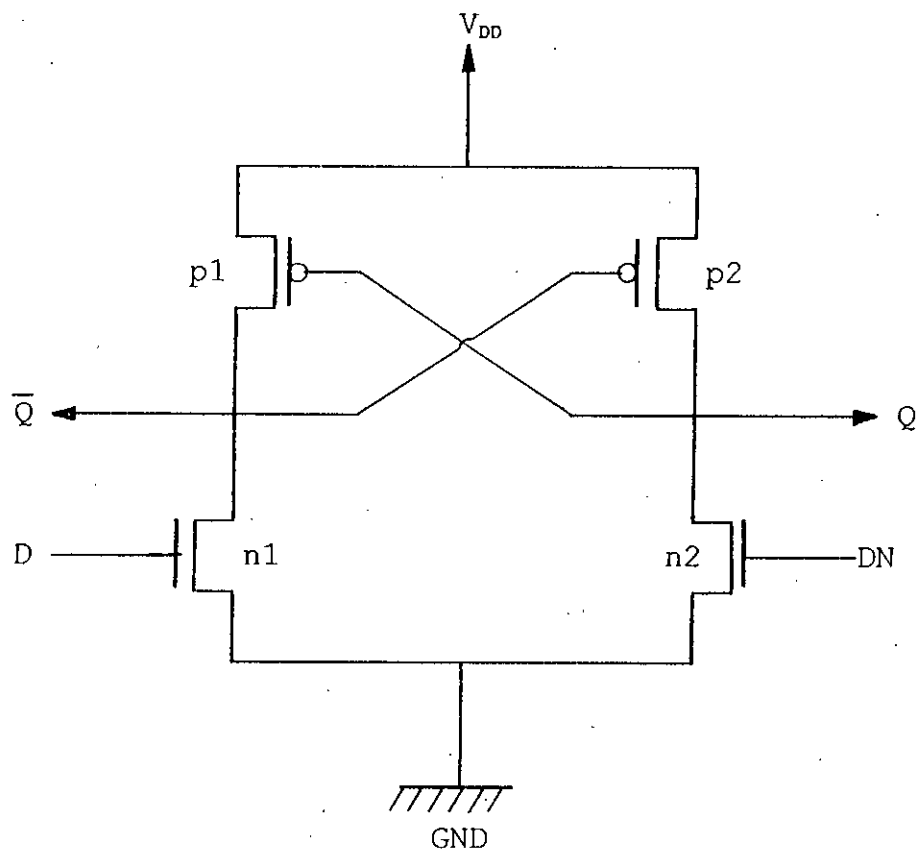


Fig. 2.7 Static DCVS basic circuit.

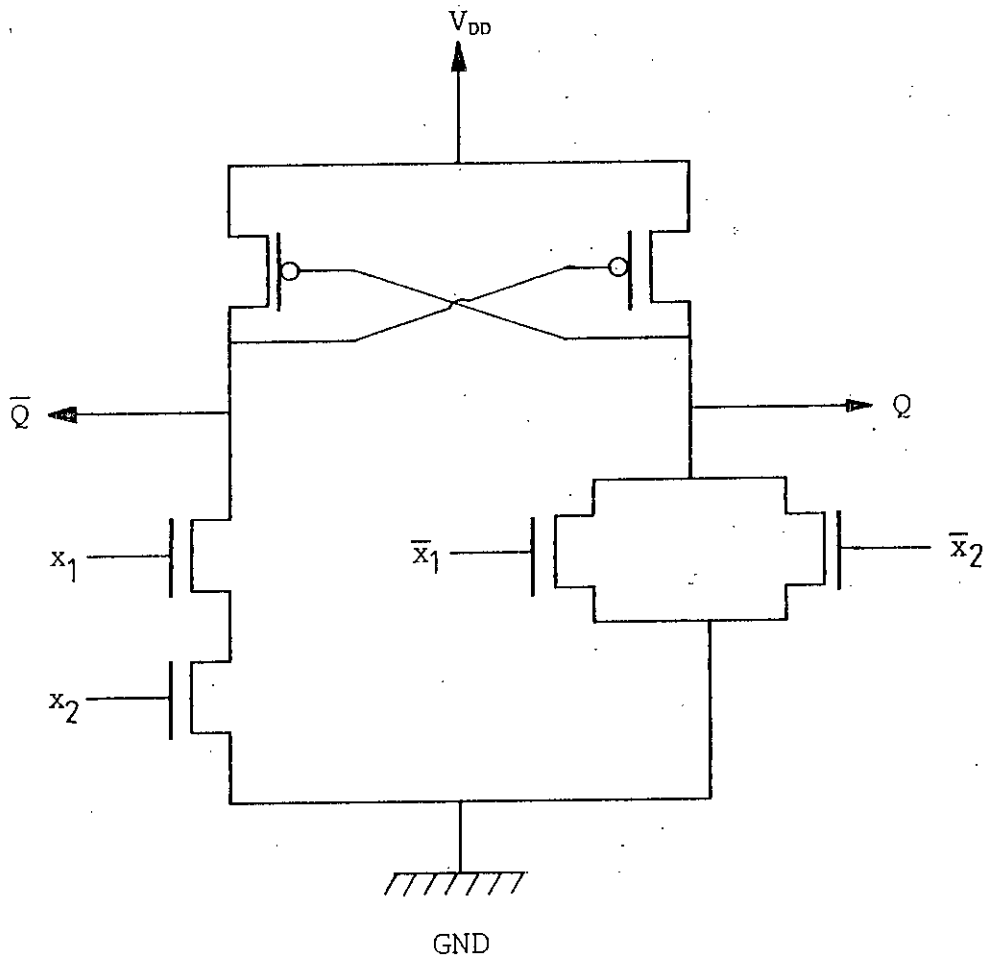


Fig. 2.8 Static CMOS two-input NAND gate.

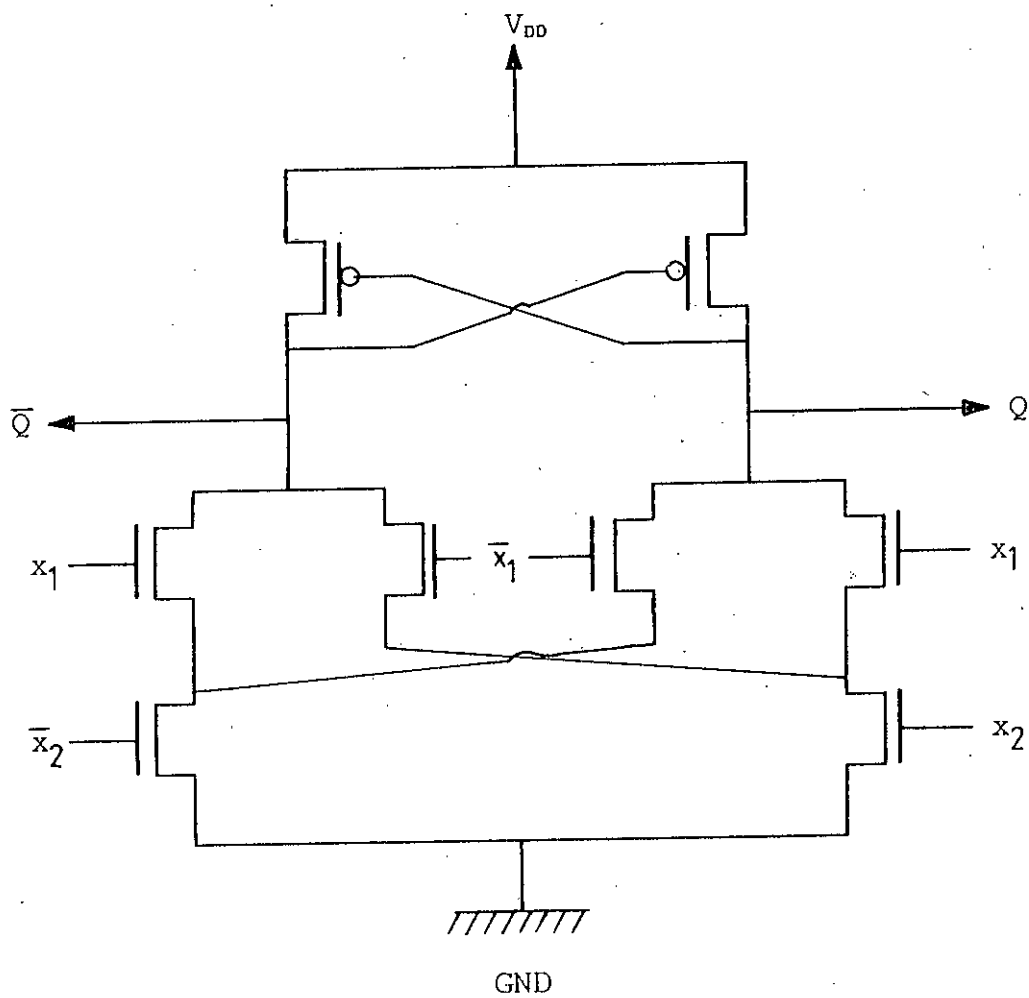


Fig. 2.9 Static DCVS two-input XOR gate.

2.3.2 Clocked Differential Cascode Voltage Switch (DCVS) Logic

Clocked DCVS logic overcomes the problems of long output settling time of pMOS latch of static DCVS logic. The load and tree arrangement of a clocked DCVS circuit are shown in Fig. 2.10. The internal nodes F and \bar{F} are precharged high during the precharge phase when PC is low. When $PC = 0$, output nodes Q and \bar{Q} are logic low. In this phase n-device $n1$ is OFF so the transistors inside the tree network are insensitive to differential inputs. At the completion of precharge, PC is made high; so, the path to V_{DD} is turned off and the path to ground is turned on. Then depending on the state of the differential inputs either node Q or \bar{Q} will float at high level or will be pull down to ground level. Feedback devices $T1$ and $T2$ hold the internal nodes F and \bar{F} statically high prior to switching within the logic tree. The feedback devices reduce charge sharing noise within the tree and improve the noise margin, with only a small sacrifice in performance [3]. The logic invert function is implicit in this closed DCVS, a clear advantage over other incomplete domino type logic families. All logic functions can be implemented using clocked DCVS logic. As an example, a two-input XOR gate in clocked DCVS is shown in Fig. 2.11.

2.3.3 Differential Split-Level (DSL) CMOS Logic

The load and tree arrangement of DSL CMOS logic circuit is shown in Fig. 2.12. Here the load circuit consists of cross-coupled current-controlled cascoded n-and p-transistors. This load is similar to that of static DCVS circuit, shown in Fig. 2.6, except two extra nMOS transistors $n10$ and $n20$ are placed between the pMOS transistor part and the logic tree of nMOS transistors. The gates of transistors $n10$ and $n20$ are connected to a common reference voltage V_{ref} . The gates of the pMOS transistors $p1$ and $p2$ are connected to the drains of $n1$ and $n2$ in cross-coupled manner.

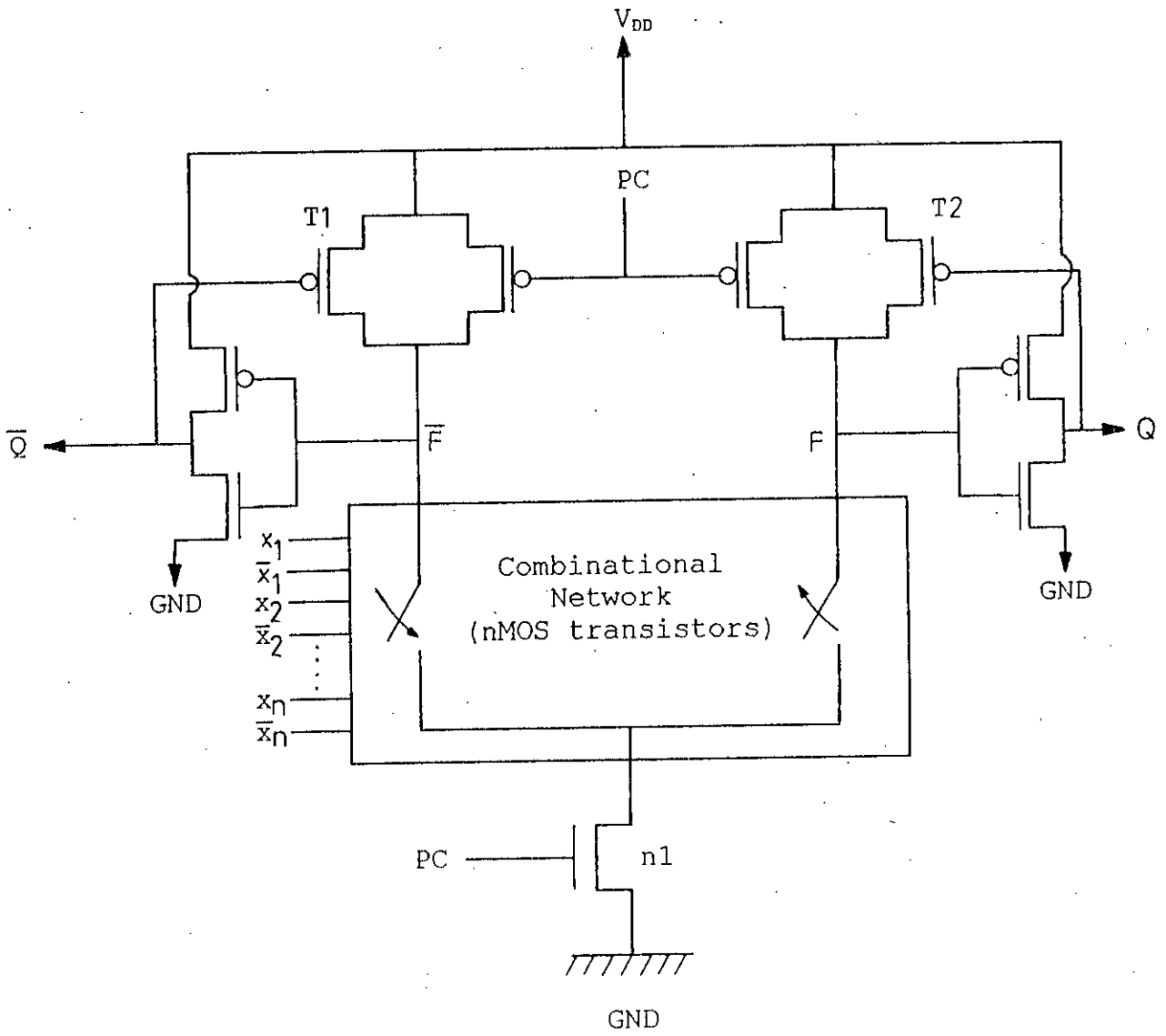


Fig. 2.10 Block diagram of clocked DCVS circuit.

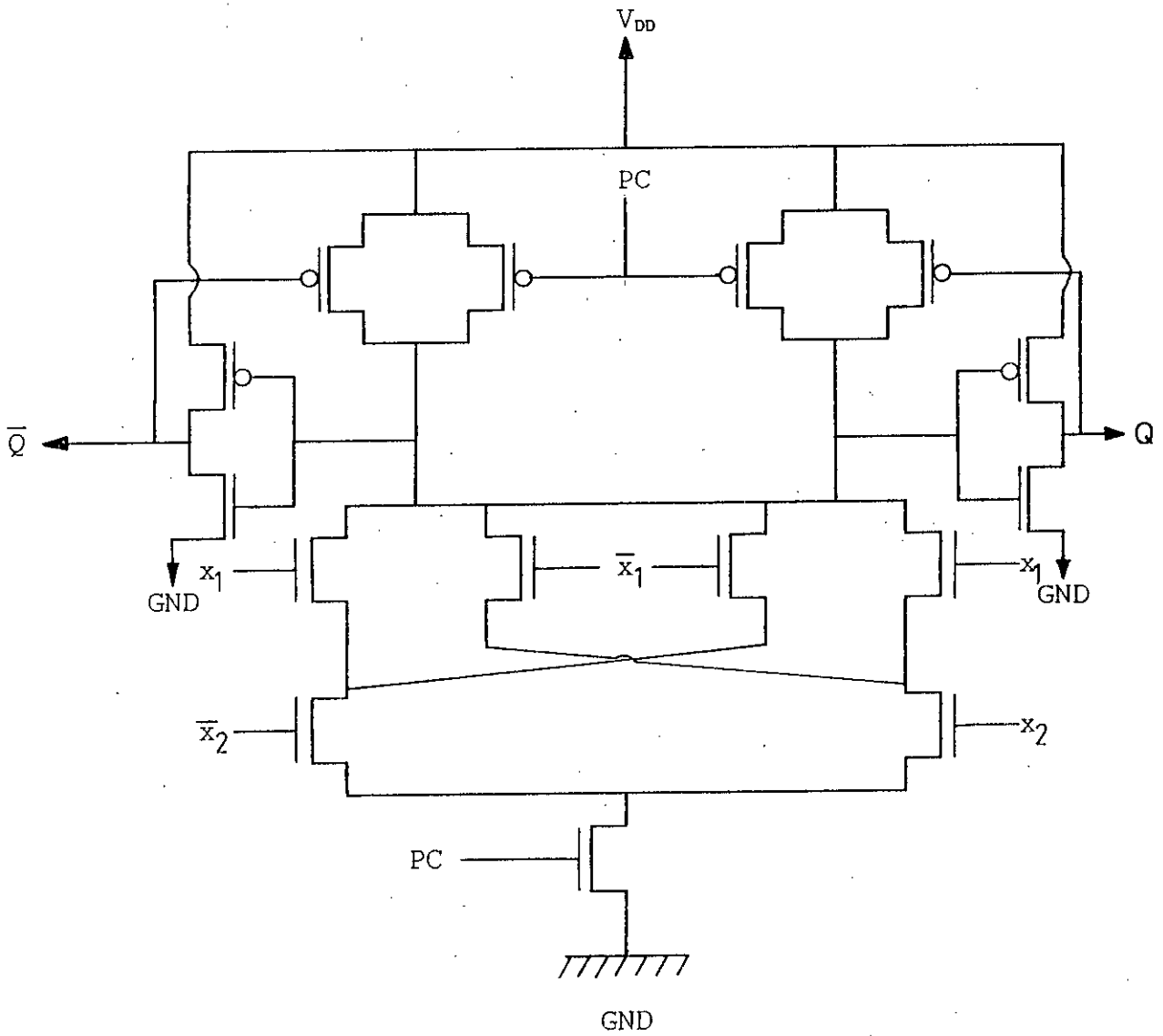


Fig.2.11 Clocked DCVS two-input XOR gate

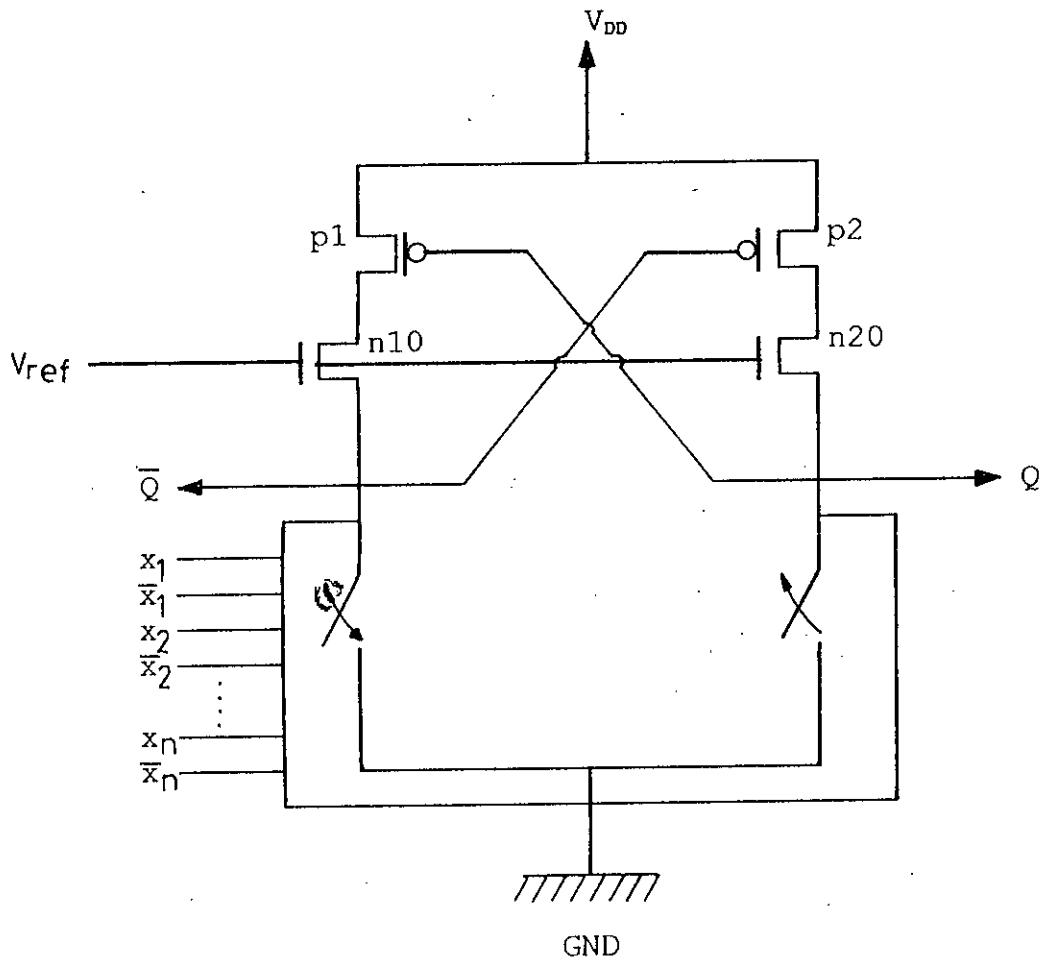


Fig. 2.12 Block diagram of DSL circuit.

To explain the switching behavior of the DSL circuit technique the differential nMOS logic trees are replaced by two nMOS transistors as shown in Fig. 2.13. Let input D be switched from a low to a high-level starting with input D low and DN high. Then nMOS n2 is ON and so, pMOS p1 is ON; then node \bar{f} has a high level of V_{DD} . In contrast, node f has a low level. The reference voltage determines the high-level at node \bar{Q} to V_{ref} minus the threshold voltage of the nMOS transistors. Node Q has a low level which is not exactly zero volt but varies from tens to hundreds of millivolts, because pMOS p2 is weakly ON. This causes static power dissipation. The nMOS transistor n10 is cut off, which causes a high impedance to V_{DD} for node \bar{Q} . If now the inputs D and DN are switched to high and low respectively then nMOS n1 turns ON and n2 turns OFF. The voltage level at \bar{Q} which is at V_{ref} minus the threshold voltage of the nMOS transistors will immediately be discharged and pMOS p2 turns more ON from its weakly ON state to its high drive state. At the same time nodes f and Q start rising because pMOS p2 was already partly ON, causing pMOS p1 to switch faster to its low drive state.

The switching speed of this circuit technique can be further improved by reconfiguring the basic circuit of Fig. 2.13. The maximum drain-source voltage of only V_{ref} minus the threshold voltage of nMOS transistors on nodes Q and \bar{Q} allows the channel length of nMOS transistors on tree network to be reduced. Because of the low voltage swing on nodes Q and \bar{Q} , it is preferable to use these nodes as outputs and inputs of the gates, thereby reducing the delay due to the wiring capacitances. Fig 2.14 shows the reconfiguration of the DSL basic circuit. At the inputs D and DN, we now have current controlled cascoded cross-coupled nMOS-pMOS loads and the outputs Q and \bar{Q} are the open drains of the logic nMOS transistors. The internal gate signals at g and \bar{g} of this figure correspond with the signals at f and \bar{f} of Fig. 2.13.

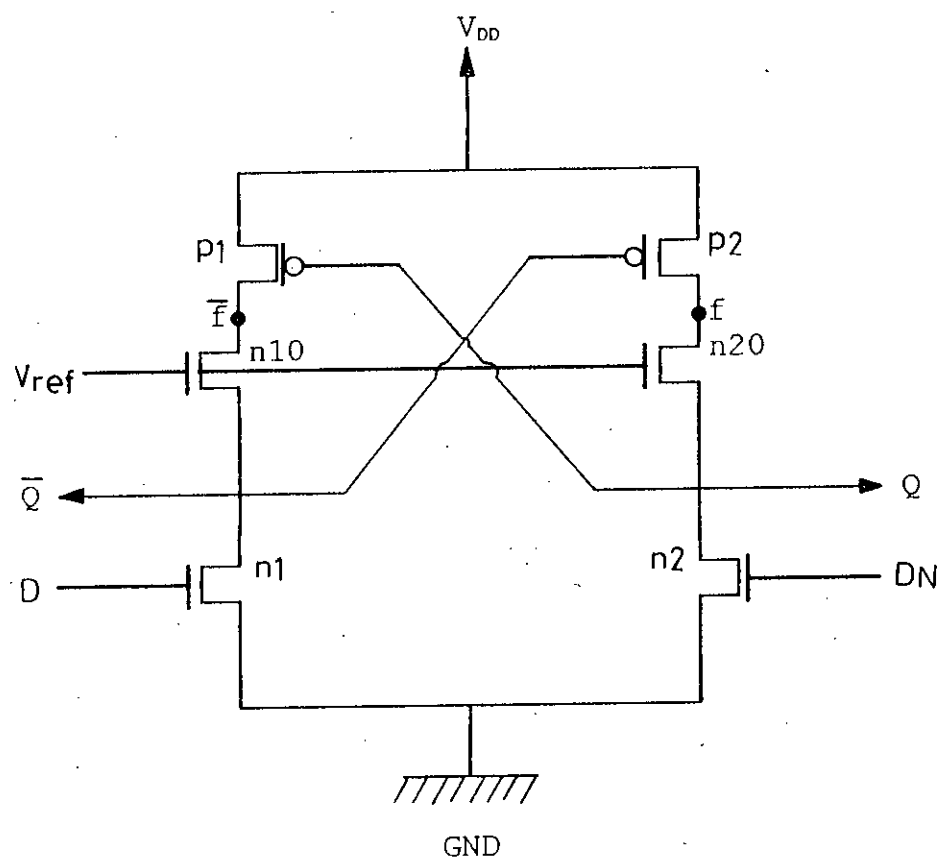


Fig. 2.13 DSL basic circuit.

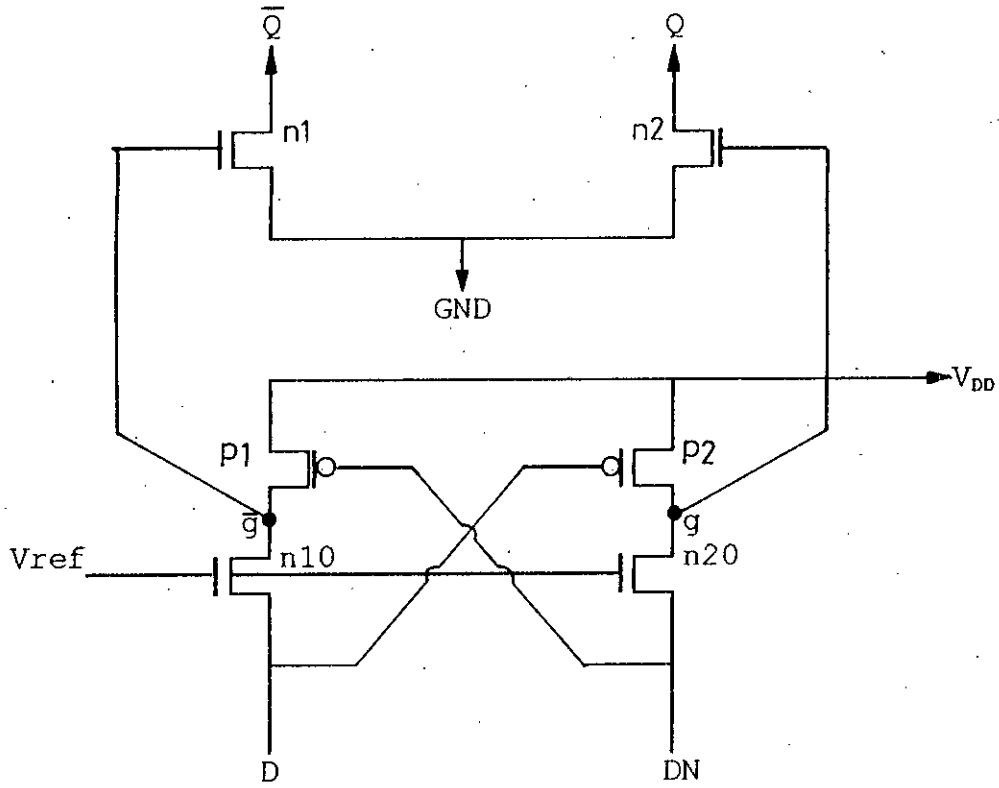


Fig. 2.14 Reconfiguration of DSL basic circuit.

Chapter 3

Design of CMOS Circuits

3.1 Introduction

This chapter presents the various aspects of designing CMOS VLSI circuits. Circuit design is the realization of the required logic for a system in terms of transistor circuits. The design objective is to produce a circuit which optimize the often conflicting requirements of minimum silicon area, minimum power consumption and maximum circuit speed. This chapter addresses these issues for designing VLSI circuits using conventional static CMOS and differential CMOS logic.

3.2 Design of Conventional Static CMOS Circuits

There are many ways [10] in which the load and driver networks of conventional static CMOS logic circuits can be derived. The objective of such design procedures is either to minimize the total number of transistors used resulting in minimum silicon area, or to keep the number of series transistors to below a certain limit. Note that in conventional static CMOS circuits, the load network consisting of p-devices is configured in a way complementary to the driver network

consisting of n-devices. Once the driver and load networks are derived, the widths and lengths of the devices are determined on the basis of performance required, i.e., rise/fall time, etc. In this section, a method of obtaining optimum networks as well as the device sizing are discussed.

3.2.1 Conventional Static CMOS Network Derivation

The load and driver networks of conventional static CMOS circuit technique can be derived easily if the switching function is given in canonic forms [10]. A switching function can be expressed in one of the two canonic forms: the sum-of-products or the product-of-sums form. Let a function f be given in a sum-of-products form:

$$f = x + \bar{y}z$$

To derive the driver network consisting of nMOS transistors first complementary function \bar{f} is obtained

$$\begin{aligned}\bar{f} &= \overline{(x + \bar{y}z)} \\ &= \bar{x} \cdot (y + \bar{z})\end{aligned}$$

Thus, \bar{f} is in product-of-sums form. $y + \bar{z}$ part of \bar{f} is implemented by using two n-devices $n1$ and $n2$ in parallel with inputs y and \bar{z} as shown in Fig. 3.1. With this parallel combination of $n1$ and $n2$, another n-device $n3$ is connected in series with input \bar{x} to obtain the complete driver network.

The dual network of the driver network is used to obtain the load network with only p-devices whose gates have same inputs as those for corresponding dual elements. The dual of $n1$ and

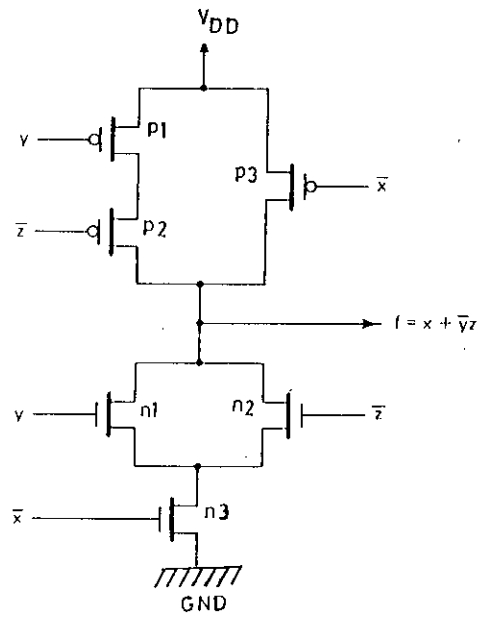


Fig. 3.1 Conventional Static CMOS circuit for $f = x + \bar{y}z$.

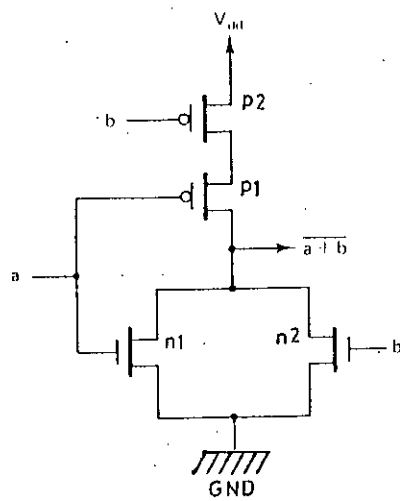


Fig. 3.2 Conventional static CMOS two-input NOR gate.

n2 conventional of driver network is obtained by using two p-devices p1 and p2 in series. Then another p-device p3 is used in parallel with this series combination.

Considering generalized function $f(x_1, x_2, \dots, x_n)$, the load network is constructed with p-devices such that there are conducting paths from V_{DD} to V_{out} for all input combinations (x_1, x_2, \dots, x_n) for which the desired function $f(x_1, x_2, \dots, x_n) = 1$; the driver network is constructed with n-devices such that there are conducting paths from V_{out} to GND for all input combinations for which $f(x_1, x_2, \dots, x_n) = 0$.

The load and drive networks bear a dual relationship by DeMorgan's theorem, stated in its most general form as

$$f(x_1, x_2, \dots, x_n) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$$

which says that the complement (\bar{f}) of any function (f) can be obtained by replacing each variable by its complement and by interchanging the OR (parallel connection) operation with the AND (series connection) operation at each level of expression for f .

Using the foregoing principle, a conventional static CMOS network for arbitrary combinational function can be derived. The commonly used two-input NOR and NAND gates are shown in Fig. 3.2 and Fig. 3.3 respectively.

The AND-OR-INVERT gate, expressed as $f = \overline{(ab+cd)}$, is shown in Fig. 3.4. It consists of a parallel connection of series n-transistors for driver network and a series connection of parallel p-transistors for load network. Another interesting complex gate is the 'carry output function' from a full adder stage,

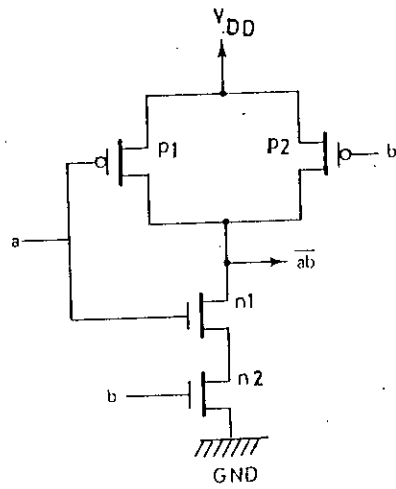


Fig. 3.3 Conventional static CMOS two-input NAND gate.

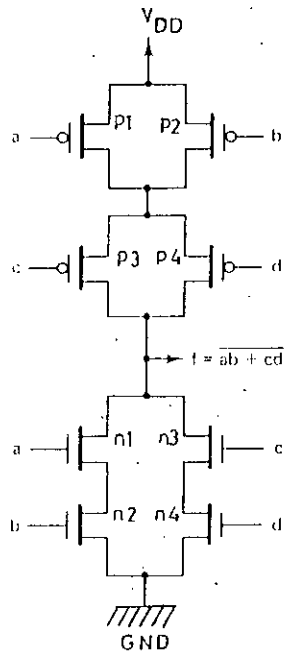


Fig. 3.4 Conventional static CMOS AND-OR-INVERT gate.

$$M(a,b,c) = ab+bc+ca \quad 3.2(i)$$

$$= ab+c(a+b) \quad 3.2(ii)$$

where M represents carry output bit if c is carry-in, and a and b represent the input bits to the stage.

Two conventional static CMOS circuits for \overline{M} based on the above two expressions [3.1(i) and 3.1(ii)] are shown in Fig. 3.5 (a) and (b).

Summarizing, the general procedure to design a conventional static CMOS network for an arbitrary combinational function f is as follows: starting with an expression for the complementary function \overline{f} , a series-parallel combination of n -devices is obtained for driver network and then the load network structure is obtained from the dual of the driver network using p -devices; the gates of the load network have the same inputs as those for the corresponding dual elements.

3.2.2 Device Sizing for Conventional Static CMOS Circuits

The selection of device sizes, i.e., channel lengths and widths in a CMOS design depends upon the performance expected from the circuit. Since conventional static CMOS circuits are "ratioless", i.e., the output logic levels are independent of the device sizes, many designers prefer using minimum geometry devices for CMOS designs. However, in some applications, devices of suitable sizes have to be used in order to obtain, for example, equal rise and fall times, or high device capability.

Fig. 3.6 shows the familiar CMOS inverter with a capacitive load, C_L . The switching speed of this CMOS gate is limited by the time taken to charge and discharge C_L . An input transition

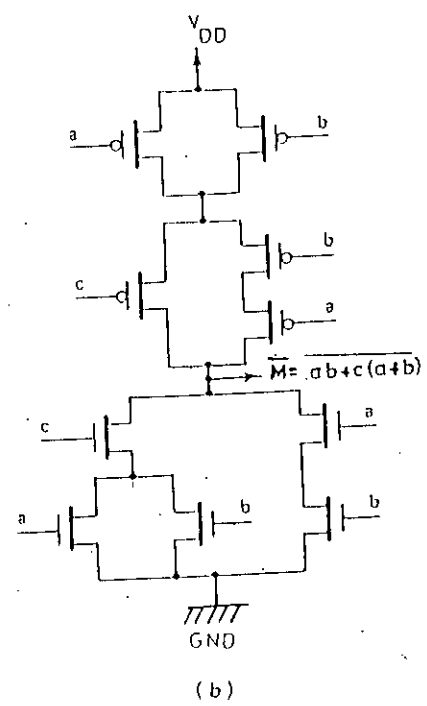
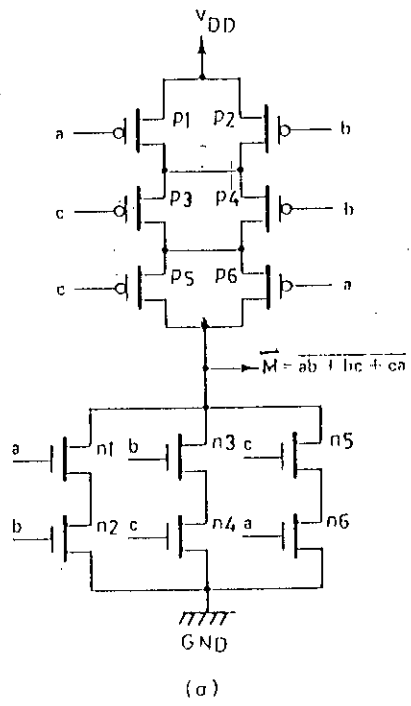


Fig. 3.5 Conventional static CMOS circuits for the carry-function.

results in an output transition that either charges C_L toward V_{DD} or discharges C_L toward GND. When the input is driven by a step waveform V_{in} , the output V_{out} is as shown in the Fig. 3.7. The fall time t_f (in sec), which is the time taken for V_{out} to fall from 90% to 10% of its steady-state value, is approximately given by [1],

$$t_f = (KC_L)/(\beta_n V_{DD}) \quad (3.1)$$

where,

$$\beta_n = (\mu_n \epsilon / t_{ox})(W_n / L_n), \text{ Farad/V-sec}$$

$$\mu_n = \text{mobility of electrons, cm}^2/\text{V-sec}$$

$$\epsilon = \text{permittivity of the gate insulator, Farad/cm}$$

$$t_{ox} = \text{thickness of the gate insulator, cm}$$

$$W_n = \text{channel width of n-device, cm}$$

$$L_n = \text{channel length of n-device, cm}$$

$$K = 3 \text{ to } 4, \text{ for } V_{DD} = 3 \text{ to } 5 \text{ volts, and } V_{thn} = 0.5 \text{ to } 1 \text{ volts.}$$

From the above expression, it can be written as

$$t_f \propto (L_n / W_n)$$

i.e. as the width of transistor is increased or the length is decreased, the fall time (t_f) decreases.

The rise time t_r (in sec) for the CMOS inverter of Fig. 3.6 which is the time taken for V_{out} to rise from 10% to 90% of its steady-state value, can be similarly approximated as [1],

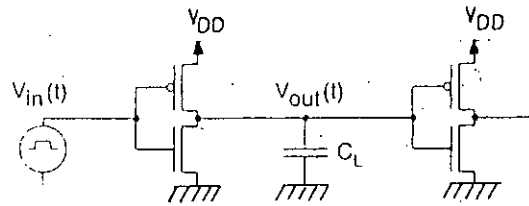


Fig. 3.6 Conventional static CMOS inverter with capacitive load, C_L .

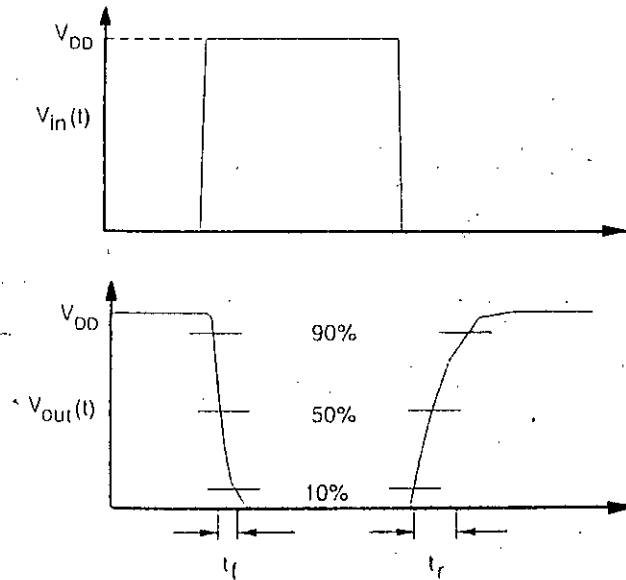


Fig. 3.7 Input and output voltage waveforms of the CMOS inverter.

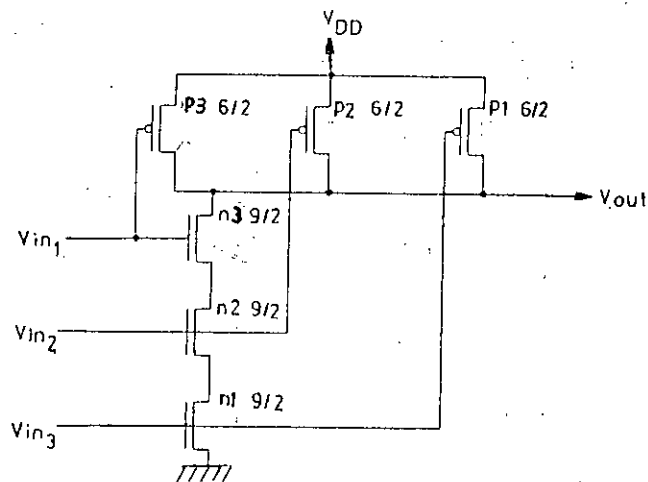


Fig. 3.8 A 3-input conventional static CMOS NAND gate.

$$t_f = (KC_L)/(\beta_p V_{DD}) \quad (3.2)$$

where,

$$\beta_p = (\mu_p \epsilon / t_{ox})(W_p / L_p), \text{ Farad/V-sec}$$

$$\mu_p = \text{mobility of holes, cm}^2/\text{V-sec}$$

$$\epsilon = \text{permittivity of the gate insulator, Farad/cm}$$

$$t_{ox} = \text{thickness of the gate insulator, cm}$$

$$W_p = \text{channel width of p-device, cm}$$

$$L_p = \text{channel length of p-device, cm}$$

$$K = 3 \text{ to } 4, \text{ for } V_{DD} = 3 \text{ to } 5 \text{ volts, and } V_{thp} = 0.5 \text{ to } 1 \text{ volts.}$$

For equally sized n- and p- transistors, where $\mu_n = 2\mu_p$, i.e., $\beta_n = 2\beta_p$, it can be seen from Eqn. (3.1) and (3.2) that

$$t_f = t_r/2$$

Thus, the fall time is faster than the rise time, primarily due to different carrier mobilities associated with the p- and n-devices. Therefore, for equal rise and fall time for an inverter

$$\beta_n = \beta_p$$

If $L_p = L_n$, then

$$(\mu_n / \mu_p) = (W_p / W_n)$$

As $\mu_n = 2 \rightarrow 3 \mu_p$ for most CMOS processes [1], the channel width of the p-device must be increased to approximately two to three times that of the n-device. So,

$$W_p = 2 \rightarrow 3 W_n$$

To accurately specify the width ratio required to achieve equal rise and fall times, an accurate ratio of β_p and β_n must be known. These, in turn, depends on the parameters of the process being used.

The delay through simple static CMOS gate may be approximated by constructing an "equivalent inverter". This is an inverter where the pull-down n-transistor and the pull-up p-transistor are of a size to reflect the effective strength of the real pull-down or pull-up path in the gate. For instance, in the 3-input NAND gate shown in Fig. 3.8, $L_p = L_n$ for all transistors. When the pull-down path is conducting, all of the n-transistors have be turned on.

The effective β of the n-transistors is given by

$$(1/\beta_{\text{neff}}) = (1/\beta_{n1}) + (1/\beta_{n2}) + (1/\beta_{n3})$$

For, $\beta_{n1} = \beta_{n2} = \beta_{n3} = \beta_n$

$$\beta_{\text{neff}} = \beta_n / 3$$

For the pull-up case, only one p-transistor has to turn on to raise the output.

Thus, $\beta_{\text{peff}} = \beta_p$

For approximately the same rise and fall time,

$$\beta_{\text{neff}} = \beta_{\text{peff}}$$

$$\text{or, } (\mu_n/\mu_p) = 3 (W_p/W_n)$$

The aspect ratio is usually indicated beside each transistor. Considering $\mu_n=2\mu_p$ the aspect ratios required for minimum width and length of $3 \mu\text{m}$ and $2 \mu\text{m}$ respectively are shown in Fig. 3.8.

3.3 Design of Differential CMOS Circuits

As discussed in Section 2.3, DCVS and DSL circuits differ only in the load network. While the DCVS circuits use only cross-coupled pMOS devices in the load network, DSL circuits use cross-coupled nMOS-pMOS load network. The logic trees of DCVS and DSL are same. Hence, as far as deriving the logic trees are concerned, the procedure is same for both DCVS and DSL circuits. Hence, the process of obtaining optimum logic trees using minimum number of n-transistors is same for both DCVS and DSL circuits. However, while DCVS circuits have logic low and high levels of zero volts and V_{DD} respectively, this is not the case for the DSL circuits. Since DSL circuits have high quiescent current at optimum V_{ref} , the sizing of the devices is a critical issue. This is to ensure that the logic low voltage levels in DSL circuits are below the acceptable level. The design procedure of both differential logic as well as the device sizing are presented next.

3.3.1 Differential Logic Tree Design

There are three design procedures for constructing differential logic trees. In one procedure, an algebraic technique based on the identification of sub-expressions common to two or more Boolean functions is used [11]. The decomposition and factorization techniques involved in this approach are quite mathematical. As such, this method does not provide the insight into circuit behavior which is often important for VLSI designers. This procedure will not be further explained in this thesis.

The remaining two procedures are much simpler and more practical for constructing DCVS/DSL logic trees. The first procedure uses the pictorial nature of Karnaugh map (K-map) [12]. This hand-processing method is shown to be an efficient approach to realizing low device-count circuits for functions upto five or six variables. However, the complexity of K-maps suddenly increases when more than five variables are considered. For higher number of variables a tabular method that is a modified form of the Quine-McCluskey approach [12], [13] can be used. The tabular method has a uniform procedural complexity for n variables.

Note that a unique one-to-one correspondence between a Boolean expression and a DCVS/DSL tree structure does not exist. So several tree structures may realize a particular logic operation.

K-map and tabular procedures can be used to implement Boolean function provided the appropriate truth tables are known. As in most of the normal cases less than six variables are involved, only the K-map approach is discussed here.

The K-map Design Procedure[12]

The input variable of a differential logic tree is represented by x_i , for $i=1, 2, \dots, n$. A literal is a variable x_i or its negation \bar{x}_i . A cube is a set P of literals such that $x_i \in P$ implies $\bar{x}_i \notin P$.

In a Karnaugh map of n variables, there are 2^n cells of which each represents a cube consisting of exactly n literals. Cells that contain ONE's are called 1-cells (similarly, 0-cells). An 1-loop that encircles two adjacent 1-cells expresses a cube with one less literal than each of the cubes representing the original 1-cell (similarly, 0-loop). Suppose that two rectangular 1-loops, each consisting of 2^i 1-cells, are adjacent on a K-map. If these 1-loops express cubes, say Cx_k and $C\bar{x}_k$, we get a new rectangular 1-loop consisting of 2^{i+1} 1-cells by combining the two 1-loops, and the new 1-loop expresses cube C (similarly for the 0-loops).

Before introducing the K-map algorithm, we give an example to demonstrate some of the ideas, i.e., given the Boolean function $Q = x_1 x_2 + x_2 x_3 + x_3 x_1$ (which is the form of the carry-out function of a full adder), construct the corresponding differential logic tree. The K-map is shown in Fig. 3.9. The 1- and 0-loops are encircled properly to form the minimal cover for the 1- and 0-cells, respectively.

Fig. 3.10 illustrates the resulting differential logic tree pair. The tree attached to node \bar{Q} is derived from the 1-cells and is called the 1-tree. Similarly, the 0-tree is derived from the 0-cells and is attached to node Q . Note that the 1- and 0-trees are disjoint because the 1-cells and 0-cells have been grouped separately. This DCVS/DSL circuit requires ten n -devices to realize the function Q .

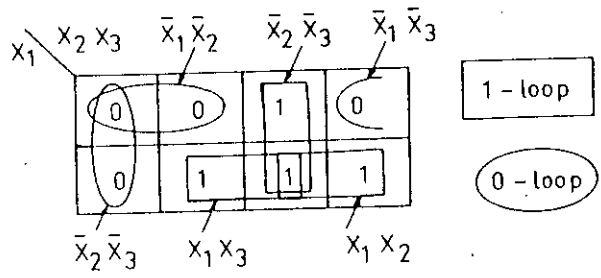


Fig. 3.9 Encirclement of the K-map for the carry-out function of a full adder.

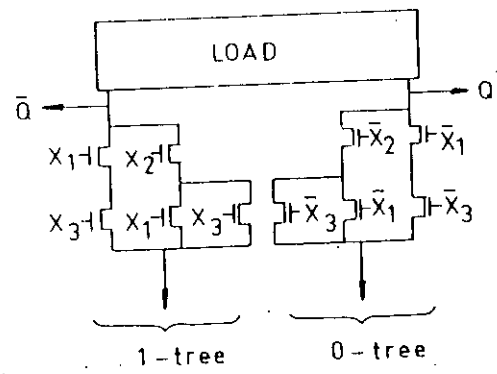


Fig. 3.10 DCVS/DSL implementation of the carry-out function of a full adder.

The K-map procedure does more than just construct the two disjointed 1- and 0-trees. It also allows the maximum commonality between these two trees to be explored; from this a "shared" tree structure leading to the minimization of device count can be developed.

Suppose a 1-cell (0-cell) representing the cube x_1P and a 0-cell (1-cell) representing the cube \bar{x}_1P simultaneously exist. Then the cell corresponding to the cube P is defined as a 10-cell (01-cell). These 01- or 10-cells act as individual cells of two different types. A 01-loop (10-loop) can be formed by encircling two or more adjacent 01-cells (10-cells).

With these concepts added, we revisit the previous example. The K-map shown in Fig. 3.11 has three types of encirclements, namely, 0-, 1- and 10-loop. The "shared" tree corresponding to the 10-loops is first constructed (Fig. 3.12), and then more branches corresponding to the 1-loops and 0-loops are added to form a complete DCVS/DSL tree (Fig. 3.13). Note that only eight n-devices are required, which is two devices fewer than for the disjointed tree in Fig. 3.10. However, the number of stacked levels has increased to three.

Generally, the reduction of the number of devices by tree sharing does not necessarily cause an increase in the number of stacked levels. In fact, the heuristic procedures that will be discussed tend to optimize both device count and number of stacked levels.

The K-map procedure consists of four steps:

- 1) Identify four different types of cells in the K-map, namely, 0-, 1-, 01- and 10-cells.
- 2) Find a minimal cover for all the 01-cells. Construct the tree corresponding to this minimal cover. The variable x_i 's in each of the tree branches are arranged from top to bottom in ascending

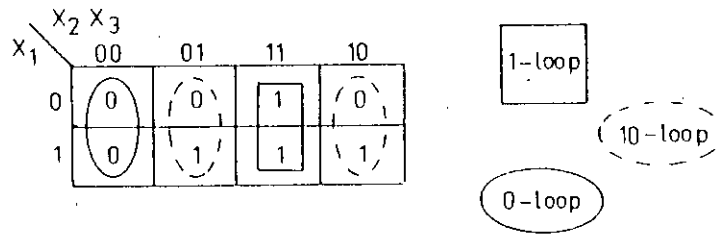


Fig. 3.11 The K-map of Fig. 3.9, but with different encirclements.

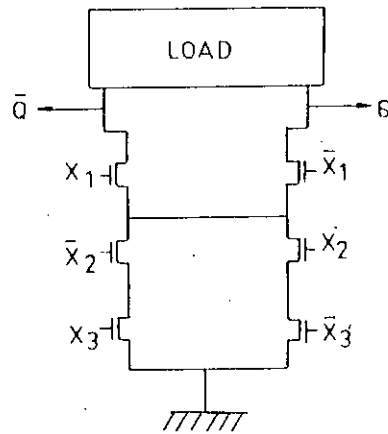


Fig. 3.12 The DCVS/DSL logic trees resulting from 10-loops of Fig. 3.11.

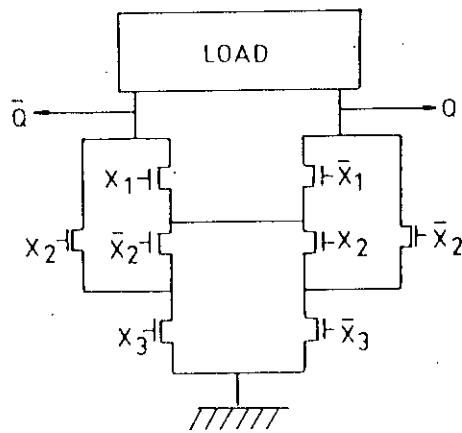


Fig. 3.13 The complete DCVS/DSL logic trees resulting from Fig. 3.11.

order with magnitude of i . Always construct tree branches corresponding to loops of smaller size first. The top pair of control inputs are x_1 associated with node Q , and \bar{x}_1 associated with node \bar{Q} . The sources of the transistors with gates x_1 and \bar{x}_1 are connected together.

3) From the prime implicants of all the 10-cells, find a minimal cover such that the tree constructed may share some of the branches with the tree in step 2. Contrary to step 2, the top pair of control inputs are \bar{x}_1 associated with node Q , and x_1 associated with node \bar{Q} .

4) Find a minimal cover for the remaining 0-cells and 1-cells. While constructing the tree, always look for the sharing of tree branches. The root of the 0-tree (1-tree) is connected to node Q (node \bar{Q}).

This procedure may create different tree structures if x_i 's are permuted (e.g., x_1 and x_2 variables are interchanged). Also, there may be several ways to choose a minimal cover and to share tree branches.

As an example, given a four-variable K-map as shown in Fig. 3.14, application of steps 1 and 2 generates the tree structure in Fig. 3.15. Further, applying step 3 generates the complete DCVS tree in Fig. 3.16. Step 4 has been skipped because there are no remaining 0-cells and 1-cells.

A different way of encircling the K-map, as shown in Fig. 3.17, leads to a different tree structure as shown in Fig. 3.18. Note that the 10-cells are not covered minimally in this manifestation, and thus the stack level in some of the tree branches is increased. This undesirable feature, combined with the large parasitic capacitances associated with the numerous shared source

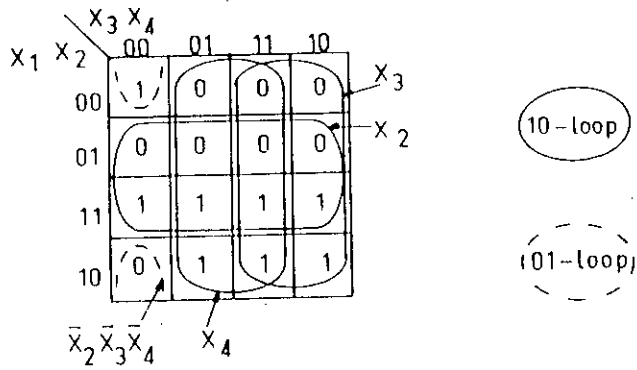


Fig. 3.14 The K-map for the function $Q = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1(x_2 + x_3 + x_4)$ showing the 10 and 01 encirclements.

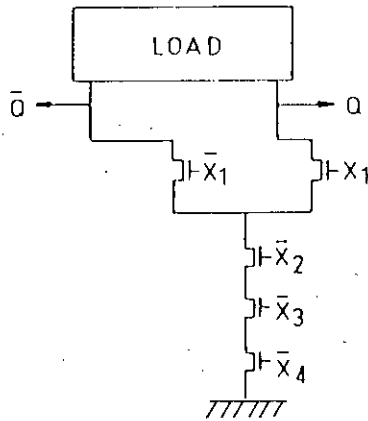


Fig. 3.15 DCVS/DSL logic tree corresponding to the 01-loops of K-map shown in Fig. 3.14.

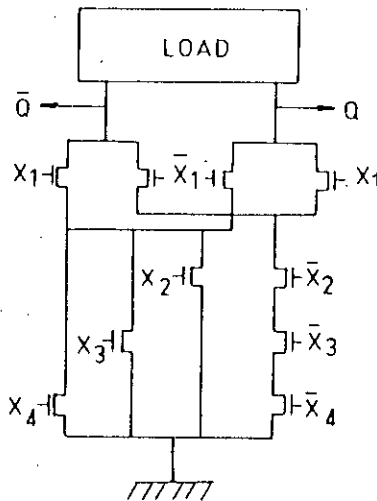


Fig. 3.16 The complete DCVS/DSL logic trees for the function $Q = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1(x_2 + x_3 + x_4)$.

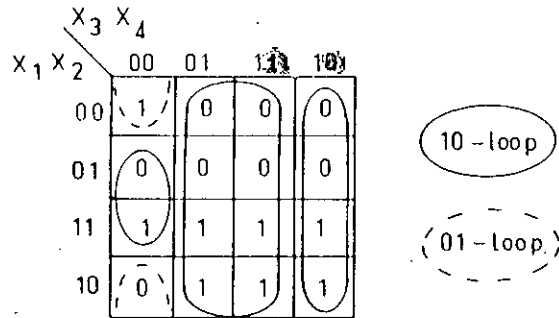


Fig. 3.17 An alternative encirclement of the K-map of Fig. 3.14.

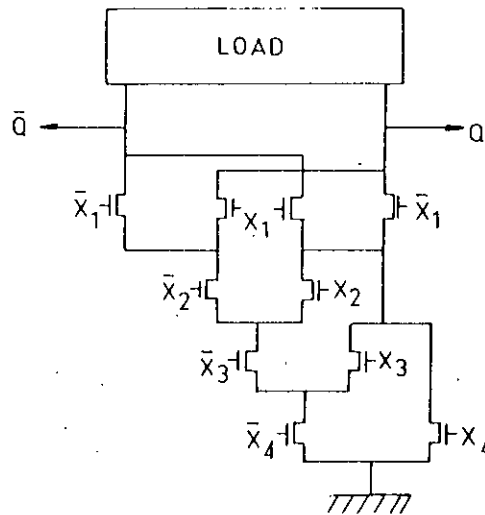


Fig. 3.18 The DCVS/DSL trees resulting from Fig. 3.17.

and drain connections, indicates that the circuit of Fig. 3.16 would have superior electrical performance to that of the circuit Fig. 3.18.

3.3.2 Device Sizing for DSL Circuits

Propagation delay is found to be minimum at a V_{ref} of about $V_{DD}/2 + V_{thn}$, where V_{thn} is the threshold voltage of n-device. But with this optimum value of V_{ref} , logic low voltage levels (V_L) at both the I/O nodes and internal nodes are not exactly zero volt. Due to high quiescent current at optimum V_{ref} , the logic low voltages at both nodes are higher than zero volt. If V_L at the internal nodes is close to the threshold voltage of the n-transistors (V_{thn}), it may lead to faulty logic operation. Therefore, care must be taken to select the appropriate sizes of both n- and p- transistors so that V_L at the internal nodes are much below the V_{thn} . A reasonable value of V_L is one-third of V_{thn} [8]. Referring to Fig. 2.13 of DSL basic circuit, logic low voltage at f (\bar{f}) can be reduced by decreasing the width of the p-transistor p2 (p1) or by increasing the width of the n-transistor n2 (n1). Decreasing the width of p2 (p1) causes its resistance to increase and thus causes lower current to flow through it; so reduced voltage is obtained at node f (\bar{f}). On the other hand, increasing the width of n2 (n1) causes its resistance to decrease and thus causes lower voltage drop across it for a certain current; so reduced voltage is obtained at node f (\bar{f}).

Chapter 4

Performance Analysis of Various CMOS Logic Families

4.1 Introduction

In this chapter, a detailed analysis of the performance of the conventional static CMOS and various differential CMOS circuits is carried out. Two forms of static differential CMOS circuits are analyzed, namely, DCVS logic and DSL logic. All these circuits are simulated to determine the propagation delay, static power dissipation, logic low and high voltage levels. The results from various circuit techniques are compared for varying device sizes and conditions of operation.

4.2 Test Arrangement

To analyze and compare the performance of various CMOS circuit techniques, a cascaded chain of ten 2-input NAND gates are used as shown in Fig. 4.1. Note that one of the inputs of each NAND gate in this figure is tied internally to a logic high level of 5 Volt. The other input is connected to the output of preceding gate. Unlike conventional static CMOS circuits, the differential

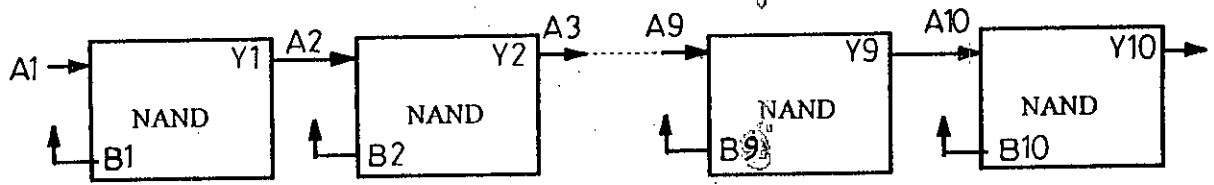


Fig. 4.1 Cascaded chain of ten 2-input NAND gates

CMOS circuits require complementary inputs and produce complementary outputs. Hence, for differential CMOS circuits, the input and outputs lines of Fig. 4.1 represent complementary inputs and outputs respectively. The circuits are simulated using a public domain version of SPICE (spice3e). Level 2 model parameters for a 2-layer metal 1.5 μm n-well CMOS process are used for all simulations. These model parameters are given in Appendix A. All simulations are carried out at the temperature of 27°C (default temperature in spice3e). The length of all devices is kept to the process minimum, i.e., 1.6 μm for all simulations, unless otherwise specified. All circuits are analyzed for various device aspect ratios (W/L ratios) by using variable device widths.

A structured approach is adopted for simulation of the cascaded chain of Fig. 4.1. Description of only one NAND gate of Fig. 4.1 is written using the subcircuit function of SPICE. Then the whole cascaded chain of Fig. 4.1 is modeled in a main SPICE program by making 10 references to the subcircuit. In all the simulations, propagation delay between the output of the first gate and the output of the ninth gate of Fig. 4.1 is measured. Static power dissipation is measured for all the ten gates.

4.3 Analysis of Conventional Static CMOS Logic

The circuit diagram of a 2-input NAND gate using conventional static CMOS logic is shown in Fig. 4.2. The digits represent the node number assignments for SPICE subcircuit description which is given in Appendix B.

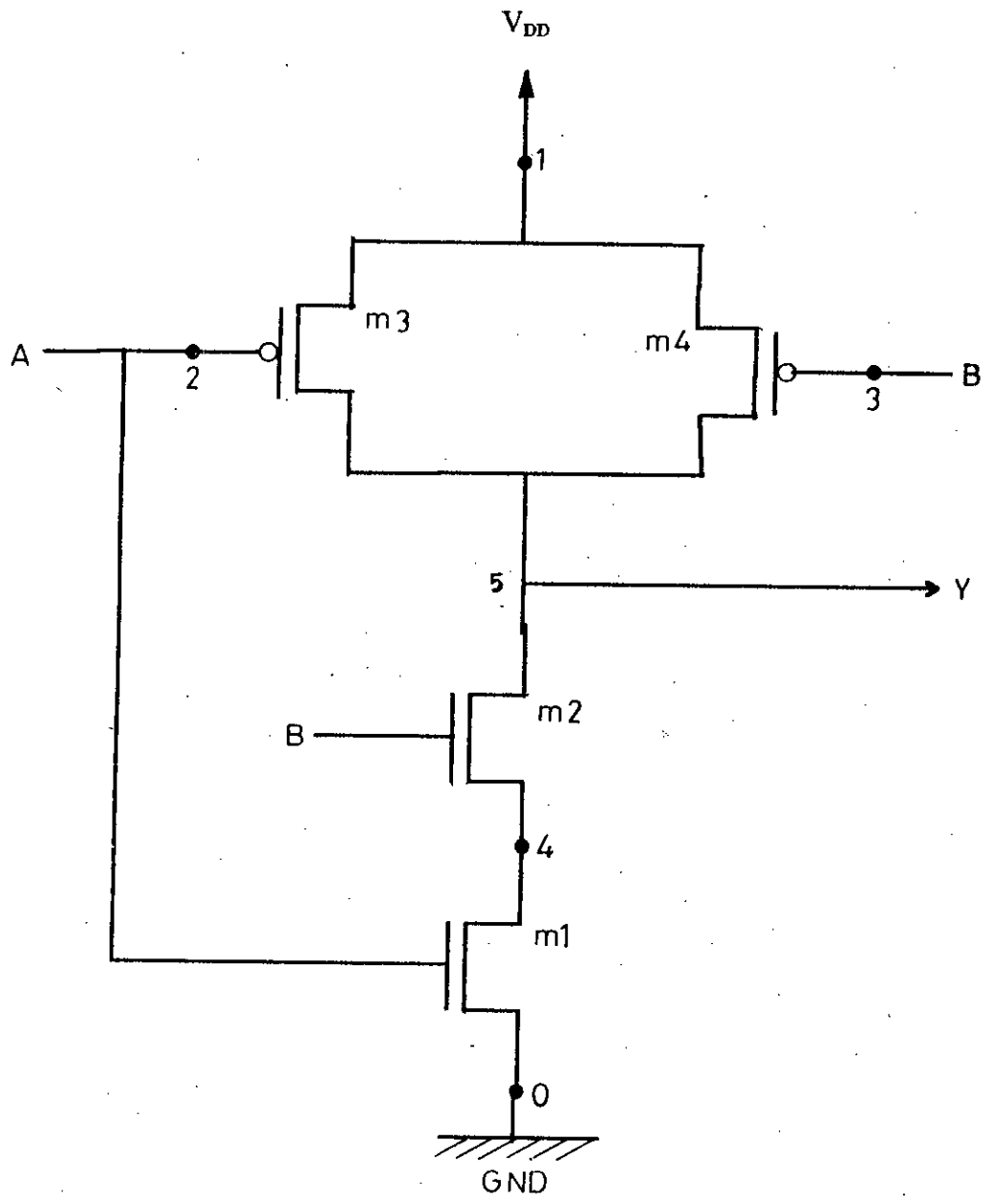


Fig. 4.2 2-input static CMOS NAND gate.

The results of SPICE simulation on the cascaded chain of Fig. 4.1 using the static CMOS NAND gate of Fig. 4.2 are given in Table 4.1. The notations W_n and W_p refers to the widths of the n- and p- devices respectively in μm . From the results, it is clear that static power dissipation of conventional static CMOS circuits is about tenths of nanowatt for $V_{DD} = 5$ volt.

Table 4.1: Simulated performance of conventional static CMOS circuit

Propagation Delay t_d (nsec)	Static Power Dissipation P_{ss} (nwatt)	Logic Low Output Voltage V_l (volt)	Logic High Output Voltage V_h (volt)	Dimensions (μm)
1.99	0.795	0	5	$W_n=W_p=2$
1.86	0.795	0	5	$W_n=5, W_p=4$

The logic low output voltage is zero volt and logic high output voltage is 5 volt (for $V_{DD} = 5$ volt) for each of the two sets of transistor dimensions. So, conventional static CMOS circuit technique gives correct output levels regardless of gate geometry ratios of both the load and driver transistors. Thus, conventional static CMOS is a ratioless logic family. This is because the load and driver transistors are never on simultaneously under steady-state conditions. Also, it is clear from the values of logic low and high voltage levels that static CMOS logic has a high noise immunity.

4.4 Analysis of Static Differential Cascode Voltage

Switch (DCVS) Logic

The circuit diagram of a 2-input static DCVS NAND gate is shown in Fig. 4.3. This circuit is used as the basic block in the cascaded NAND chain of Fig. 4.1. The digits in Fig. 4.3 represent the node number arrangements for SPICE subcircuit description which is given in Appendix C.

The simulation results of the cascaded chain of Fig. 4.1 using the static DCVS NAND gate of Fig. 4.3 are shown in Table 4.2. It is clear that the static power dissipation of the static DCVS chain is very low and comparable to that obtained for the static CMOS chain. Also, the output logic low and high voltage levels are shown to be zero volt and 5 volt respectively. Thus, static DCVS circuits retain the high noise immunity of static CMOS circuits shown earlier. However, as shown in Table 4.2, the propagation delay through a cascaded chain of 8 DCVS NAND gates is approximately double the delay obtained for the static CMOS chain.

Table 4.2: Simulated performance of static DCVS circuit

Propagation Delay t_d (nsec)	Static Power Dissipation P_{ss} (nwatt)	Logic Low Output Voltage V_l (volt)	Logic High Output Voltage V_h (volt)	Dimensions (μm)
4.17	1.14	0	5	$W_n=W_p=2$
4.45	1.45	0	5	$W_n=5, W_p=4$

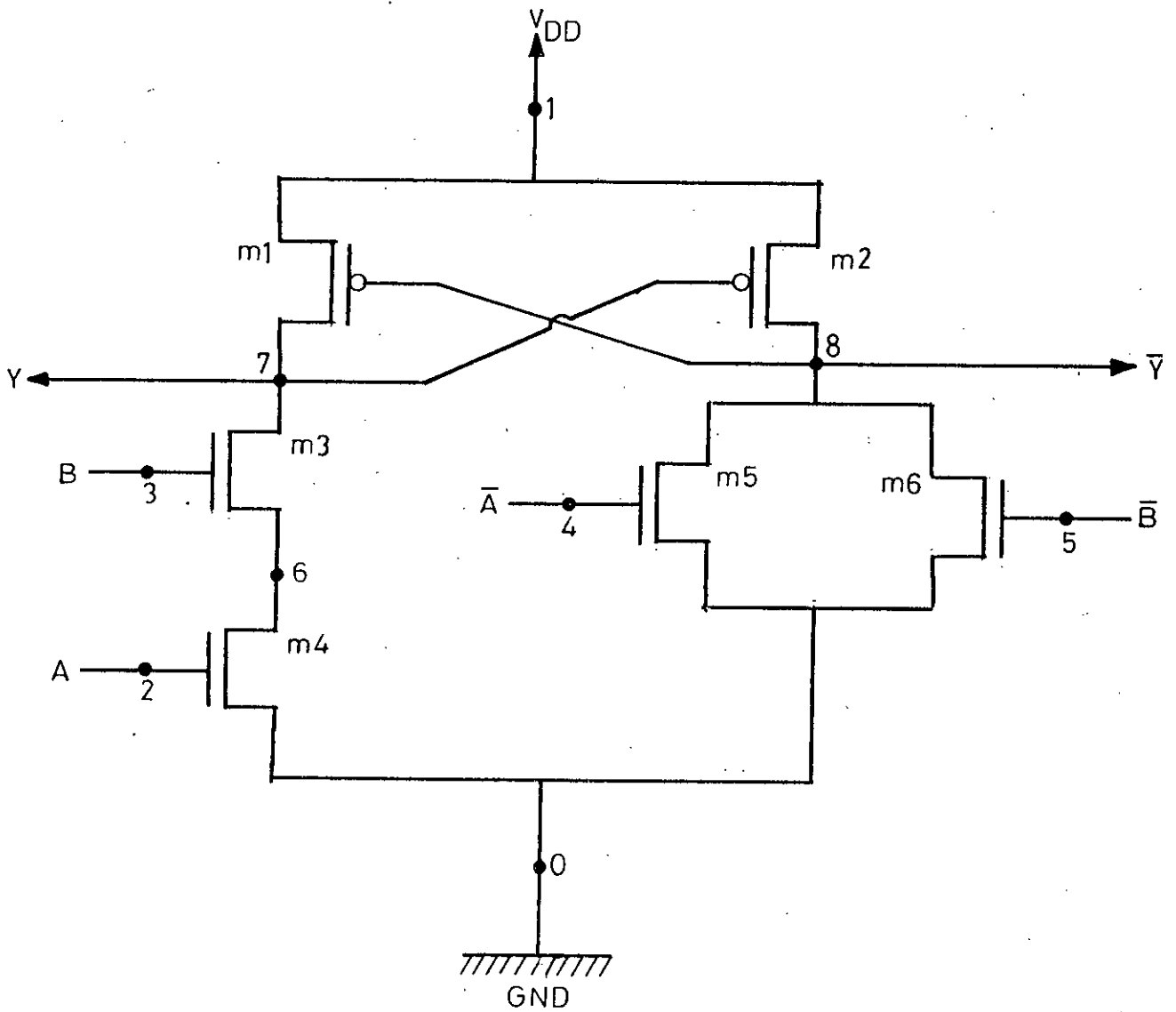


Fig. 4.3 2-input static DCVS NAND gate.

4.5 Analysis of Differential Split-Level (DSL) Logic

The circuit diagram of a 2-input DSL NAND gate is shown in Fig. 4.4. Note that the complementary input and output nodes of this gate are open source and drain respectively. However, when connected in a cascaded chain as in Fig. 4.1, the open drain outputs of one gate are loaded by the cross-coupled nMOS-pMOS load at the inputs of the next gate. The only exceptions are at the input of the first gate and the output of the last gate. An extra cross-coupled nMOS-pMOS load is connected at the output of the tenth gate of Fig. 4.1. The complementary signals are applied to the open source inputs of the first gate of Fig. 4.1 using two nMOS transistors as shown in Fig. 4.5. Thus, the circuit shown in Fig. 4.5 acts as an interface between conventional static CMOS and DSL logic.

The simulation results of the cascaded chain of NAND gates of Fig. 4.1 using the DSL gate of Fig. 4.4 as the basic block are given in Table 4.3. It shows the simulation results on propagation delay through eight NAND gates, output logic low and high voltage levels and static power dissipation at various reference voltages ranging from 2.5 volt to 5 volt, i.e., full power supply voltage. Apart from the simulation results for device lengths equal to the process minimum, i.e., 1.6 μm , results are also obtained for the logic n-transistor's length equal to about half the process minimum. As explained in

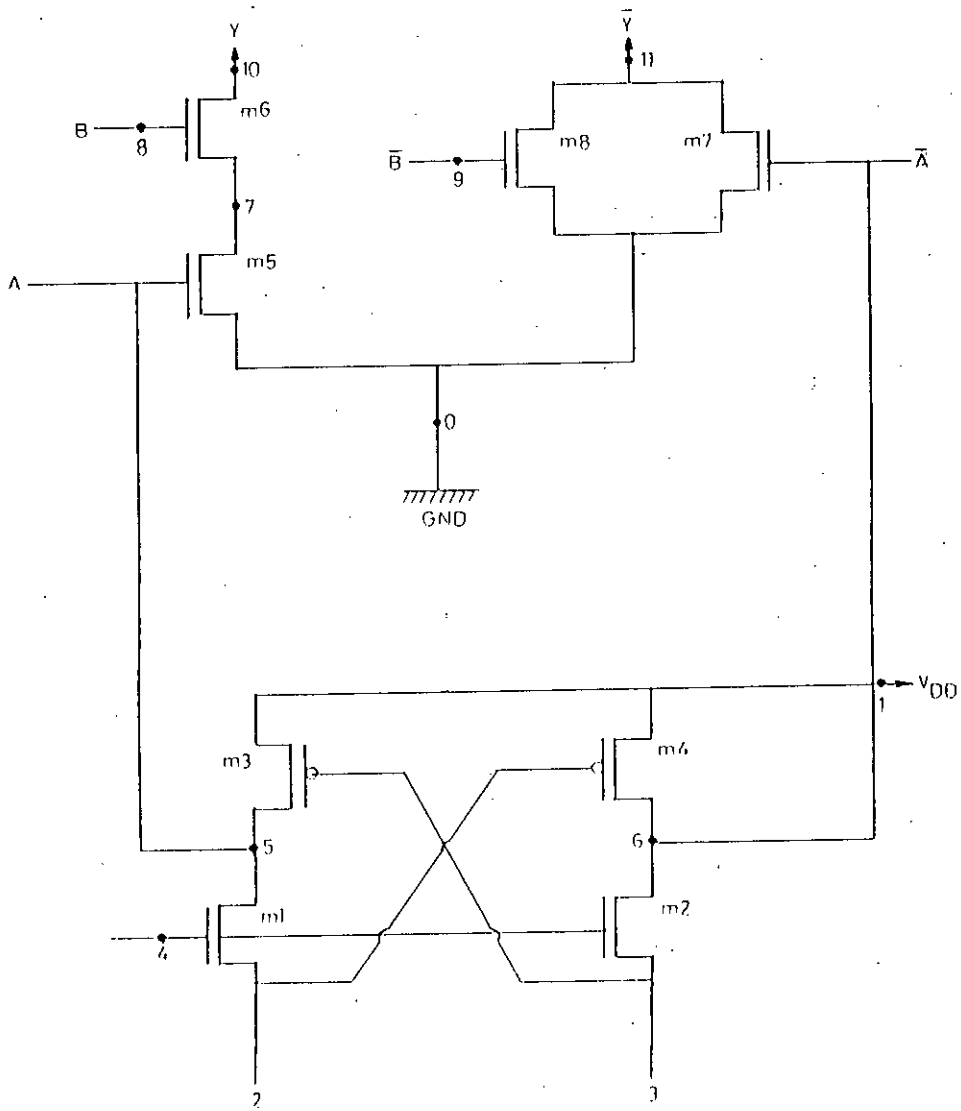


Fig. 4.4 2-input DSL NAND gate.

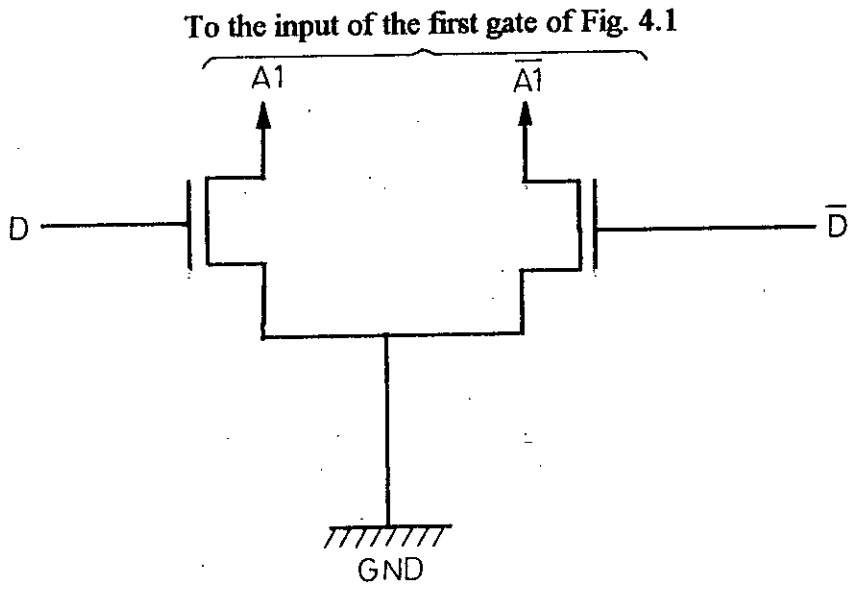


Fig. 4.5 Interface between conventional static CMOS and DSL logic.

Table 4.3: Simulated performance of DSL circuit

Set No.	V_{ref} (Volt)	t_d (nsec)	P_{ss} (mW)	$V_{l(int)}$ (mV)	$V_{h(int)}$ (Volt)	$V_{l(o)}$ (mV)	$V_{h(o)}$ (Volt)	Dimensions (μm)
Set 1	5.0	3.70	0.0083	1	5	1	4.00	$L_n=L_p=1.6$ $W_n=W_p=2$ (for all devices)
	4.5	3.20	0.397	80	5	52	3.45	
	4.0	2.78	1.08	230	5	141	3.06	
	3.5	2.47	1.96	465	5	260	2.60	
	3.0	2.56	3.37	886	5	430	1.85	
Set 2	5.0	5.35	0.0027	0	5	0	4.01	$L_n=L_p=1.6$ (for all devices) $W_{m1}=6, W_{m2}=4$ $W_{m3}=2, W_{m4}=2$ $W_{m5}=6, W_{m6}=6$ $W_{m7}=4, W_{m8}=4$
	4.5	4.80	0.360	26	5	17	3.50	
	4.0	4.20	1.08	78	5	50	3.07	
	3.5	3.62	1.96	157	5	91	2.60	
	3.0	3.20	3.00	272	5	141	2.16	
	2.5	3.30	4.10	450	5	191	1.70	
Set 3	5.0	1.97	0.0027	0	5	0	4.03	$L_{m1}=L_{m2}=L_{m3}=L_{m4}=1.6$ $L_{m5}=L_{m6}=L_{m7}=L_{m8}=0.88$ $W_{m1}=6, W_{m2}=4$ $W_{m3}=2, W_{m4}=2$ $W_{m5}=6, W_{m6}=6$ $W_{m7}=4, W_{m8}=4$
	4.5	1.80	0.365	17	5	5	3.55	
	4.0	1.67	1.05	55	5	15	3.08	
	3.5	1.55	1.94	110	5	28	2.60	
	3.0	1.45	3.225	194	5	51	2.16	
	2.5	1.57	6.6	414	4.73	79	1.39	

chapter 3, this reduction of logic n-transistor's channel length to below the process minimum is possible at the optimum V_{ref} of approximately 3.5 Volt, since the maximum drain to source voltage (V_{DSmax}) of these transistors is only $V_{DD}/2$ at this value of reference voltage.

It is seen from Table 4.3 that the static power dissipation in the DSL chain is much higher than those in conventional static CMOS and static DCVS chain (as given in Tables 4.1 and 4.2 respectively). However, the static power dissipation (P_{ss}) reduces with increasing V_{ref} . This is depicted in Fig. 4.6 for the results of Set 1 in Table 4.3.

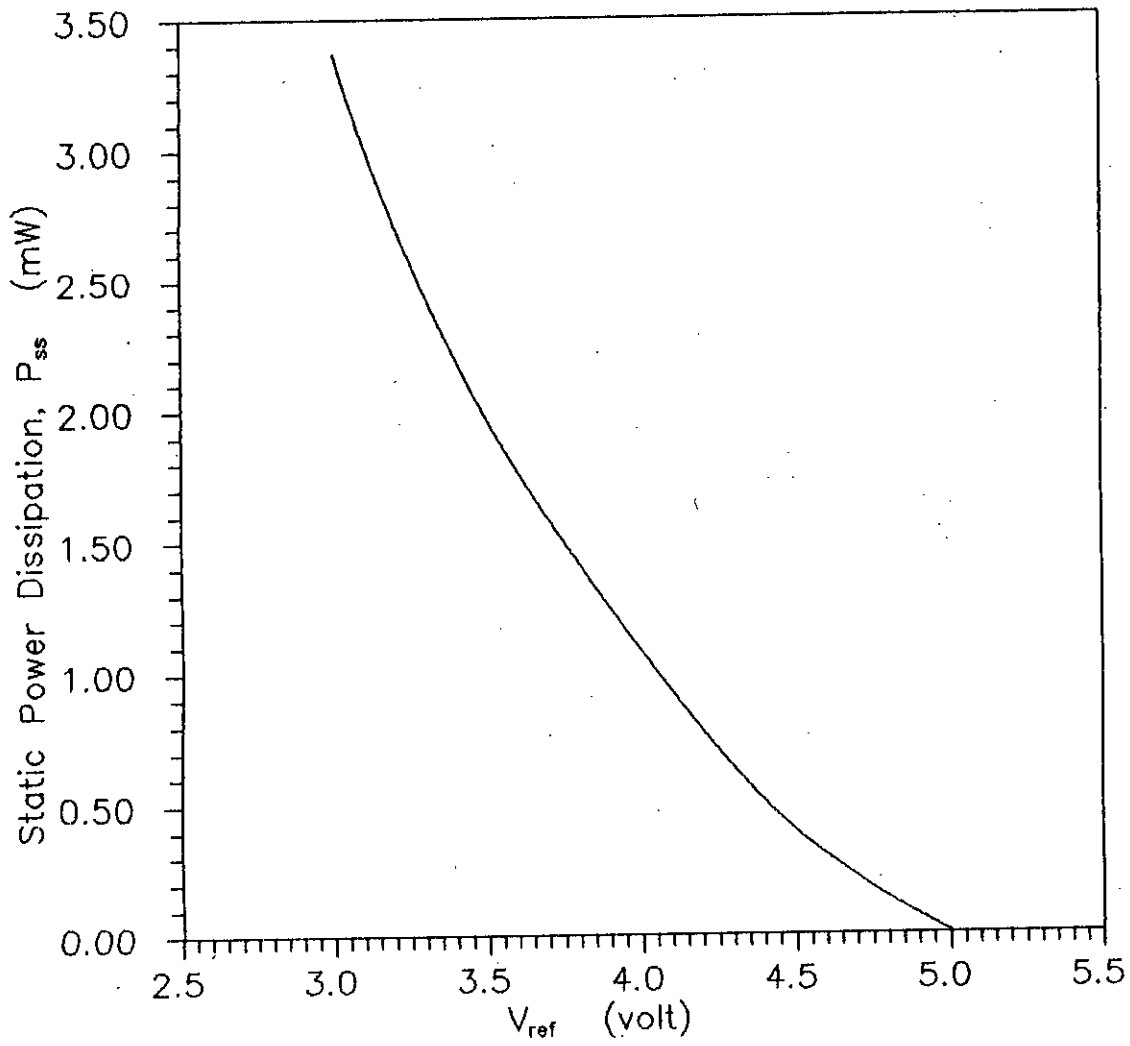


Fig. 4.6 Static power dissipation as a function of reference voltage

The logic low voltage levels from the results of Set 1 in Table 4.3 at both the internal nodes and I/O nodes of the NAND gate of Fig. 4.4 are plotted against V_{ref} as shown in Fig. 4.7 and Fig. 4.8 respectively. It is clear that the logic low voltage levels reduce with V_{ref} thereby providing higher noise immunity with increasing V_{ref} .

The variation of logic high voltage level, obtained from the results of Set 1, in Table 4.3 at the I/O nodes of the DSL NAND gate (Fig. 4.4) is shown in Fig. 4.9. It shows that the logic high voltage at the I/O nodes increases with increasing V_{ref} . Again, the noise immunity improves with increasing V_{ref} . Note that the logic high voltage at the internal nodes of the NAND gate is independent of V_{ref} and is equal to the power supply voltage of 5 volt as shown in Table 4.3.

The variation of propagation delay (as obtained from the results of Set 1 in Table 4.3) across eight DSL NAND gates is plotted against V_{ref} in Fig. 4.10. This shows that the propagation delay is minimum at a V_{ref} of about 3.5 volt which is approximately equal to the optimum V_{ref} of $(V_{DD}/2+V_T)$ for $V_{DD} = 5$ volt.

As explained earlier, the results confirm that although the static power dissipation reduces and noise immunity improves when V_{ref} is increased beyond the optimum value of 3.5 volt, the propagation delay increases.

For minimum geometry devices, the minimum delay of 2.5 nsec at $V_{ref} = 3.5$ volt (shown in Fig. 4.10) is 25% higher than the delay for conventional static CMOS chain (1.99 ns) as shown in Table 4.1. However, as shown in Set 3 of Table 4.3, when the channel lengths of the logic n-

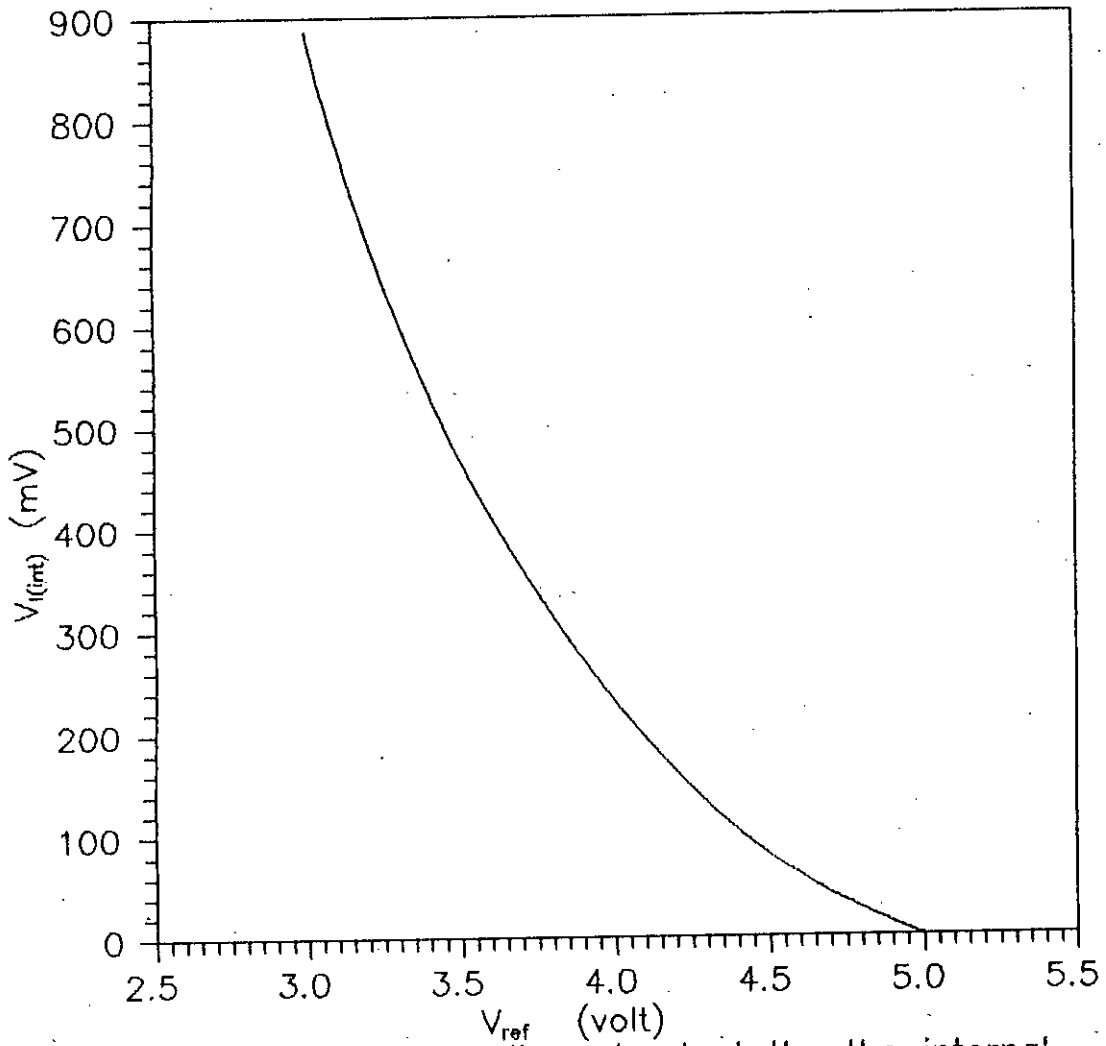


Fig. 4.7 Logic low voltage level at the the internal nodes as a function of reference voltage

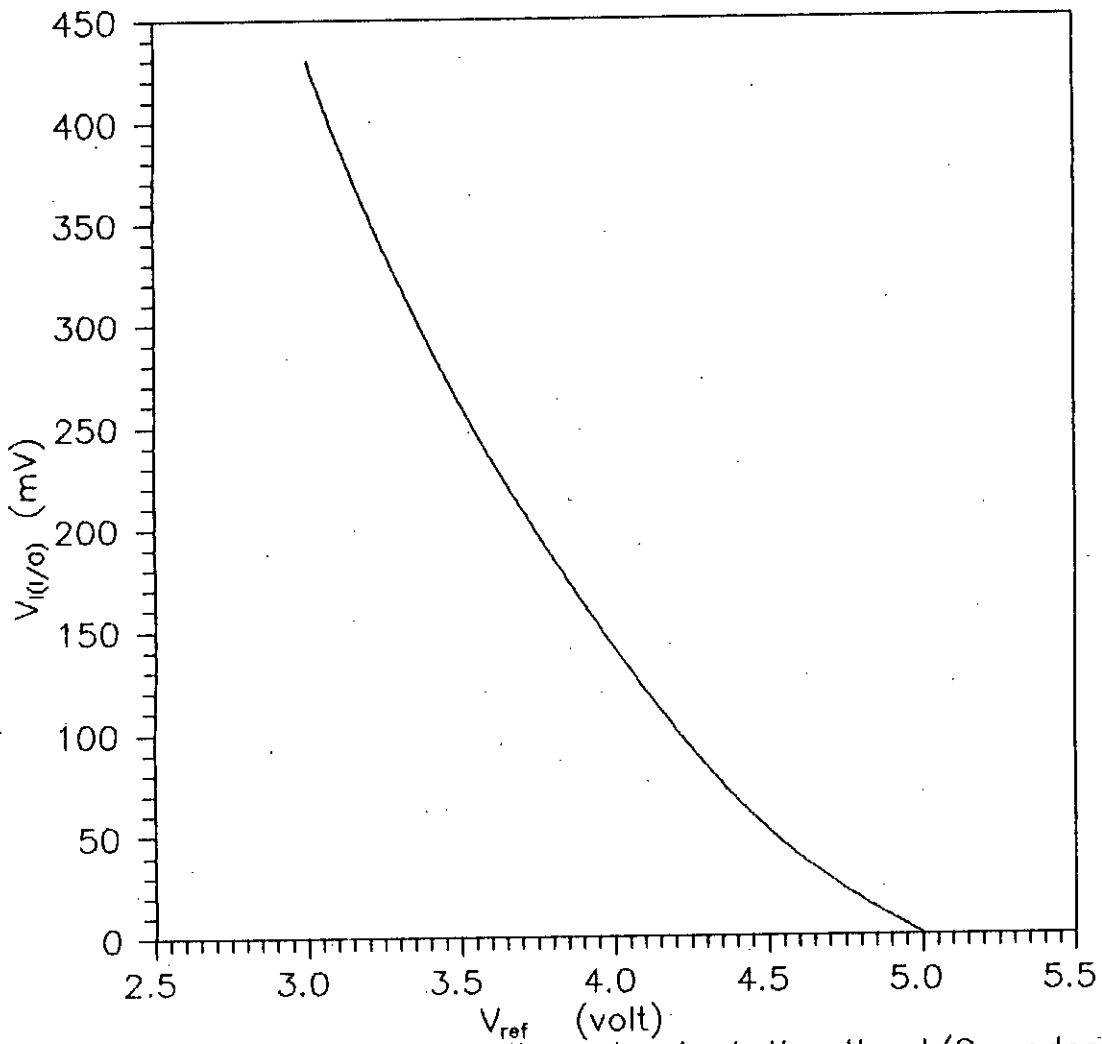


Fig. 4.8 Logic low voltage level at the the I/O nodes as a function of reference voltage

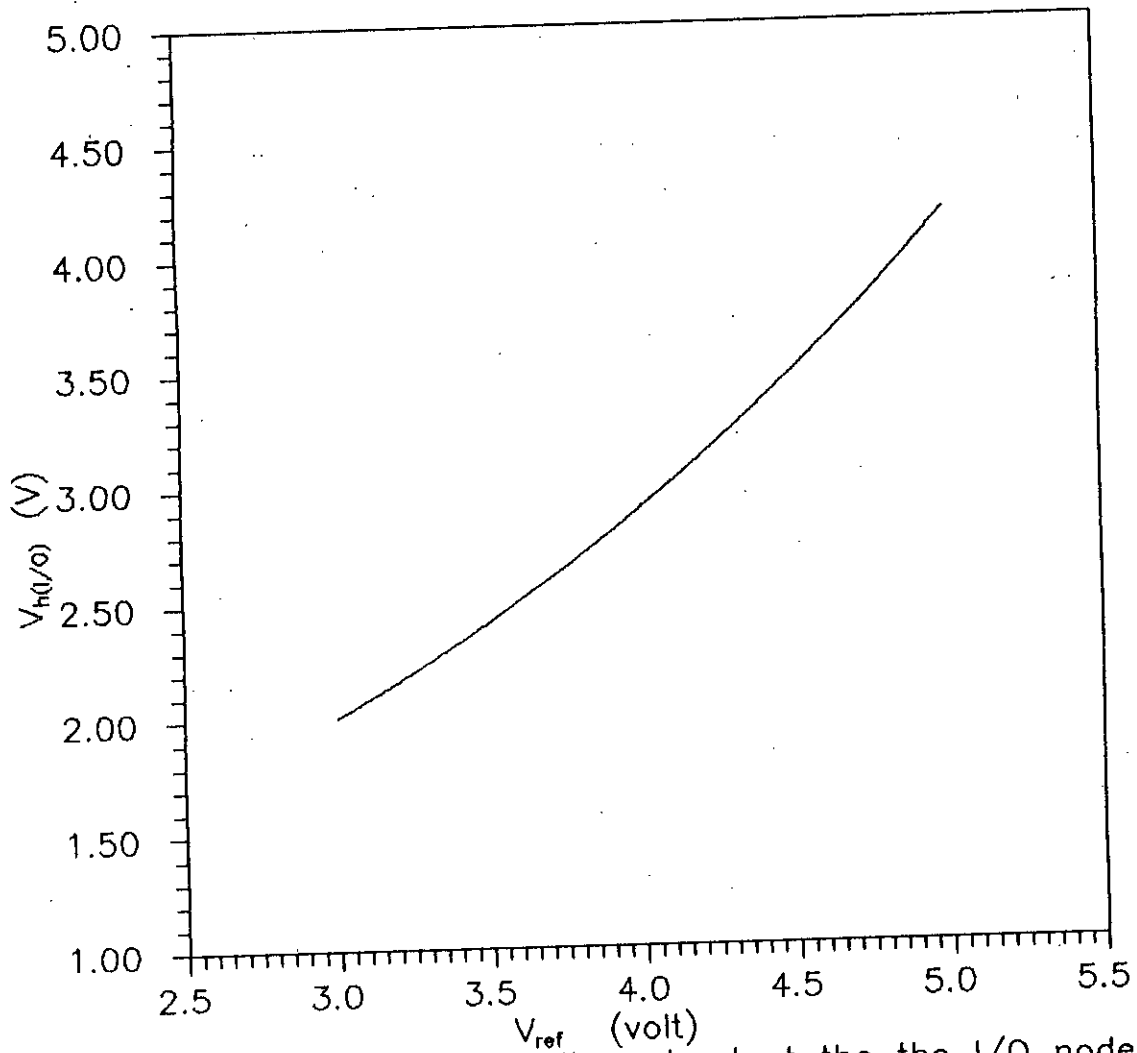


Fig. 4.9 Logic high voltage level at the the I/O nodes as a function of reference voltage



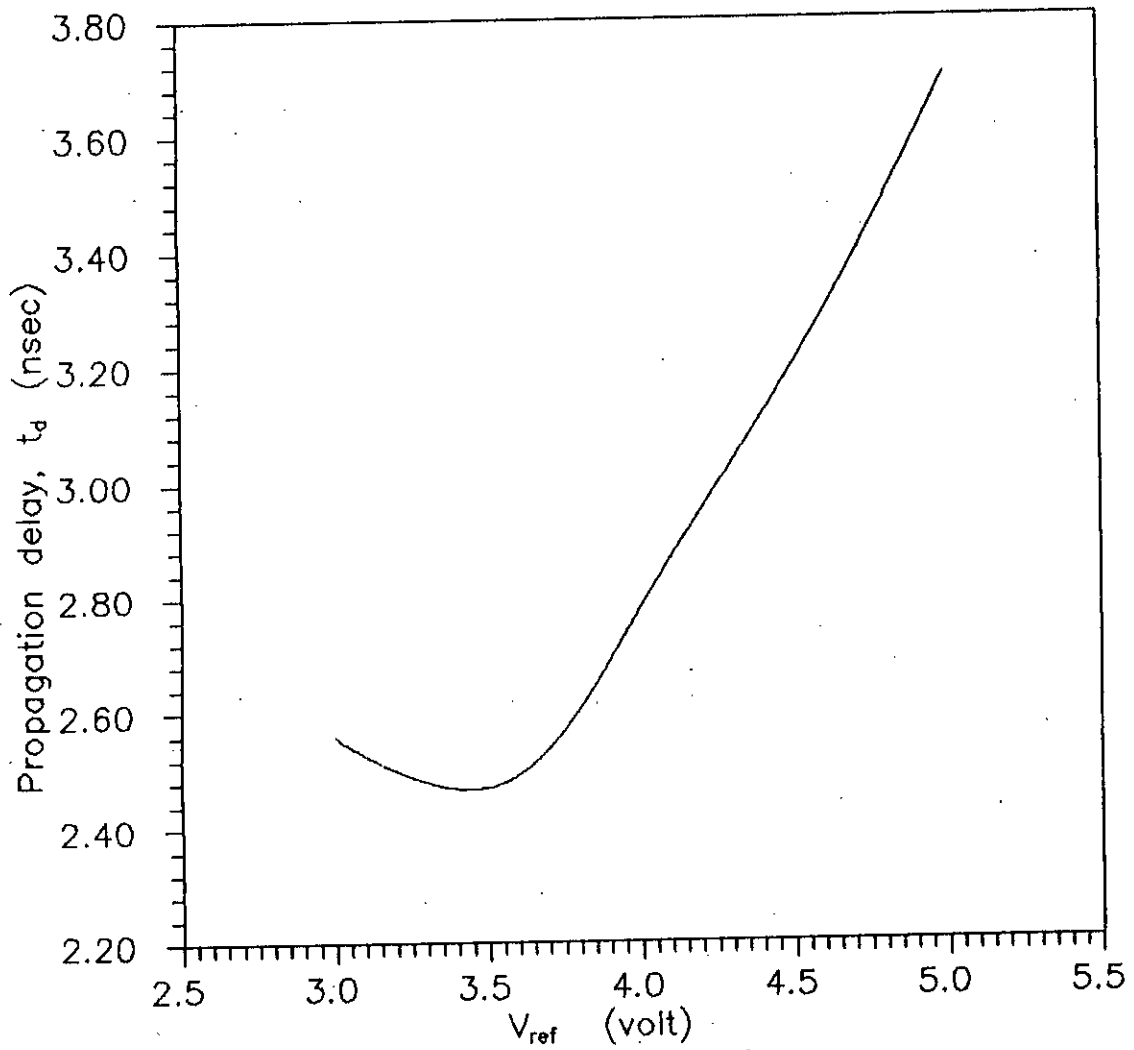


Fig. 4.10 Variation of propagation delay with reference voltage

transistors in Fig. 4.4 is reduced to $0.88 \mu\text{m}$ (about half the process minimum of $1.6 \mu\text{m}$) then the delay at $V_{\text{ref}}=3.5$ volt is only 1.55 nsec. This is about 75% of the delay obtained in static CMOS case as shown in Table 4.1.

4.6 Comparison of the CMOS Logic Families

From the discussions of Sections 4.3 to 4.5 it is clear that conventional static CMOS and static DCVS logic have negligible static power dissipation and very high noise immunity. However, the propagation delay in static DCVS is more than double the amount of delay in conventional CMOS. This contradicts with the comments made in the original paper [3] about the speed advantage of static DCVS over conventional CMOS. Clearly, the results obtained in the present work shows that the problem of output settling in static DCVS circuits outweighs its advantage of having lower input gate capacitance loading compared to conventional CMOS. It was shown in [5] that static DCVS is slightly slower than conventional static CMOS. However, the results presented in this thesis shows that static DCVS is more than 2 times slower.

The most important results presented in this thesis are those on DSL logic. In the original paper on DSL logic [6], it was claimed that DSL would be 5 times faster than conventional CMOS even if channel lengths of the logic n-transistors are not reduced. However, the results presented in Sections 4.3 to 4.5 shows that even at the optimum reference voltage, DSL logic is slightly slower than conventional CMOS. However, when the channel lengths of the n-transistors in the logic trees

are reduced to almost half the process minimum, then DSL logic is slightly faster than standard CMOS. Moreover, reducing the channel length to half the process minimum means that the logic swing at the I/O nodes of DSL gates cannot be higher than $V_{DD}/2$ in order to avoid the risk of drain-to-source punchthrough [6]. Therefore, the reference voltage can not be increased to more than the optimum value of $(V_{DD}/2+V_T)$. Thus, the idea of keeping the quiescent power low by making V_{ref} equal to V_{DD} during inactive periods as proposed in [7] can not be implemented if n-devices of reduced lengths are used. Nevertheless, at the optimum V_{ref} (3.5 volt in this case), the quiescent power dissipation of DSL circuits is very high compared to other CMOS logic families. Also, the noise immunity at optimum V_{ref} is quite low.

89348

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

Static DCVS logic was claimed to be much faster than conventional static CMOS [3]. However, it was shown in [5] that static DCVS is slightly slower than static CMOS. But the results obtained in this thesis shows that static DCVS is more than 2 times slower than static CMOS. This is a finding not reported before.

DSL logic was shown to be faster than static CMOS by all previous researchers [5], [6]. However, the results given in this thesis have shown that this is not the case. It was claimed in both [3] and [5] that DSL is faster than static CMOS at optimum reference voltage even without the reduction of the channel lengths of the n-transistors in DSL logic trees, but the present work has shown that it is slower. When the channel lengths of the logic n-transistors are reduced to half the process minimum then DSL logic is shown to be slightly faster than conventional static CMOS. However, it was claimed in [6] that with reduction of channel length of the logic n-transistors, DSL logic can be 10 times faster than static CMOS.

The high quiescent power dissipation of DSL logic coupled with its low noise immunity and little or no speed advantage over static CMOS may not make it as attractive for VLSI implementation as claimed in the original paper [6]. However, it must be emphasized here that the

accuracy of the results presented in this thesis is limited by the accuracy of the models employed in spice3e which was used for all simulations carried out during the course of this work.

5.2 Future Work

Although static DCVS logic suffers from the disadvantage of high output settling times, its input gate capacitance loading is 2 to 3 times lower than static CMOS [3]. Hence, the fact that static DCVS logic is more than 2 times slower than static CMOS (as shown in this thesis) sounds unusually low. Also, despite the use of differential logic trees and the half V_{DD} swing at the I/O nodes of DSL gates at optimum V_{ref} , DSL logic has been shown to be slower than static CMOS in this thesis. As mentioned in the previous section, the accuracy of the results given in this thesis is limited by the accuracy of spice3e which is a public domain version of SPICE. Moreover, only level 2 model parameters were used in all the simulations carried out during the course of this work. Therefore, it would be useful to check the results obtained in this thesis using a commercial version of SPICE, preferably HSPICE. Also, higher level model parameters may be used in future simulations to account for higher order terms in the drain-to-source currents in MOS devices [1] thereby giving more accurate results.

The performance of other recently proposed CMOS logic families, such as Complementary Pass-Transistor Logic (CPL) [14] and Enhancement Source-Coupled Logic (ESCL) [15] may also be investigated.

With the tremendous increase in the complexity of VLSI circuits in recent years [16], testing has become an important aspect in the design of VLSI chips [17]. Therefore, investigations into the testability of various CMOS logic families would be another important area of research.

References

1. N. H. E. Weste, K. Eshraghian, "Principles of CMOS VLSI Design - A System Perspective," Addison-Wesley Publishing Company, Second Edition, 1993.
2. S. J. Harold, "An Introduction to GaAS IC Design," Prentice Hall, U.K., 1993.
3. G. Heller and W. R. Griffin, "Cascode voltage switch logic: A differential CMOS logic family," ISSCC Digest, pp. 16-17, February 1984.
4. K. Erdelyi, W. R. Griffin, and R. D. Kilmoyrer, "Cascode voltage switch logic design," VLSI DESIGN, pp. 78-86, October 1984.
5. K. M. Chu and D. L. Pulfrey, "A comparison of CMOS circuit techniques: Differential cascode voltage switch logic versus conventional logic," IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 4, pp. 528-532, August 1987.
6. M. G. Pfennings, W. G. J. Mol, J. J. J. Bastiaens, and J. M. F. Van Dijk, "Differential split-level CMOS logic for subnanosecond speeds," IEEE Journal of Solid-State Circuits, Vol. SC-20, NO. 5, pp. 1050-1055, October 1985.
7. S. M. Aziz and W. A. J. Walker, "Quiescent power reduction in differential split-level CMOS," Journal of Semicustom ICs, Vol. 9, No. 2, pp. 28-31, December 1991.
8. Linda E. M. Brackenbury, "Design of VLSI Systems -- A Practical Introduction," Macmillan Education Ltd., 1987.
9. Saburo Muroga, "VLSI System Design," John Wiley and Sons, 1982.

10. Amar Mukherjee, "Introduction to nMOS and CMOS VLSI Systems Design," Prentice-Hall International Editions, 1986.
11. R. K. Brayton and C. McMullen, "The decomposition and factorization of Boolean expressions," in Proceedings of IEEE Int. Symp. Circuits Syst. (Rome, Italy), 1982, pp. 49-54.
12. K. M. Chu and D. L. Pulfrey, "Design procedures for differential cascode voltage switch circuits," IEEE Journal of Solid-State Circuits, Vol. SC-21, No. 6, pp. 1082-1087, December 1986.
13. S. Muroga, *Logic Design and Switching Theory*. New York, Wiley, 1979, pp. 163-180.
14. K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi and A. Shimizu, "A 3.8-ns 16×16-b multiplier using complementary pass-transistor logic," IEEE Journal of Solid-State Circuits, Vol. 25, No. 2, pp. 388-395, April 1990.
15. Maleki and S. Kiaei, "Enhancement source-coupled logic for mixed-mode VLSI circuits," IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 39, No. 6, June 1992.
16. Alpert and D. Avnon, "Architecture of the Pentium microprocessor," IEEE Micro, pp. 11-21, June 1993.
17. Williams and K. Parker, "Design for testability-a survey," IEEE Trans. on Computers, Vol. C-31, pp. 2-15, January 1982.

Appendices

Appendix A

Spice model parameters for a 2-layer metal 1.5 μm n-well CMOS process

```
.model nenh nmos LEVEL=2 LD=0.325U TOX=250E-10 NSUB=2E+16 VTO=0.7  
UO=510 UEXP=0.22 +UCRIT=24.3K DELTA=0.4 XJ=0.4U VMAX=54K NEFF=4.0  
RSH=55 JS=2U CJ=130U CJSW=620P +MJ=0.53 MJSW=0.53 PB=0.68+CGDO=320P  
CGSO=320P
```

```
.model penh pmos LEVEL=2 LD=0.3U TOX=250E-10 NSUB=5E+16 VTO=-0.1  
UO=210 UEXP=0.33 +UCRIT=51K DELTA=0.4 XJ=0.5U VMAX=47K NEFF=0.88  
RSH=75 JS=10U CJ=490U CJSW=590P +MJ=0.46 MJSW=0.46 PB=0.78 CGDO=320P  
CGSO=320P
```

Appendix B

Spice description of a cascaded chain of ten 2-input NAND gates using conventional static CMOS logic

```
*main circuit description
CASCADED CMOS NAND GATES (10 stages)
```

```
*File name is nand.mos
```

```
vdd 1 0 5
```

```
.options acct reld=0.0001
```

```
x1 1 0 10 20 cmosnand
```

```
x2 1 0 20 30 cmosnand
```

```
x3 1 0 30 40 cmosnand
```

```
x4 1 0 40 50 cmosnand
```

```
x5 1 0 50 60 cmosnand
```

```
x6 1 0 60 70 cmosnand
```

```
x7 1 0 70 80 cmosnand
```

```
x8 1 0 80 90 cmosnand
```

```
x9 1 0 90 100 cmosnand
```

```
x10 1 0 100 110 cmosnand
```

```
vin 10 0 pulse(0 5 10n 2n 2n 23n 50n)
```

```
.include subnand.mos
```

```
.include ../model
```

```
.tran 0.5n 60n
```

```
.print tran v(20) v(100) i(vdd)
```

```
.plot tran v(20) v(100) i(vdd)
```

```
.end
```

```
* Subcircuit description for single 2- input NAND gate (using minimum process lengths
* and widths)
```

```
.subckt cmosnand 1 0 2 5
```

```
* File name is subnand.mos
```

```
m1 4 2 0 0 nenh l=1.6u w=2u
```

```
m2 5 3 4 0 nenh l=1.6u w=2u
```

```
m3 5 2 1 1 penh l=1.6u w=2u
```

```
m4 5 3 1 1 penh l=1.6u w=2u
```



```
vb 3 0 5
.ends cmosnand
```

```
* Subcircuit description for single 2- input NAND gate (using lengths and widths for equal
* rise and fall times)
```

```
.subckt cmosnand 1 0 2 5
```

```
* File name is subnand.mos
```

```
m1 4 2 0 0 nenh l=1.6u w=4u
```

```
m2 5 3 4 0 nenh l=1.6u w=4u
```

```
m3 5 2 1 1 penh l=1.6u w=5u
```

```
m4 5 3 1 1 penh l=1.6u w=5u
```

```
vb 3 0 5
```

```
.ends cmosnand
```

Appendix C

Spice description of a cascaded chain of ten 2-input NAND gates using static DCVS logic

```
* main circuit description
CASCADED DCVS NAND GATES (10 STAGES)
*File name is nand.dcvns
vdd 1 0 5
.options acct reltol=0.001
x1 1 0 10 20 30 40 dcvnsnand
x2 1 0 30 40 50 60 dcvnsnand
x3 1 0 50 60 70 80 dcvnsnand
x4 1 0 70 80 90 100 dcvnsnand
x5 1 0 90 100 110 120 dcvnsnand
x6 1 0 110 120 130 140 dcvnsnand
x7 1 0 130 140 150 160 dcvnsnand
x8 1 0 150 160 170 180 dcvnsnand
x9 1 0 170 180 190 200 dcvnsnand
x10 1 0 190 200 210 220 dcvnsnand
.include subnand.dcvns
.include ../model
va 10 0 pulse(0 5 10n 2n 2n 23n 50n)
van 20 0 pulse(5 0 10n 2n 2n 23n 50n)
.tran 0.5n 60n
.print tran v(30) v(40) v(190) v(200) i(vdd)
.plot tran v(30) v(40) v(190) v(200) i(vdd)
.end
```

* Subcircuit description for single 2- input NAND gate (using minimum process lengths
* and widths)

```
.subckt dcvnsnand 1 0 2 4 7 8
*File name is subnand.dcvns
m1 7 8 1 1 penh l=1.6u w=2u
m2 8 7 1 1 penh l=1.6u w=2u
m3 7 3 6 0 nenh l=1.6u w=2u
```

```
m4 6 2 0 0 nenh l=1.6u w=2u
m5 8 4 0 0 nenh l=1.6u w=2u
m6 8 5 0 0 nenh l=1.6u w=2u
vb 3 0 5
vbn 5 0 0
.ends dcvsnand
```

* Subcircuit description for single 2- input NAND gate (using lengths and widths for equal
* rise and fall time)

```
.subckt dcvsnand 1 0 2 4 7 8
*File name is subnand.dcv
m1 7 8 1 1 penh l=1.6u w=4u
m2 8 7 1 1 penh l=1.6u w=4u
m3 7 3 6 0 nenh l=1.6u w=5u
m4 6 2 0 0 nenh l=1.6u w=5u
m5 8 4 0 0 nenh l=1.6u w=5u
m6 8 5 0 0 nenh l=1.6u w=5u
vb 3 0 5
vbn 5 0 0
.ends dcvsnand
```

Appendix D

Spice description of a cascaded chain of ten 2-input NAND gates using DSL logic

```
* main circuit description
CASCADED DSL NAND GATES (10 STAGES)
* File name is nand.dsl
vdd 1 0 5
.options acct reldtol=0.0001
mm1 10 13 0 0 nenh l=1.6u w=4u
mm2 20 14 0 0 nenh l=1.6u w=4u
x1 1 0 11 12 10 20 30 40 dslnand
x2 1 0 21 22 30 40 50 60 dslnand
x3 1 0 31 32 50 60 70 80 dslnand
x4 1 0 41 42 70 80 90 100 dslnand
x5 1 0 51 52 90 100 110 120 dslnand
x6 1 0 61 62 110 120 130 140 dslnand
x7 1 0 71 72 130 140 150 160 dslnand
x8 1 0 81 82 150 160 170 180 dslnand
x9 1 0 91 92 170 180 190 200 dslnand
x10 1 0 101 102 190 200 210 220 dslnand
.include subnand.dsl
.include ../model
va 13 0 pulse(0 5 10n 2n 2n 23n 50n)
van 14 0 pulse(5 0 10n 2n 2n 23n 50n)
.tran 0.5n 60n
.print tran v(30) v(40) v(190) v(200) i(vdd)
.plot tran v(30) v(40) v(190) v(200) i(vdd)
.end
```

* Subcircuit description for single 2-input NAND gate (using minimum process lengths and widths)

```
.subckt dslnand 1 0 5 6 2 3 10 11
* File name is subnand.dsl
m1 5 4 2 0 nenh l=1.6u w=2u
m2 6 4 3 0 nenh l=1.6u w=2u
```

m3 5 3 1 1 penh l=1.6u w=2u

m4 6 2 1 1 penh l=1.6u w=2u

m5 7 5 0 0 nenh l=1.6u w=2u

m6 10 8 7 0 nenh l=1.6u w=2u

m7 11 6 0 0 nenh l=1.6u w=2u

m8 11 9 0 0 nenh l=1.6u w=2u

vref 4 0 3.5

vb 8 0 5

vbn 9 0 0

.ends dslnand

* Subcircuit description for single 2-input NAND gate (using lengths and widths for equal
* rise and fall time)

.subckt dslnand 1 0 5 6 2 3 10 11

* File name is subnand.dsl

m1 5 4 2 0 nenh l=1.6u w=6u

m2 6 4 3 0 nenh l=1.6u w=4u

m3 5 3 1 1 penh l=1.6u w=2u

m4 6 2 1 1 penh l=1.6u w=2u

m5 7 5 0 0 nenh l=1.6 w=6u

m6 10 8 7 0 nenh l=1.6 w=6u

m7 11 6 0 0 nenh l=1.6 w=4u

m8 11 9 0 0 nenh l=1.6u w=4u

vref 4 0 3.5

vb 8 0 5

vbn 9 0 0

.ends dslnand

