Low-Power Digital CMOS VLSI Circuits and Design Methodologies

by

Muhammad M. Khellah

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

Waterloo, Ontario, Canada, 1999

0-612-44771-5

Abstract

While improving circuit's speed and reducing its area have been the primary figure of merits in digital VLSI design, more efforts are now spent on minimizing power dissipation. This is becoming equally true for both high-performance chips, such as microprocessors, to reduce cooling costs and improve reliability, as well as portable devices because of their limited energy budget offered by batteries. Over the past few years, the power problem was addressed on all fronts: process, circuits, gates, architectures and systems. This thesis continues this trend by proposing novel low-power techniques and design methodologies at the circuit, gate and architectural levels.

At the circuit level, two new low-swing schemes are presented. The first approach is based on charge sharing and can be used to reduce the swing and so power in dynamic digital circuits with high capacitive loads. Compared to conventional techniques, the proposed approach not only reduces power but also improves the speed as verified by both simulations and measurements. Three application circuits that benefit from this scheme are explored: internal bus lines, match lines in Contents-Addressable Memories (CAMs), and bit-lines in Read-Only Memories (ROMs). For each application, a test chip is fabricated and tested, and measurements have confirmed the functionality and high speed down to the low-voltage region of operation. The second low-swing circuit technique is based on current-injection. This approach is applied to the write and read operations in multi-port SRAM cell design. Simulations have shown the superiority of the approach in terms of both speed and power as compared to conventional ones. A test chip of a 3-port register file is fabricated and testing results have proved the applicability of the technique.

At the gate-level, we propose DVDV; a new automated design approach for reducing power dissipation in high-speed deep sub-micron CMOS Logic circuits. The main idea is to utilize a library of gates having Dual supply voltages (V_{dd}) and Dual threshold voltages (V_{th}), hence the name DVDV, to achieve high-speed at the lowest possible dynamic and leakage power dissipation. A simple algorithm for DVDV technology mapping is developed and implemented in C under the Berkeley's SIS-1.2 environment. Application to benchmark circuits shows large power savings and speed improvement as compared to using two supply voltages (and a single V_{th}) or two threshold voltages (and a single V_{dd}).

At the architectural-level, a method to characterize the effective capacitance in data path macros is developed. Given a library of hard-macros, a capacitance model based on multiple linear regression is derived for each macro. The capacitance models can be used later during architectural-level power estimation. The characterization methodology assumes no specific input data statistics, requires little knowledge about the module structure, allows the user to trade-off accuracy and characterization time, and propagates power from transistor-level (real) implementations. Simulation experiments on a set of data-path components show accuracy within 15% from a transistor-level tool on the average.

Acknowledgments

First and foremost, all thanks are due to God Almighty for the innumerable blessings through my life, one of which is this thesis.

My thesis supervisor, Professor Mohamed I. Elmasry, deserves a special acknowledgment for his vision, foresight, guidance and support throughout the research work.

I would like to express my sincere thanks to my parents who have instilled in me the courage to continue my education. I also extend my thanks to my sisters, my brother Fakhry, and all members of my family-in-law. My brother Fakhry has my gratitude for proofreading an early version of this dissertation.

I would also like to thank all my colleagues in the VLSI research group for their support, useful discussions, and on a personal note, for making the past four years a little more bearable. In particular, I would like to thank Amr Hafez, Ayman Elsayed, Mohamed Allam, Amr Wassal, and Alaa Elraey.

On the administrative side, I would like to acknowledge the VLSI system administrator, Phil Regier, for making things run smoothly. Special thanks goes to Wendy Boles, the graduate officer at the E&CE department, and Gehan Sabry, Prof. Elmasry's assistant, for their help with the administrative work.

Lastly, I am grateful to the National Science and Engineering Research Council (NSERC), Micronet, and the Ontario Graduate Scholarship (OGS) program, for the financial support of this work. Chip fabrication through the Canadian Micro-electronics Corporation (CMC) is greatly appreciated.

And to my wife, Nabeeleh. Thank you for putting my dreams ahead of yours. I believe you worked as hard on this thesis as I did particularly when you had to take care of our children, Mahmoud and Yasmeen. For this, I dedicate this thesis to you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation for Low-Power

Historically, the earliest and the most pressing demands for low-power came from portable devices such as wrist watches and pocket calculators. These products have always put a large emphasis on minimizing power in order to maximize battery life. However, designers of mainstream electronics, such as microprocessors, has focused on boosting circuit speed or processing power to the maximum possible while minimizing area cost. So at the beginning, the VLSI design space had two dimensions of delay and area and designers had attempted to achieve a reasonable balance between these often conflicting goals.

Only over the last decade os so has power become as important as speed and area. Several factors are behind this trend. First, the prolific growth of portable battery-energized electronics with high-throughput such as laptops, Personal Digital Assistants (PDA's), cellular phones, and pagers has placed power as a pivotal design constraint possibly outplacing speed and area in some cases. A major concern here is that the computational requirements of these products is ever increasing, demanding more energy from the small battery. Unfortunately, the battery capacity has only improved by a factor of 2-4 over the last 30 years, which is far behind the 4 orders of magnitude increase in computational power over the same time span [1]. For example, advanced battery technologies such as Nickel-Metal-Hydride offer around 30-35 Watt-hours/pound. Using a battery of a reasonable size with this capacity will allow a maximum of four hours of operation of a state-of-the-art portable multi-media computer before recharging is needed. Using larger batteries increase weight which violates a basic requirement of portability. The only alternative is to design for low-power, which is the focus of this thesis.

Another driving force behind low-power stems from high-end products. Traditionally, power consumption of these devices has been considered as an afterthought with speed and area being the dominant performance metrics. However, as predicted by Moore's law, the number of transistors per chip (chip density) has been increasing by about 1.5 times per year. Recent analysis combining Moore's law and other practical limits projected that 1G transistor chips will be available by the year 2000 and by 2020, 100G transistor chips will be a materiality [2]. Combined with the by now realistic sub 1GHz and larger operating frequencies, the result is a sharp rise in the power dissipation and so operating temperature of these chips. Packaging and cooling of such high-density high-performance and high-power chips are becoming prohibitively expensive. Therefore, low-power has a monetary advantage in this case.

In addition to cooling and packaging expenses, the excessive increase in temperature in high-speed products will at some point cause malfunction of the silicon devices due to various mechanisms. Interconnect fatigue, gate dielectric breakdown and electrical parameter shift are just a few examples of temperature-sensitive failures which, for reliable design, have to be prevented through power (and so temperature) minimization.

Beside the above elements, another motive for low-power is related to the environment. As more micro-electronics products are being used in everyday's life, the demand on energy will sharply increase. Therefore, the lower the power consumption, the lesser the heat generated and so the lower the cost required for extra cooling systems in offices and homes. In this respect, low-power design facilitates competitive cost-to-performance ratio of the electronic equipment.

1.2 Research Objectives

Like many other design problems, low-power digital VLSI design is a "broad engineering" concept requiring optimizations throughout the design process from system level down to the device level. The research presented in this dissertation continues this philosophy and offers a set of novel low-power techniques and design methodologies at the **circuit, gate** and **architectural** levels of abstraction. The fundamental ambitions are:

1) to achieve the lowest possible power dissipation at a better or at least equal speed as conventional approaches while minimizing area cost, and

2) to demonstrate the feasibility of the proposed approaches though application to real-life situations by virtue of chip fabrication or testing on benchmark circuits.

In general, low-power is achieved through either (1) minimizing the voltage swing required to represent logic signals which does not only cut energy per operation significantly but also delay, or (2) selectively reducing the supply and threshold voltages of some circuit components to minimize the total power while meeting a timing constraint. These objectives have been successfully accomplished in the following:

1) **HiCapCS:** a low-swing circuit approach using charge sharing for low-power design of internal bus interconnect line, bit-lines in Read-Only Memories (ROMs), and match-line in Content Addressable Memories (CAMs) [reported in Chapter 3],

2) **CIW/R:** a low-swing circuit scheme using current mode which enables fast and low-power write and read operations of Static Random Access Memories (SRAMs) [reported in Chapter 4],

3) **DVDV:** a new technique at the gate level which utilizes dual supply voltages and dual threshold voltages to minimize the total power dissipated by a logic network while meeting a timing constraint [reported in Chapter 5], and

4) **MLR:** a power macro-modelling approach based on linear regression for fast and accurate architectural level power estimation [reported in Chapter 6].

1.3 Thesis Overview

This thesis divides into seven chapters. This chapter has presented a motivation for and an introduction to low-power design. Chapter 2 gives detailed description of sources of power dissipation in digital CMOS circuits and then review a large array of low-power design techniques at levels from process to system. The concepts presented in Chapter 2 serve as a background and motivate the need for the work presented in later chapters.

The contributions of this research are presented in Chapter 3 through 6. Chapter 3 and 4 offer two distinct schemes for low-power design of a class of digital circuits through voltage swing reduction. The method given in Chapter 3 is most suitable for dynamic digital circuits with high capacitive nodes. A new model (called HiCapCS) to create a low swing on these nodes using charge sharing is presented. The technique is compared to conventional design approaches in terms of energy and speed, and is shown to be superior in all cases. Three applications of HiCapCs are examined: internal bus lines, bit-lines in Read Only Memories (ROMs), and match lines in Contents-Addressable Memories (CAMs). The design and measurement results of three chips utilizing this scheme are described.

Chapter 4 presents a another power saving low-swing scheme based on current injection. The approach is applied to the write and read operations in multi-port Static Random Access Memories (SRAMs). The design of a 32x64 3-port register file chip that employs the proposed technique is described. Both simulation and testing results are given to confirm the speed and power advantages over traditional approaches.

While the above pieces of research focus on low-power circuit techniques, the contributions in Chapter 5 and 6 target low-power design methodologies for higher stages of the design process; namely gate and architecture. Pushing low-power concerns to these levels allow: (1) possibly larger power savings since it is generally believed that decisions made at higher levels of abstraction could have the largest impact on power [3], and (2) faster time-to-market because low-power Computer-Aided Design (CAD) optimization tools have to deal with less number of objects (cells or blocks versus transistors). The challenge here is to have a design methodology that can integrate the low-power axis into the conventional design space with two dimensions of speed and area.

With the above in mind, Chapter 5 presents a new design method for low-power digital logic design. The technique, called DVDV, utilizes a library of gates having Dual supply voltages (V_{dd}), and Dual threshold voltages (V_{th}), hence the name, to achieve high-performance while minimizing both dynamic and leakage power. A heuristic using Depth-First-Search (DFS) to implement DVDV is described. The algorithm is integrated in the Berkeley's SIS-2 logic design environment. Applying DVDV to a set of benchmark circuits shows significant power savings over the dual V_{dd} (with a single V_{th}) scheme, and faster speeds than those possible with the dual V_{th} (and a single V_{dd}) approach.

In Chapter 6, an efficient approach for characterizing power dissipation in data-path macros is given. The goal of this work is to produce a set of power macro-models that can be used at the architectural-level to give a quick, yet accurate, estimate of power dissipations for different design alternatives. The proposed method is based on vector simulation rather than on input statistics, and so is not biased towards certain input behavior. Furthermore, the characterization process requires little knowledge about the internal structure of the modules, allows the user to trade-off between accuracy and characterization time, and propagates effective capacitance directly from transistor-level (real) implementations. Exercising the approach over a set of data-path components show much better accuracy than an existing scheme, with similar estimation times.

Finally, Chapter 7 presents conclusions and topics for future work.

Chapter 2

Low-Power Digital CMOS VLSI Design

The Complementary Metal Oxide Semiconductor (CMOS) device technology is the process of choice for high integration and performance at low-power. It is believed that CMOS will stay the mainstream process for many more years to come. In recent years, aggressive scaling of CMOS to the sub-micron levels has set the pace toward Ultra Large Scale Integration (ULSI) era where it is now possible to integrate a complete system on a single chip (or SOC).

In addition to its high-performance and high integration features, CMOS is an excellent candidate for ultra low-power and low-cost applications. Whereas 20 years ago it took $10M and 10kW to produce a 10 MIPS computer, current game machines with embedded microprocessors make the same performance attainable at $10 only with less than 1W of peak power consumption [4]. In 1972, Swanson and Meindl [5] derived that the minimum supply voltage (V_{dd}) at which a static CMOS circuit can properly operate is given by:

$$(V_{dd})_{min} \geq \rho V_T \qquad (2.1)$$

where $V_T = kT/q$ is the thermal voltage (with k being the Boltzman constant and q is the electron charge), and ρ is factor between 2 to 4. At room temperature (T = 300K), V_T is about 26mV and so the minimum supply voltage can be as low as 0.1V.

This chapter serves as background for the research work presented in this dissertation. We will first give detailed description of the sources of power dissipation in digital CMOS circuits followed by a review of some low-power techniques at all levels of the design process.

2.1 Power Consumption in CMOS Circuits

To reduce the power dissipation of a CMOS circuit, the various sources must be identified. There are two types of power consumption relevant to circuit design: the peak power and the average power. The peak power is related to the maximum instantaneous current drawn from the supply which can result in large voltage drops/bounces on the resistive power/ground rails. This can adversely affect circuit reliability and causes overheating of the devices which degrade the circuit performance. It is therefore essential to have the peak power under control. On the other hand, average power dissipation dictates the battery size and weight needed to operate the circuit for a given amount of time. Reducing the average power is thus more critical than the peak power for low-power applications. Fortunately, approaches for average power optimization also reduces peak power as noticed in [6].

In a typical digital CMOS circuit, there are two main classes of power dissipation: dynamic power (P_{dyn}) and static power (P_{static}). The term dynamic arises from the transient switching behavior of the CMOS circuit. Dynamic power can be further divided into two main components: switching power (P_{switch}) and short-circuit (P_{sc}) power. Similarly, static power has two main elements: DC power (P_{DC}) and leakage power ($P_{leakage}$).

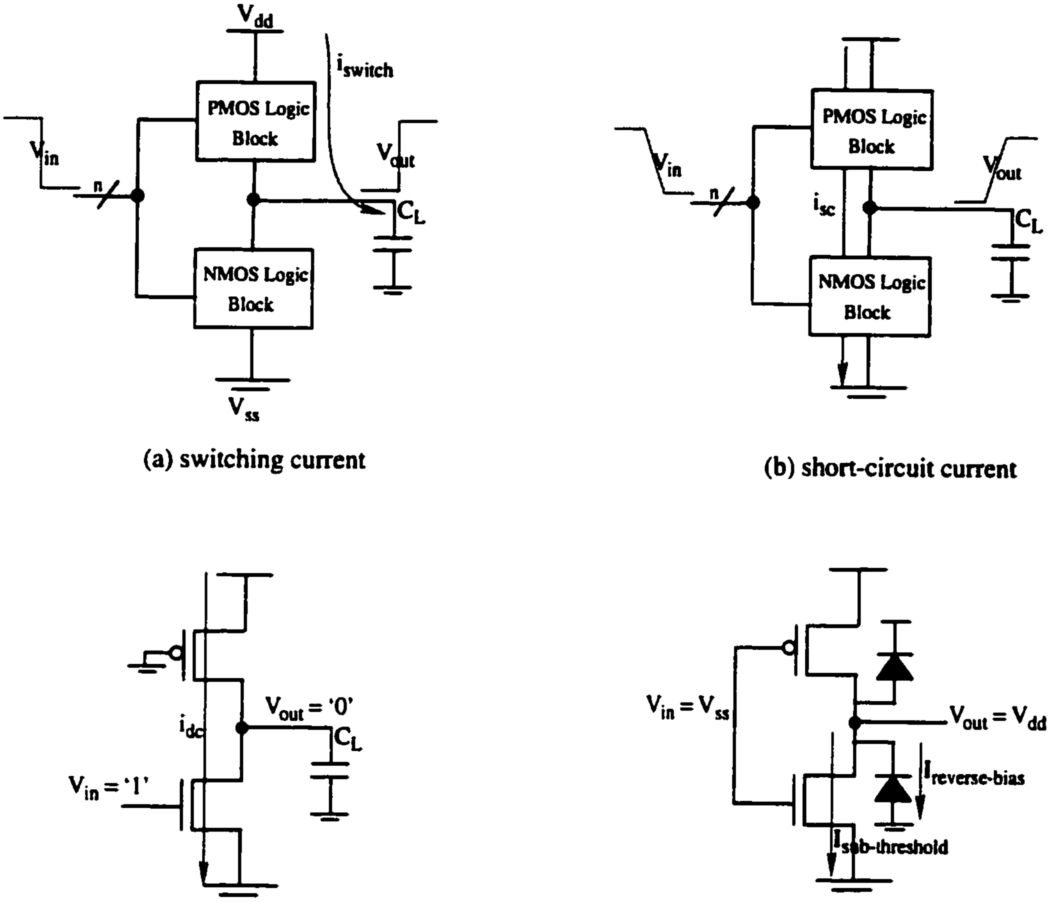
2.1.1 Dynamic Power

2.1.1.1 Switching Power

Switching power is dissipated during logic transitions as a result of charging parasitic capacitances composed of gate, diffusion and interconnect. The situation is modeled in Fig 2.1a where the parasitic capacitances are lumped at the output in the capacitor C_L. During a low-to-high (or 0-to-1) transition at the output, a certain amount of energy is drawn from the supply. Part of this energy is used to charge capacitor C_L, while the rest is dissipated as heat in the PMOS network. Assuming for now that the input waveform has zero rise and fall times which makes it impossible for the NMOS and PMOS devices to be simultaneously "on". The values of the total energy drawn from the supply (E_{supply}) and the energy stored in the capacitor (E_{cap}) can be found by integrating the instantaneous power over the period of interest as follows:

$$E_{supply} = \int_0^{\infty} i_{switch}(t)\, V_{dd} dt = V_{dd}\int_0^{\infty} C_L \frac{dV_{out}}{dt} dt = C_L V_{dd} \int_0^{V_{swing}} dV_{out} = C_L V_{dd} V_{swing} \quad (2.2)$$

$$E_{cap} = \int_0^{\infty} i_{switch}(t)\, V_{out} dt = \int_0^{\infty} C_L \frac{dV_{out}}{dt} V_{out} dt = C_L \int_0^{V_{swing}} V_{out} dV_{out} = \frac{C_L V_{swing}^2}{2} \quad (2.3)$$



(a) switching current

(b) short-circuit current

(c) DC current

(d) reverse-bias drain and sub-threshold leakage currents.

Figure 2.1: CMOS circuit models for power calculation

Since the swing on the output node is a full swing (that is $V_{swing} = V_{dd}$) then $E_{cap} = \frac{C_L V_{dd}^2}{2} = \frac{E_{supply}}{2}$. In other words, only half of the energy supplied by the power source is stored on C_L while the other half has been wasted by the PMOS network. During a high-to-low output transition, the capacitor is discharged and the stored energy is dissipated in the NMOS devices.

EQ 2.2 gives the total energy consumed by a CMOS gate per 0-to-1 switching event. Notice that in a logic network, not all gates switch all the time because they have different topologies and so different responses to input transitions. A useful measure that is usually employed is the switching activity of gate *i*, or α_i, which is average number of times the output node of gate *i* will make a 0-to-1 transition in a given clock cycle. If we assume that the inputs of gate *i* can change at periodicity (clock rate) of *f* times per second, then (over a long period of time) the average switching power consumption of gate *i* is given by:

$$P_{switch} = \alpha_i C_i V_{dd}^2 f \tag{2.4}$$

In general, α_i is less than one. However, if timing information is considered then unwanted spurious transitions (glitches) that occur due to signal races can cause α_i to be greater than one. In other words, the node can make several false 0-to-1 transitions due to signal skews before settling to its steady-state value. If a zero-delay model is used (where a gate delay is assumed to be zero), then α_i is strictly less than one and represents the probability that the output node will make a 0-to-1 transition in a given cycle.

To clarify the concept of the switching activity (α) under a zero delay-model, consider a 2-input static CMOS NOR gate and assume that all possible input combinations (00,01,10,11) are equally likely and independent. If P_l donates the probability of the output node being at logic "1", then from Table 2.1, $P_l = 1/4$ and:

$$\alpha = P_{0\rightarrow 1} = P_0 P_1 = (1 - P_1) P_1 = \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16} \tag{2.5}$$

Similar probabilities can be also derived for other CMOS gates. In the general case, a logic network has several levels of cascaded logic gates and so P_l of a gate becomes a function of the gate's input probabilities. For instance, for the 2-input NOR gate above if

Table 2.1: Truth table for a 2-input static CMOS gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

we let P_A and P_B represent the probabilities that inputs A and B equal "1", respectively then $P_1 = (1\text{-}P_A)(1\text{-}P_B)$ and:

$$P_{0\rightarrow 1} = (1-P_1)P_1 = [1-(1-P_A)(1-P_B)]\,[(1-P_A)(1-P_B)] \qquad (2.6)$$

Again similar expressions can be derived for other types of gates.

The above analysis suggests that the total average power dissipation of a logic network with N gates can be easily calculated by propagating signal probabilities from the primary inputs and then finding α_i (for $i = 1,2,\ldots,N$) at each gate i using a similar expression as in EQ 2.6. However, the problem is more involved if transitions due to glitches should be accounted for. Moreover, signal correlations at the primary inputs and those internally generated due to reconvergant fanouts can not be captured using simple probabilities. More discussion on this issue will be given in Chapter 6.

2.1.1.2 Short-Circuit Power

In CMOS circuits, short-circuit power is a result of the transient current that flows between V_{dd} and ground when both the NMOS and PMOS devices are turned on during logic transitions, as illustrated in Fig. 2.1b. The finite rise and fall times of the input signals result in this direct path which exists for a short time during switching. If I_{SC} is the mean short circuit current drawn by a CMOS gate over a finite period of time T, then the average short-circuit power (P_{sc}) is given by:

$$P_{sc} = I_{SC} \times V_{dd} \tag{2.7}$$

In [7], H. Veendrick found that I_{SC} is affected by the input/output rise/fall times. In addition, he showed that for equal input and output edge times, the short circuit power is minimized and becomes less than 15% of the total dynamic power.

2.1.2 Static Power

Unlike dynamic power, static power is consumed during steady-state where there is no input/output transitions. Static power has two sources: DC power and leakage power. The first component is an inherent property of some CMOS circuit styles, while the second is an outcome of the fact that a MOS transistor is not a perfect switch and so leaks some current.

2.1.2.1 DC Power

A conventional CMOS logic gate (Fig 2.1a) dissipates no DC power, and only consume power during logic transitions as explained in the previous section. However, other CMOS circuit families like pseudo-NMOS wastes DC current when the output is at logic '0' as illustrated in Fig 2.1c for a pseudo-NMOS inverter. Note that in this case, logic '0' is not equal to V_{ss} and depends on the current ratio between the PMOS and NMOS devices. For good high noise margin (low V_{OL}), the PMOS pull-up is made much weaker than the NMOS pull-down, resulting in rise time speed degradation. This demerit in addition to the DC current dissipation limit the of applications that can benefit from this logic style.

2.1.2.2 Leakage Power

The second source of static power which is common to all CMOS based circuits is the leakage power. This power has two main components as illustrated in Fig. 2.1d. The first element is caused by the reverse-bias current that flows in the parasitic diodes between the diffusions (junctions) area and the bulk of a MOS transistor. This current has a typical value of 1 fA/junction which results in a minute contribution to the total power dissipation in a digital circuit [8].

The second more important source of leakage power is caused by the sub-threshold current which is produced by the weak inversion layer that exists in a MOS channel when the gate-to-source voltage (V_{gs}) is below the threshold voltage (V_{th}) of the device. This current increases *exponentially* as V_{th} decreases or the junction temperature increases. In low-voltage circuits, V_{th} is scaled down to offset the delay increase since the delay of a static CMOS gate is propotional to $\frac{C_L V_{dd}}{\kappa(V_{dd}-V_{th})^2}$ where κ is a factor that depends on the process and the sizing of the gate[1].

Under the low-V_{th} scenario, the sub-threshold leakage power component can become significant. This is particularly true in circuits having small dynamic power dissipation due to either operating at low frequencies or spending most of their times in stand-by (idle) mode. The exact expression of the sub-threshold current as well as design techniques focusing on its reduction will be given in Chapter 5.

[1] More accurate and detailed formulation of the CMOS gate delay will be given in Chapter 5.

2.1.3 Power Metric

In the above discussion, it was emphasized that for a properly designed CMOS digital circuit, the switching power is the main source of power dissipation. However, for emerging low V_{dd} low V_{th} circuits, this is not true any more and sub-threshold leakage power is becoming more significant. Thus, the total power dissipation is approximated by:

$$P_{total} \approx P_{dyn} + P_{leakage} \approx P_{switch} + P_{sub-threshold} \qquad (2.8)$$

Note that the term low-power (LP) design is different from low-voltage (LV) and low-energy (LE). Low-voltage means low-power since reducing the supply voltage entails quadratic reduction in the switching power. However, the term low-power spans wider area of requirements from system down to device levels as will be shown in the next section.

On the other hand, low-energy implies low-power but the opposite is not necessarily true. For example, switching power can be reduced by decreasing the frequency of operation (*f* in EQ 2.4). However the energy (CV_{dd}^2) will not be reduced, because the same amount of energy is consumed as before but a over a longer period of time. This will not be useful for battery operated devices where low-energy is more meaningful than low-power. Thus what is desirable is to ***reduce energy with a better or at least constant delay***, which is the measure adopted in this thesis. Note that in the case of constant delay, low-power means low-energy and vice-versa.

2.2 Low Power Design Approaches

Historically, low-power micro-electronics started with the inventions of the transistor in the 1940's and the integrated circuits in the late 1950s [2]. These breakthroughs have replaced the power starving and heat generating vacuum tubes into transistors integrated in one or more chips which resulted in an unparallel power savings from watts range to the milli-watts range. Another major factor that contributed to significant power reduction is the emergence of CMOS technology with the advantages of low-power, low-voltage, low-cost and high-integration over bipolar process.

An important aspect of low-power design is that it is a system characteristic, requiring optimizations at all levels in the design hierarchy, as illustrated in Fig 2.2. In addition, an accompanying computer-aided design (CAD) methodology is needed to

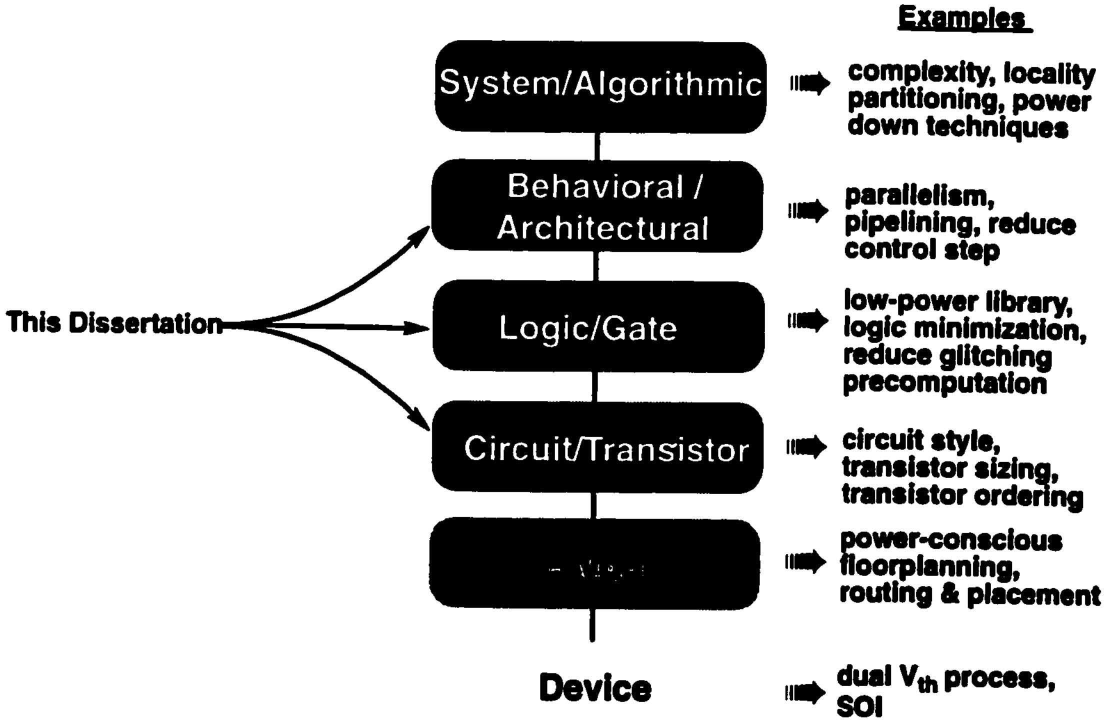

Examples
System/Algorithmic
complexity, locality partitioning, power down techniques
Behavioral / Architectural
parallelism, pipelining, reduce control step
This Dissertation
Logic/Gate
low-power library, logic minimization, reduce glitching precomputation
Circuit/Transistor
circuit style, transistor sizing, transistor ordering
power-conscious floorplanning, routing & placement
Device
dual V_th process, SOI

Figure 2.2: Low-power design space.

help the designer in assessing the effects of different power optimizations on other system's aspects like area and performance. In this thesis, we present a set of novel low-power circuit techniques and design methodologies at the circuit, gate, and architectural levels as shown in Fig. 2.2.

In the following, a brief overview of the state-or-the-art techniques targeting low-power at all levels of the design space will be given. This list is not meant to be complete, but rather to provide the reader/designer with an idea of what possible low-power approaches are available at each level. Each presented scheme will eventually converge to one or more of the primary low-power concepts; namely reducing the supply voltage (V_{dd}), reducing the voltage swing (V_{swing}), reducing the physical capacitance (C_L) and reducing the switching activity (α), assuming that the delay ($1/f$) is not allowed to increase. To evaluate the effect of different optimizations on power dissipation, a means for power estimation is needed. A review of power estimation techniques at different design stages will be presented later in Chapter 6.

2.2.1 Device Level

Scaled CMOS process is generally viewed as the technology most suitable for low-power, battery-operated products demanding high-speed operation. However, scaling the supply voltage without a corresponding decrease in the threshold voltage causes speed degradation. If the subthreshold voltage is scaled aggressively, then the leakage current increases drastically, increasing power as a result.

To address this problem at the device level, processes with dual (or multi) V_{th} were developed [9-10]. The idea is to use low V_{th} transistors in the active mode to maintain good speeds under low V_{dd}, while utilizing high V_{th} devices for ultra low leakage power in the stand-by mode. Numerous techniques applying this concept (or variations of it) will be described later in Chapter 5.

Another improvement in power and/or performance at the device level can be achieved by implementing scaled CMOS on Silicon-On-Insulator (SOI) as shown in Fig. 2.3. The dielectric isolation in SOI offers several advantages over bulk CMOS such as latch-up prevention, high packing density, lower parasitic capacitance and absence of body effect. The last two factors are the main reasons for the 20 to 50% performance increase in CMOS circuits implemented in SOI compared to those using bulk CMOS for the same lithography. This speed advantage can be traded of for lower supply voltage to reduce power. Furthermore, because the body of an SOI device is float, its source-to-body voltage (V_{sb}) is always less than or equal to zero. This will give rise to lower V_{th} (because of negative body bias) and so permits further reduction in the supply voltage.

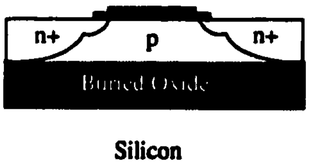


Figure 2.3: NMOS on SOI

The floating body is also a main disadvantage of SOI since it causes problems such as history-dependence (where the delay of the gate is a function of the switching history of the device), false evaluation in dynamic logic, and increased leakage current due to V_{th} lowering [11]. In addition, the low thermal conductivity of the oxide underneath the thin silicon layer results in an increase in device temperature (self-heating) with subsequent reduction in current and possible impact on reliability. Finally, demonstration of mainstream complex application implemented in SOI has yet to be shown before accepting the technology as an alternative to bulk CMOS.

2.2.2 Layout (Physical) Level

At this stage, the main attempt is to reduce the load and/or the switching activity through power-conscious layout schemes. Some of these techniques are reviewed next:

- *floorplanning*: traditional floorplanning techniques attempt to minimize the area of the floor plan. These techniques can be modified to reduce power consumption by minimizing the lengths (and so the physical capacitance) of nets with high switching activities. Gains up to 20% have been reported [12].

- *placement*: placement is the process of assigning locations to gates in a gate network with the objective to reduce the area and/or the critical-path delay. As with floorplanning, conventional placement techniques can be tuned to account for switching activities of nets. With this modification, an average power reduction of 8% has been obtained [13].

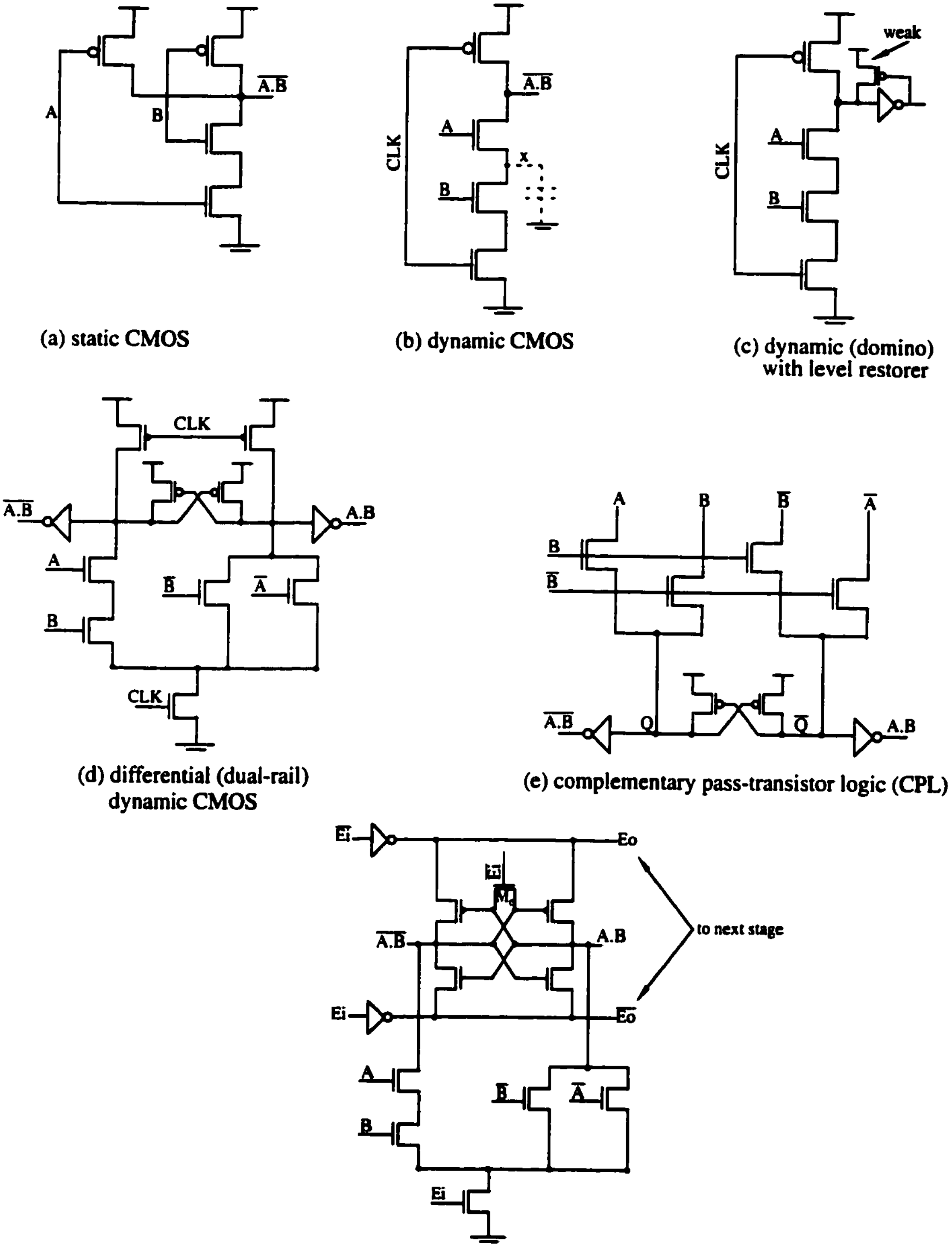
- *routing*: similar to placement, the goal here is to minimize the interconnect lengths so as to minimize the area and/or delay. Original routing algorithms can be modified by simple net weighting where the net weights are derived from the switching activity values of the driver gates. During the routing process, the above weighting scheme is utilized to give higher priority to nets with high switching activities. Experimental results using the above approach have shown only a slight reduction in power dissipation [14].

- *wire and driver sizing*: this process aims to reduce the delay of time critical nets by increasing the wire and/or driver sizes. Increasing the wire size increase the capacitive load on driver and hence the power dissipation. A simultaneous wire and driver sizing approach can minimize the interconnect delay without much increase in power consumption. Gains up to 10% have been reported [15].
- *clock tree generation*: power dissipation of the of the clock net contributes a very large portion to the total power dissipation since the clock is usually the fastest and most heavily loaded net in a digital system. Therefore, for low-power dissipation, it is important to reduce the capacitive load seen by the clock. Original clock routing algorithms which aims to achieve zero-skew through wire sizing can be modified to reduce clock power dissipation. This was done in [16] where instead of increasing wire sizes to reduce the skew and hence increase the capacitive load on the clock, a balanced buffer insertion scheme to partition a large clock tree into a few sub-trees with minimum wire sizes was used. Savings in power up to 60% were reported.

2.2.3 Circuit Level

At the circuit-level, considerable potential in power savings exits by means of proper choice of the circuit style. In general, CMOS circuits can be divided into *static* and *dynamic*, with several alternatives available under each category. In some applications, it is quite possible to combine more than one style to optimize for delay, power and/or area.

The conventional static and dynamic CMOS implementations of a 2-input NAND gate is shown in Fig 2.4a and 2.4b, respectively. The static NAND gate operates in a similar fashion as shown in Fig 2.1a. The operation of the dynamic gate proceeds in two phases: charge (with *CLK* low), and evaluate (with *CLK* high). During the charge phase, there is no path to ground and the inputs can change values. In the evaluate



(a) static CMOS

(b) dynamic CMOS

(c) dynamic (domino) with level restorer

(d) differential (dual-rail) dynamic CMOS

(e) complementary pass-transistor logic (CPL)

(f) asynchronous sense differential logic (ASDL)

Figure 2.4: Various static and dynamic CMOS circuit styles.

phase, the output node will either keep the charge if none of the pull down transistors are turned on or otherwise it will get discharged.

As compared to static CMOS, dynamic style requires substantially lower number of transistors: N+2 versus 2N, where N is the number of inputs. This helps to reduce both area as well as input capacitance, making dynamic logic a better candidate for designing complex and fast circuits than static CMOS.

From power viewpoint, the reduction in parasitic capacitance and the elimination of short-circuit and glitching power is a clear advantage over static CMOS. On the other hand, the switching activity (α) of a dynamic gate is higher than its static counterpart. The is because a dynamic gate consumes power every time its output equals to "0", independent from the previous values. In other words $\alpha_{dynamic} = P_0$ which is always greater than $\alpha_{static} = P_0P_1$, as derived in the previous section. Furthermore, the clock net, which drives two transistors per gate and has a full switching rate, represents a major source of dynamic power dissipation not present in static CMOS circuits.

Other issues of concern in dynamic logic design arise whenever the output evaluates to "1". In this case, the output node is floating and so the charge can be shared with intermediate nodes (such as node x in Fig 2.4b). Also, the charge can disappear if left for a long period of time due to leakage current. These situations can be remedied (at a slight increase in switching delay) by using a static inverter with a weak PMOS feedback that acts as a level restorer as shown in Fig 2.4c. This configuration is also called domino CMOS.

In a recent study [17], it was found that the increased leakage current at low V_{th} can result in a false discharge of the dynamic output node even if a level restorer is used. This sets a lower limit on the minimum V_{th} that can be used for proper dynamic CMOS operation. Based on functionality, static CMOS circuits can significantly tolerate more leakage and so allows lower V_{th} than dynamic logic.

Another conventional dynamic style is shown in Fig. 2.4d. This configuration belongs to the differential class of circuits in which both the output signal and its inverted output are simultaneously available. This is a clear advantage over the single-ended operation since it eliminates the need (and delay) of an extra inverter in case both the signal and its complement are needed. Another feature of the differential operation is the increased noise immunity since common mode noise signals are rejected to a large extent.

The circuit in Fig2.4d operates in a similar manner as the one in Fig 2.3c. One clear distinction, however, is that the cross-coupled level-restorer does not fight the discharge current needed to pull down one of the outputs to low. This will improve the speed as compared to Fig 2.3c. On the other hand, the high-functionality (and so complexity) of this logic style makes it most suitable for implementing large logic functions such as XOR and full-adders.

Over the last few years, various static and dynamic logic techniques targeting low-power and high-performance were proposed. Parts (e) and (f) of Fig 2.4 show two key approaches for the static and dynamic cases, respectively. The scheme in Fig 2.4e is referred to as Complementary Pass-Transistor (CPL) logic [18]. It consists of an NMOS pass transistor logic network driven by two sets of complementary inputs and two inverters used as buffers. The cross-coupled weak PMOS transistors are used to restore the low swing passed by the NMOS transistors. The differential nature of this style allow compact (low-power) and fast realization of complex logic function, as stated previously.

The main drawback of CPL is the degradation in both power and performance when V_{dd} is scaled. This happens because the NMOS transistors pass reduced voltage swing equal to V_{dd} - V_{th}, where V_{th} is subject to the body effect. As V_{dd} is scaled, it will become harder to pull up the intermediate nodes (Q or $\overline{Q}$) causing more delay and short-circuit power in the driving inverters. Both these problems can be greatly reduced by using low V_{th} NMOS pass transistors. However, this comes at the expense of increased leakage current. Finally, similar to all differential approaches, the increased interconnect wiring and so capacitance make CPL inappropriate for cell based systems or gate array design styles [19].

Very recently, a new dual-rail dynamic logic style called Asynchronous Sense Differential Logic (ASDL) was proposed [20]. An example AND/NAND circuit using this style is shown in Fig 2.4f. Unlike the conventional differential dynamic circuit (Fig 2.4d), where one of the outputs has to be charged to V_{dd} every clock cycle, ASDL saves energy by *recycling* the charge on the high node to the other one. This is performed during the equalize phase ($\overline{Ein}$ = "1") by turing on transistor M_e to bring the output voltages to an intermediate level. The gate goes to the evaluate phase when $\overline{Ein}$ becomes low. In this phase, this sense circuit, consisting of the back to back inverters, is turned on. Then depending on the applied inputs, a small voltage difference is created on the output nodes. This difference will be quickly amplified by the sense circuit to CMOS levels. ASDL gates can be easily cascaded by propagating the enable signals from one stage to another as shown in Fig 2.4f.

The ASDL scheme suffers from increased complexity in the output stage which prevents its use in simple circuits. Furthermore, because the sensitivity of the sense circuit is important for reliable operation, the layout of the differential blocks must be fully symmetrical to minimizes mishmashes in the threshold and gain factors of the NMOS devices.

In summary, circuit designer can pick from a wide variety of static and dynamic circuit styles to minimize delay, area and/or power. Several studies have compared these styles in terms of the above factors plus robustness [19] [21] [22]. In general, the choice of a given technique or a variation of it is application dependent and the designers have to make certain trade-offs to achieve their goals. For example, dynamic CMOS is very popular in the design of high-performance systems, at the expense of higher power and more sensitivity to noise than static CMOS. Differential logic is an efficient method for implementing certain logic circuits such as full-adders and carry propagate/carry save functions. Finally, static CMOS fits all applications and is the style of choice for robust, low-voltage amenable and low-power designs.

Before concluding this section, it is worth mentioning that the above schemes are best suited for implementing digital logic circuits. Other low-power techniques exist for designing circuits with high-capacitive nodes and memory elements. A review of these approaches plus improvement upon them will be given in Chapter 3 and 4. Further notice that none of the above methods have addressed the delay and leakage problems associated with the low V_{dd}, low V_{th} design paradigm. A comprehensive review of circuit schemes targeting these issues is given in Chapter 5.

2.2.4 Gate (Logic) Level

Similar to the above two levels, the following techniques achieve power reduction through minimizing switching activity and/or physical capacitance. Schemes for low-power digital CMOS *logic* design by means of supply voltage and/or threshold voltage scaling will be reviewed in Chapter 5.

- *don't care sets*: these sets are input combinations that will never occur at the inputs lines of a gate. In [23], methods that employ don't-care sets to minimize the probability of a power consuming transition at the output node of the gate were described. An average of 10% reduction in power was reported.

- *common sub-expression elimination*: in logic synthesis, this is a technique used to reduce the area of a multi-level gate netlist. In [24], an extension of the scheme to minimize power dissipation as well was described. The is done by weighing each candidate expression with a power saving factor based on how its extraction will affect the loading on its input lines and the amount of logic sharing. The candidates with the most power savings are then chosen to be eliminated. Results show 12% reduction in power compared to a minimum literal netlist.

- *technology mapping (gate sizing)*: this is the process of binding an optimized logic network to the gates in some target library with the objective of minimizing a combined cost function of delay, area and power. In general gate sizing is a non-linear, nonconvex, constrained, discrete optimization problem [25]. Several approaches have been proposed in the past years to solve this problem using heuristic based greedy approaches, various programming methods, and continuous approaches. Power savings up to 25% were reported.

- *state assignment*: the assignment of binary logic values to the states in a finite-state machine (FSM) has a large influence on the area of its final implementation. Therefore, conventional approaches for state assignment targeted minimum area only. Some researchers, as in [26], extended those techniques to account for power dissipation. The idea is to minimize the (coding) distance between those states with high transition frequencies to one another, while accounting for the capacitive overhead of the combination logic used to control the machine. Power reduction of 10-17% were reported.

- *retiming*: this is a method originally used in pipelined circuits to minimize the delay (that is, to increase the clock frequency) through selective re-positioning of the flip flops. In [27], a retiming method targeting low-power dissipation was proposed. The basis is to identify circuit nodes with high hazard activity and high load capacitance as candidates for adding a flip-flop to filter out unnecessary transitions. Power savings of up to 8% were reported.

- *gated clocks*: in a sequential circuit, not all registers will have their values modified during every clock cycle. If the clock inputs to these registers are disabled whenever they are not updated, then some power can be saved. Applying this technique to some FSM circuits has produced 10-30% savings in power [28].

- *precomputation:* pre-computation is a technique in which some of the outputs of a digital circuit (called predictors) are precomputed one clock cycle before they are required. These predictors can be used in the next clock cycle to reduce internal switching activity by disabling those parts of the circuit that do not contribute to the final computation. In [29], this approach was shown to achieve up to 50% reduction in power in some circuits. However, this number ignores the logic overhead required to implement the predictors.

2.2.5 Architectural and Behavioral Levels

The high-level (architectural and behavioral) is the design entry for a large portion of digital systems and decisions made at this level could have the largest impact on power savings. Researchers have reported orders of magnitude reduction by picking the proper high-level choice. The following is a quick overview of the state-of-the-art low-power architectural and behavioral level design tools/methodologies:

- *parallelism and pipelining*: parallelism can be used to reduce power by decreasing the amount of time needed to do a computation and use the extra time available to scale down the supply voltage. Parallelism has a significant impact on power reduction since if the supply voltage is reduced by half, dynamic power dissipation will be reduced quadratically. The actual savings due to parallelism is less, however, due to the increase in the capacitance. Similar to parallelism, pipelining is another technique that trades extra speed for low-voltage and hence low-power. Parallelism and pipelining were implemented as part of the HYPER-LP [30]; a system for low-power architectural- and behavioral-level transformations targeting DSP designs. It should be noted here that not all applications/circuits can be parallelized/pipelined. Furthermore, this technique can lower the supply voltage up to a point where the added capacitance overhead outweighs the savings in power.

- *low-power high-level transformations*: a number of transformations for power optimization were proposed and implemented as part of the HYPER-LP system [30]. These include: using parallelism/pipelining to lower the supply voltage (see the above item), operation reduction to decrease the amount of switched capacitance, strength reduction (operation substitution) to consume less energy per computation, resource utilization to reduce switched capacitance, path balancing to lower glitching, and word-length reduction to minimize the average interconnect capacitance. An algorithm that implements these transformations was developed. The algorithm uses a combination of heuristic and probabilistic techniques to construct a low-power design. The results indicate that an order of magnitude reduction in power can be obtained by using the above transformations.

- *glitch reduction*: a technique for automatic reduction of glitches at the architectural-level was proposed in [31]. The approach is suited to control-flow intensive designs where glitches generated at control signals have a significant impact on the circuit's power consumption. The technique includes restructuring multiplexor networks to enhance data correlations, eliminate glitchy control signals and reduce glitches on data signals. In addition, clocking control signals and inserting selective rising/falling delays are used. On the average, this scheme saves power by 17%.

- *multiple supply voltages*: the idea here is to use more than one supply voltage such as a low supply voltage is used for operations (modules) which are not on the critical path while keeping the supply voltages of operations on the critical path to the maximum. In [32], a dynamic programming approach was used to solve the multiple supply voltage scheduling problem, and an average energy saving of 53% (with a computation time constraint of 1.5 time the critical path delay) was reported.
- *reducing memory accesses*: in many digital systems, memory consumption accounts for a large portion of the total power dissipation. Therefore, it is important to minimize the number of memory accesses required to perform a given task. In [33], a simulated annealing based technique which allocates variables to memory banks in an embedded DSP was described. The approach tries to maximize simultaneous data transfers form different memory banks to registers in order to minimize access overhead and so reduce energy consumption. Energy savings up to 47% were achieved with this approach. Another approach based on network flow for optimizing the number of internal/external memory data accesses for low-power was described in [34].
- *using multiple clocks*: in [35], a multiple clocking scheme for low power RTL design was presented. The basis is to partition a high-level description of the circuit into n modules fed by n non-overlapping clocks. Each module is then operated only during its corresponding duty cycle at a frequency equals to f/n. However, the overall single frequency of the circuit remains f, the single clock frequency. This schemes reduces power because inactive partitions are turned off during their off duty cycle. A resource allocation algorithm to synthesize clock module partitions at the RTL was proposed. Gains up to 50% were reported.

- *low-power architectural synthesis*: this refers to the process of automatically generating a Register-Transfer Level (RTL) representation of a digital system from its Control/Data Flow Graph (CDFG) description. Three main tasks are performed during architectural synthesis: (1) *allocation* which determines how many instances of each hardware resource is required, (2) *binding* (or assignment) which maps operations in the CDFG to resources, and (3) *scheduling* which decides at which clock cycle each operation is to be executed. Traditional architectural synthesis aims at minimizing the number of resources to perform a particular task and/or to minimize the total execution time. For low-power design, minimizing power dissipation adds an extra dimension (and so more complexity) to the original architectural synthesis problem. Various methods based on integer linear programming (ILP) as in [36], iterative improvement techniques as in [37], and graph-based algorithms as in [38] were proposed to solve this problem. Similar to other low-power techniques, the aim here is to reduce the switching activity, the load capacitance, and/or the supply voltage.

2.2.6 System/Algorithmic Levels

The number of operations needed to execute a given task can make a big difference in the power dissipation between one algorithm and another. The type of operation is also another important factor that should be considered when making algorithmic trade-off. Other power minimization strategies that have been suggested at this level include using Gray-code representation (as opposed to binary) to minimize the switching activities on memory address lines [39], stopping the clock feeding idle blocks in a synchronous digital system [40], and:

- *dynamically varying the threshold voltage:* it is well-known that a key method for low-power design is voltage scaling. To compensate for speed degradation due to reduced voltage, one approach is to use architectural techniques, such as parallelism, which trade off area for lower power dissipation. Another more promising approach which does not increase the silicon area is to reduce the threshold voltage. However, scaling down V_{th} causes an exponential increase in the sub-threshold current. As such a "power-optimum" V_{th} must be chosen to compromise between improving the circuit speed at reduced supply voltage and increasing the sub-threshold current. In a recent study [41], Chandrakasan et. al. found that for "continuously operational" circuits, the above approach of using a single scaled V_{th} to be very promising for power savings. For example, for a ring oscillator circuit operating at 11.8Mz, scaling V_{dd}/V_T from 1.4V/0.45V to 0.55V/0.06V resulted in 3 times reduction in energy consumption at the same performance. The above technique is not energy-efficient, however, for "event-driven computations" where long period of inactivates are followed by short periods of computations such as in an X-server. This type of behavior makes the increase in the sub-threshold current at low V_{th} more than offsets the savings in dynamic power obtained by reducing the supply voltage. To solve this problem, they vary V_{th} dynamically using a special SOI process with active substrate. When the system is active, a low V_{th} is used to maintain the same speed under low V_{dd}, while a high V_{th} is employed in the idle mode to reduce the sub-threshold current. Applying this technique to an X-server processor resulted in 43% power savings in the adder component of the processor, 80% for the shifter, and 97% for the multiplier.

- *data-driven dynamic voltage scaling*: Chandrakasan et. al. noticed that in many DSP algorithms, the computational workload per sample varies significantly depending on the sample data locality and correlation [42]. This observation was used to reduce the power consumption of a digital signal processor by dynamically varying the supply voltage depending on the predicted amount of work required by each data sample. If the workload for a given sample is less than a the peak, then with a fixed voltage paradigm, the processor will perform the work and sets idle for the rest of the sample period. However, if the supply voltage is allowed to change, then one can "stretch" the delay of the processor during this sample, without loss in throughput, by decreasing the supply voltage. If the workload for a given sample is at its peak, on the other hand, then a supply voltage high enough to keep the throughput of the processor should be used. Using this scheme, a maximum factor of five in power reduction is possible.

2.3 Summary

This chapter has provided an introduction to low-power digital CMOS design. This included a detailed description of all power consuming mechanisms in CMOS circuits, as well as an overview of a number of low-power techniques for levels of abstraction ranging from device to system levels.

Over the last decade, a great portion of research efforts has focused on low-power aspects of digital circuits. The amount of literature in this area is immense and could not be covered in this short review. However, the discussion in this chapter was useful enough to make it evident that the dominant component of power dissipation arises from the switching power. This power is needed to charge and discharge parasitic capacitances and is given approximately by $P_{switch} = \alpha C V_{swing} V_{dd} f$, with $V_{swing} = V_{dd}$ in CMOS circuits. For the same speed (f), techniques were presented to decrease P_{switch} by reducing the physical capacitance, C, the switching activity, α, the voltage swing V_{swing}, and the supply voltage, V_{dd}.

It is well-know that reducing V_{dd} is the most effective means for power reduction in CMOS circuits because of the quadratic dependence between P_{switch} and V_{dd}. However, lowering V_{dd} incurs speed degradation, which is largely noticeable as V_{dd}

approaches the threshold voltage, V_{th}. One scheme to overcome this problem is to compensate for performance reduction at: (1) the architectural level through parallelism and pipelining, (2) the logic-level through, for example, gate sizing, (3) the transistor-level through novel circuit techniques that offers large speed-ups as compared to conventional CMOS and so allow deep scaling of the supply volage, and (4) the device level by using low-voltage and high-performance processes such as SOI. Another approach to this problem, which does nor require major changes to current design trends or technologies, is to scale V_{th} as well. However, this comes at a large increase in the leakage (subthreshold) current.

This thesis deals with the issues pertaining to reducing both switching and leakage power consumption through novel techniques at the circuit (Chapter 3 and 4), gate (Chapter 5), and architectural-level (Chapter 6). A Review of the state-of-the-art approaches relating to each proposed scheme will be given in the corresponding chapter.

Chapter 3

Low-Power Design Using Charge Sharing

3.1 Introduction

As described in the previous chapter, the dynamic power dissipation of a CMOS gate (*i*) (ignoring short-circuit power) is given by:

$$P_{dyn} = \alpha_i C_i V_{dd} V_{swing} f \tag{3.1}$$

where α_i is the gate activity, C_i is the total load and parasitic capacitance, V_{swing} is the voltage swing at the gate output, and f is the operating frequency of the system. Since $V_{swing} = V_{dd}$ in CMOS circuits, lowering the power supply voltage reduces the dynamic power quadratically. However, this approach leads to speed degradation for a given technology. If full scaling is applied, then the device threshold voltage (V_{th}) has to be *relatively* reduced to keep the same delay. However, this comes at the expense of increasing the sub-threshold (leakage) current. This issue will be further investigated in Chapter 5.

Another method to significantly decrease P_{dyn} without sacrificing delay is to decrease V_{swing} while maintaining the power supply voltage V_{dd}. This is particularly useful for high values of C_i. In this chapter, we describe a new method based on *charge sharing* (henceforth called HiCapCS) to reduce the swing on heavily loaded dynamic CMOS circuits. The basic idea is to generate an extremely low V_{swing} on the gate output by employing the principle of charge sharing. This swing can be then detected and restored to CMOS level using a single- or a double-ended sense amplifier. Applications of this technique include internal bus interconnect lines, bit-lines in Read-Only Memories (ROMs), and match lines in Contents-Addressable Memories (CAMs).

The rest of this chapter is organized as follows. The basic technique is described in Section 3.2. Applications to internal bus lines, CAMs, and ROMs are discussed in Sections 3.3, 3.4, and 3.5, respectively. The chapter ends with a summary.

3.2 Charge Sharing

In this section, we will describe how charge sharing can be used to produce low V_{swing} on the gate output. The conventional charge sharing model is shown in Fig. 3.1a. This model is similar to the one used in [43]. In this scheme, *LC* is a dummy capacitor that acts as a virtual ground line. Capacitor *HC* represents the large lumped capacitance at the gate output. With *CLK* high, node *HC* is charged to V_{dd} and node *LC* is discharged to zero. When *CLK* becomes low and depending on the data *D*, node *HC* will either retain its charge for logic "1" ($D = 1$ and M_2 is off), or ***share*** its charge with node *LC* for logic "0" ($D = 0$ and M_2 is on). In the latter case, and since *HC* is much larger than *LC*, logic

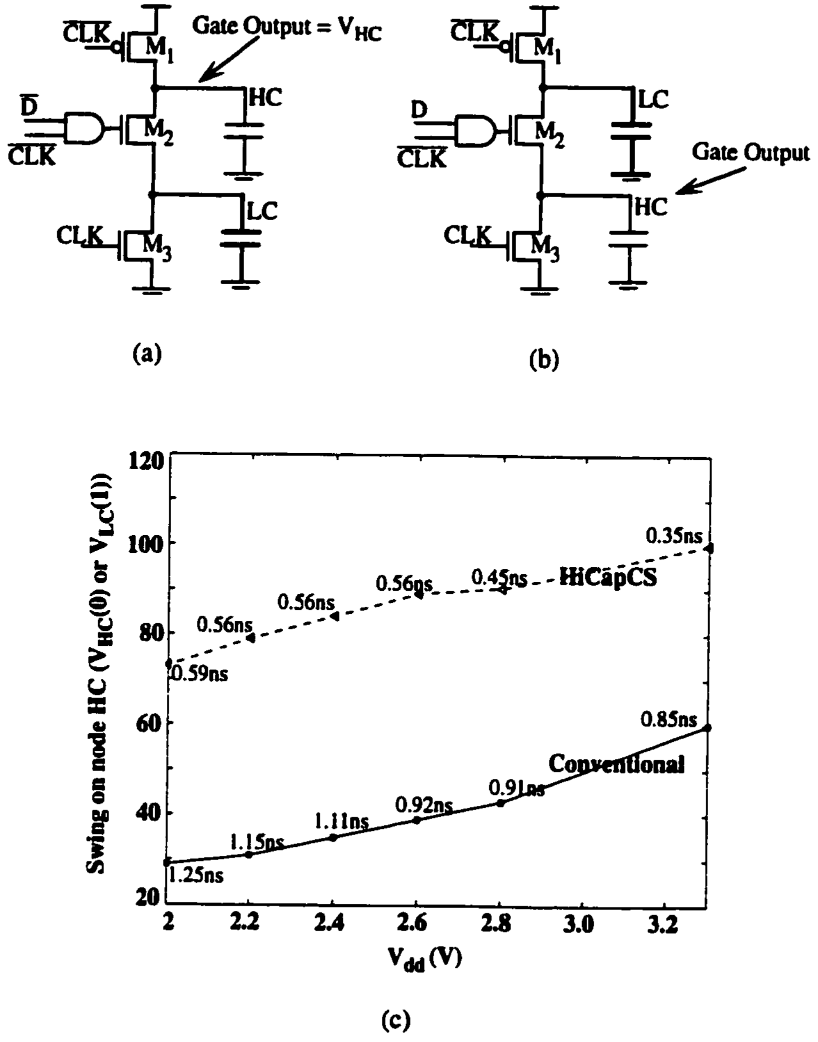


Figure 3.1: (a) Conventional charge sharing, (b) HiCapCS, (c) Swing and delay comparisons.

"0" will be slightly smaller than V_{dd}. Note that the final value on LC equals $[V_{dd} - V_{th}(V_{LC})]$, where V_{th} is subject to the body effect. Thus using charge conservation yields:

$$HC \times V_{dd} = HC \times V_{HC} + LC \times (V_{dd} - V_{th}(V_{LC})) \qquad (3.2)$$

$$V_{th} = V_{T0} + \gamma\left(\sqrt{2\phi_f + V_{LC}} - \sqrt{2\phi_f}\right) \tag{3.3}$$

where V_{T0} is the threshold voltage at zero bias, γ is the body effect coefficient, ϕ_f is the Fermi potential, and V_{LC} is the voltage difference between the source node of transistor M_2 (Fig. 3.1a) and the substrate node (bulk). Solving EQ 3.2 for V_{HC} gives:

$$V_{HC}(0) = V_{dd} - \left(\frac{LC}{HC}\right)(V_{dd} - V_{th}) \tag{3.4}$$

EQ 3.4 gives the value of V_{HC} for logic "0", while $V_{HC}(1) \approx V_{dd}$, as mentioned before. Thus the voltage difference between logic "1" and logic "0" equals $\left(\frac{LC}{HC}\right)(V_{dd} - V_{th})$. Since LC << HC, the above quantity is much smaller than the CMOS counterpart and hence power can be reduced (ignoring low-swing restoration overhead for now).

The problem here is that transistor M_2 will suffer from body effect and so the resultant swing on *HC* (EQ 3.4) may be prohibitively small from practical viewpoint[1]. In addition, when turned on, M_2 operates in the saturation region with a small $[V_{gs}$-$V_{th}(@V_{LC})]$, making it slow. Increasing *LC* will improve both the speed and swing at the expense of power and area. To alleviate the above problems, the HiCapCS scheme shown in Fig. 3.1b is proposed. The main difference is that the positions of *HC* and *LC* are interchanged. As a result, discharging of *LC* (for logic "1" in this case) occurs mostly in the linear region with larger $(V_{gs}$-$V_{th})$ value and much smaller body effect. This will (1) improve discharging speed by reducing the on resistance of M_2 and (2) increase the voltage swing on *HC*. Thus for logic "1", both *LC* and *HC* reach the same value as given by:

[1] For example, it may not be large enough to be detected by a following sense amplifier.

$$V_{HC}(1) = V_{LC} = \left(\frac{LC}{HC+LC}\right)V_{dd} \tag{3.5}$$

Since in HiCapCS logic "0" is equal to ground, the voltage difference between the two logic states is $\left(\frac{LC}{HC+LC}\right)V_{dd}$. This is still much smaller than the full-swing CMOS but larger (better) than the conventional charge charging scheme shown in Fig. 3.1a.

To verify the above argument, the circuits in Fig. 3.1a and 3.1b were simulated using HSPICE with a 0.35 µm CMOS process and the following parameters: $HC = 2$pf, $LC = 50$ff, $W_{MI}= W_{MI} = W_{MI} = 2$ µm. The simulation results are shown in Fig. 3.1c. For each scheme, the figure plots the resultant V_{HC} versus the supply voltage. For each value, the figure also gives the amount of time needed to obtain about 90% of the swing. It is clear that, for the same LC and HC values and irrespective of V_{dd}, HiCapCS produces more swing in a shorter time than the conventional scheme.

Notice that with HiCapCS a low swing is obtained very quickly without any DC power dissipation or the need for pulsed operation or reference voltage generator. Moreover and unlike the conventional model, HiCapCS can be easily employed in several high capacitive CMOS circuits. This work explores three such applications based on the model shown in Fig. 3.1b with some variations. For each case, both simulation and experimental results are given to demonstrate the speed and power advantages over traditional circuits.

3.3 Internal Bus Line Design Using HiCapCS

In a deep sub-micron IC where both integration density and chip dimensions are large, global interconnect wires are responsible for a large portion of the total energy. It is expected that the power dissipation associated with buses and clocking networks in future ULSI chips can be up to 50% of the total on-chip power consumption [43]. This is due to the slower scaling of the interconnect capacitance as compared to the gate capacitance, making the former more dominant as feature sizes continue to shrink. This power can be decreased by reducing the average wire length through novel architectures utilizing local (short) interconnects, or by data encodings to reduce switching [44].

At the circuit level, reducing the swing of signals running on these lines can lead to significant power reduction. In this section, we will demonstrate how HiCapCS can be used to achieve this goal. Comparisons to conventional bus designs are also presented. Finally, implementation and testing details of a prototype 2-bit bus chip will be described.

3.3.1 Circuit Design

Fig. 3.2 shows a circuit that applies HiCapCS to an internal bus line (*i*). Capacitor C_A is a dummy capacitor that can be implemented using polysilicon gate capacitance. Capacitor C_{Bi} ($C_{\overline{B}i}$) represents the interconnect (wiring) capacitance of bus line (*i*). The circuit works in a differential manner as follows. When *CLK* is high, nodes B_i, $\overline{B}_i$, D_i, and $\overline{D_i}$ are discharged to zero and node *A* is charged to V_{dd}. At the same time, the input data (Din_i) is allowed to change. When *CLK* is low, the charge on node *A* will be *shared* with either B_i or $\overline{B}_i$ depending on the bus data (Din_i). The swing produced on any of these lines is given by EQ. 3.5 while the other node will have zero swing. The voltage difference between B_i and $\overline{B}_i$ will be converted into a CMOS full-swing using a PMOS current-latch sense amplifier (PCLSA). The sense amplifier enable ($\overline{SAE}$) signal is a delayed version of *CLK*. A delay is needed to allow for the low-swing to develop and travel on the bus line to the PCLSA receiver.

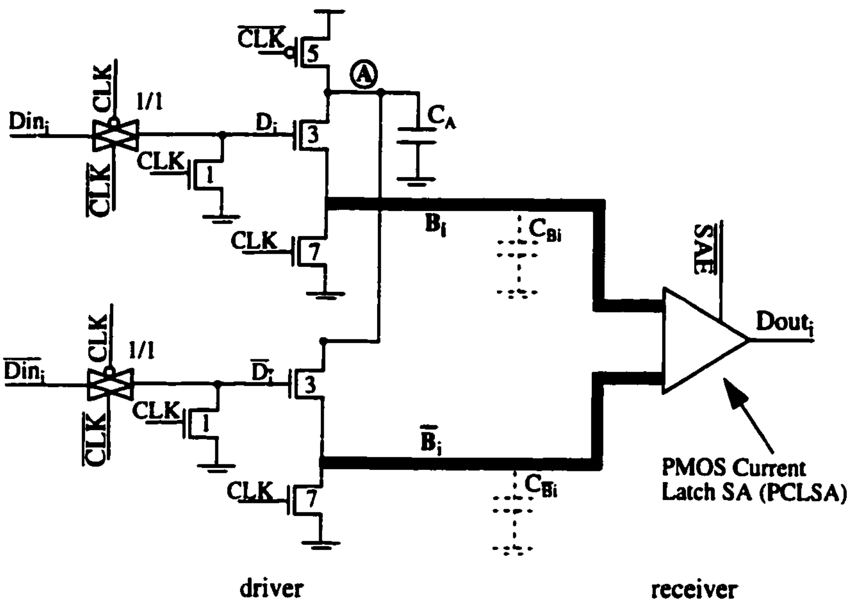


Figure 3.2: Internal bus line design using HiCapCS.

The PCLSA is shown in Fig. 3.3. This amplifier is comparable to the NMOS current latch sense amplifier (NCLSA) [45] except that pmos transistors are used (instead of NMOS) to sense the voltage difference. The operation of the amplifier proceeds as follows. When $\overline{SAE}$ is high, nodes N_1-N_4 are pre-discharged to low. When $\overline{SAE}$ falls, the sense amplifier is activated, and the voltage difference between B_i and $\overline{B}_i$ is

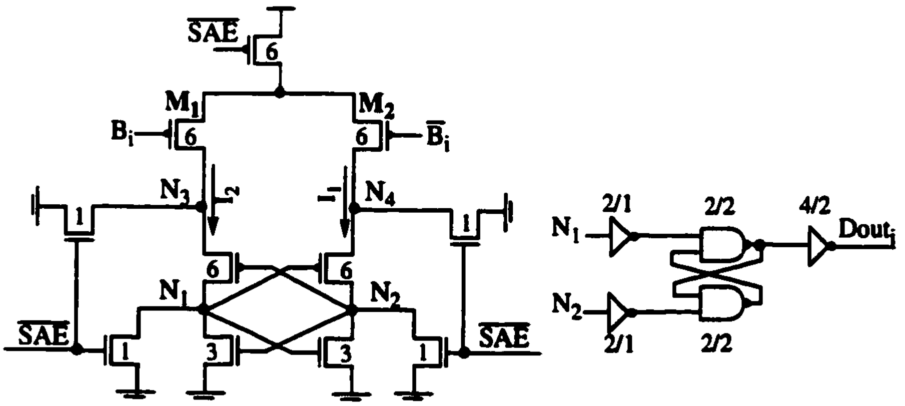
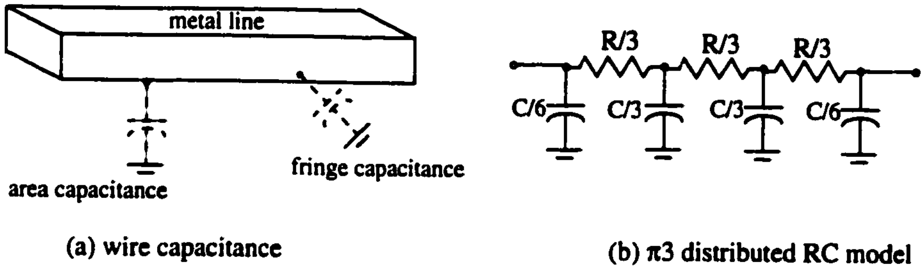


Figure 3.3: PMOS current latch sense amplifier (PCLSA).

converted into a current difference between I_l and I_2. This in turn will create a voltage difference between N_l and N_2, which is amplified into a full-wing CMOS value by the two back-to-back inverters latch. After it evaluates, this latch prevents any DC power dissipation. The cross-coupled NAND latch will hold the output data after $\overline{SAE}$ returns back to high.

The circuit in Fig. 3.2/3.3 was simulated using HSPICE with a 0.35 μm 3-metal process. The transistor sizes used in the simulation are given in the figures. The interconnect line is a metal-1 wire with a length of 13mm. This corresponds to a bus capacitance (C_{Bi}) of about 2pf. Note that C_B represents the sum of (1) both the area and fringe capacitance from the wire to the substrate (see Fig. 3.4a), and (2) parasitic capacitance from all transistors connected to the wire. A value of 70ff was used for the dummy capacitor (C_A). Parallel to C_A is about 25ff parasitic capacitance from associated transistors. Taking the above combination of C_A and C_{Bi}, a low-swing of about 150 mv on the bus line was obtained for V_{dd} = 3.3.



(a) wire capacitance

(b) π3 distributed RC model

Figure 3.4: Interconnect line model

Although the PCLSA is capable of sensing a differential voltage as small as 50mv as shown by simulations, using a larger swing is desirable for several reasons. First, it allows us to operate the circuit at supply voltages lower than 3.3. Second, it compensates for mismatches between (1) the capacitance of the two wires (B_i and $\overline{B}_i$) and, (2) devices M_1 and M_2 (Fig. 3.3) due to process, layout, or temperature variations. Finally, it improves the immunity of the circuit against supply voltage noise.

To accurately model the distributed *RC* effect of the interconnect wire, a π3 model was used as shown in Fig. 3.4b. The *R* and *C* in the figure are the total resistance and capacitance of the wire as estimated from the process parameters. Fig. 3.5 shows the simulated waveforms from HSPICE at frequency f = 100 MHz. Notice that the swing on either B_i and $\overline{B}_i$ is about 150 mv as explained before. Also note that it takes about 0.4ns for the low-swing signal to develop and travel from the driver to the receiver after *CLK* becomes low, and about 0.45ns for the output signal ($Dout_i$) to become available after

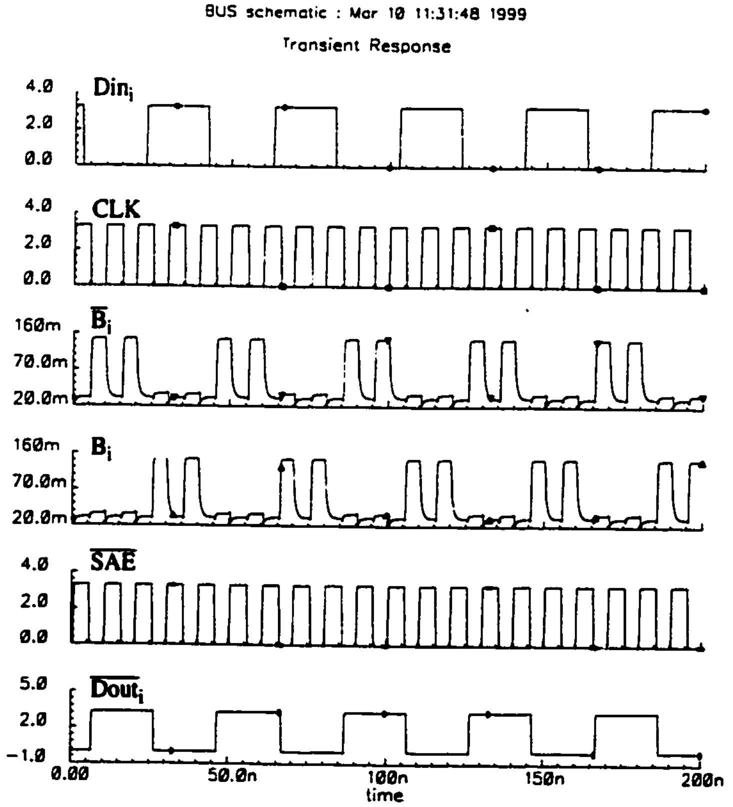


Figure 3.5: HSPICE simulation results for the HiCapCS bus.

enabling the PCLSA. Thus the total delay from *CLK* = low to $Dout_i$ is 0.95ns. The power consumption at f = 100MHz, including that of the driver and receiver, is 252 μW.

3.3.2 Comparisons

In this section, we will compare the proposed HiCapCS bus architecture to: (1) conventional static CMOS configuration, and (2) data-dependent bus [43] which is based on the conventional charge sharing model shown in Fig. 3.1b.

3.3.2.1 Conventional Static CMOS Bus Line

The conventional CMOS bus driver/receiver configuration is shown in Fig. 3.6. The driver has an increased size to reduce driver/receiver transmission delay and short-circuit power. With this configuration, the swing on the interconnection (V_{swing}) equals the power supply, V_{dd}.

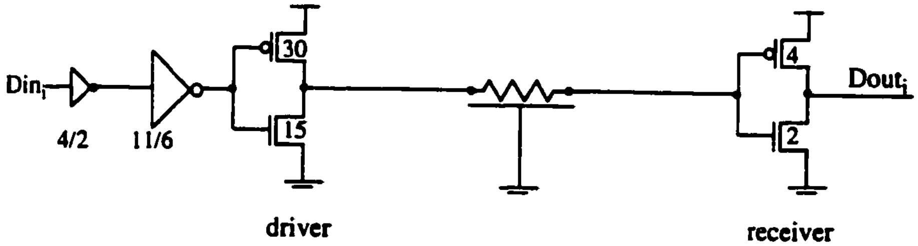


Figure 3.6: Conventional CMOS bus line.

The circuit in Fig. 3.6 was simulated using the same 0.35 μm process. As for the HiCapCS case, the metal interconnect capacitance is about 2pf and the π3 model was used for the interconnect. The simulated delay from Din_i to $Dout_i$ is 0.94ns which is comparable to that of the HiCapCS bus. On the other hand, the total power dissipation is 1230μW which is about 5 times larger than HiCapCS at the same frequency.

It is important to note that the CMOS bus transceiver consumes power only when the input data (*Din$_i$*) changes. This is unlike HiCapCS where: (1) capacitor C_A has to be charged and discharged, and (2) the PCLSA has to be turned on, every clock cycle irrespective of the input data. Thus, for a bus line (*i*) with switching activity α_i, HiCapCS is a power efficient alternative to the conventional CMOS only if $\alpha_i > 0.2$ (1/5). This lower bound is still smaller than the average case ($\alpha_i = 0.5$).

Because it requires full rail-to-rail swing, the maximum transient current of the CMOS design is about 5 times larger than that of the low-swing HiCapCS. This means that HiCapCS produces much less noise on the supply line as compared to CMOS. From area perspective, requiring two lines per bus bit is not a great disadvantage of HiCapCS. This is because the area penalty is determined by the larger pitch amongst the driver and the receiver. Given the large driver size in the CMOS case, the area penalty of HiCapCS can be at most as bad as the CMOS configuration. On the other hand, having a differential operation gives a good immunity to common-mode noise.

3.3.2.2 Data-Dependent Bus Design

A low-swing internal bus design based on the conventional charge sharing model (Fig. 3.1b) was proposed in [43], and the circuit is shown in Fig 3.7. When *CLK* is low, all bus lines are charged to V_{dd}. In the evaluate mode, *CLK* becomes high. Then, depending on the number of input bits that switch from "1" to "0", call it *n*, the voltage on *each* of these lines is reduced by $1/(n+1)$ of the supply voltage. This happens since the charge originally held by the *n* wiring capacitance is *shared* with the initially discharged dummy ground line (DummyGND) which has an equivalent capacitance of one bus line. Since the generated logic swing depends on the transmitted data (such as the larger *n* is, the smaller the swing), the architecture is called Data-Dependent Bus (DDB).

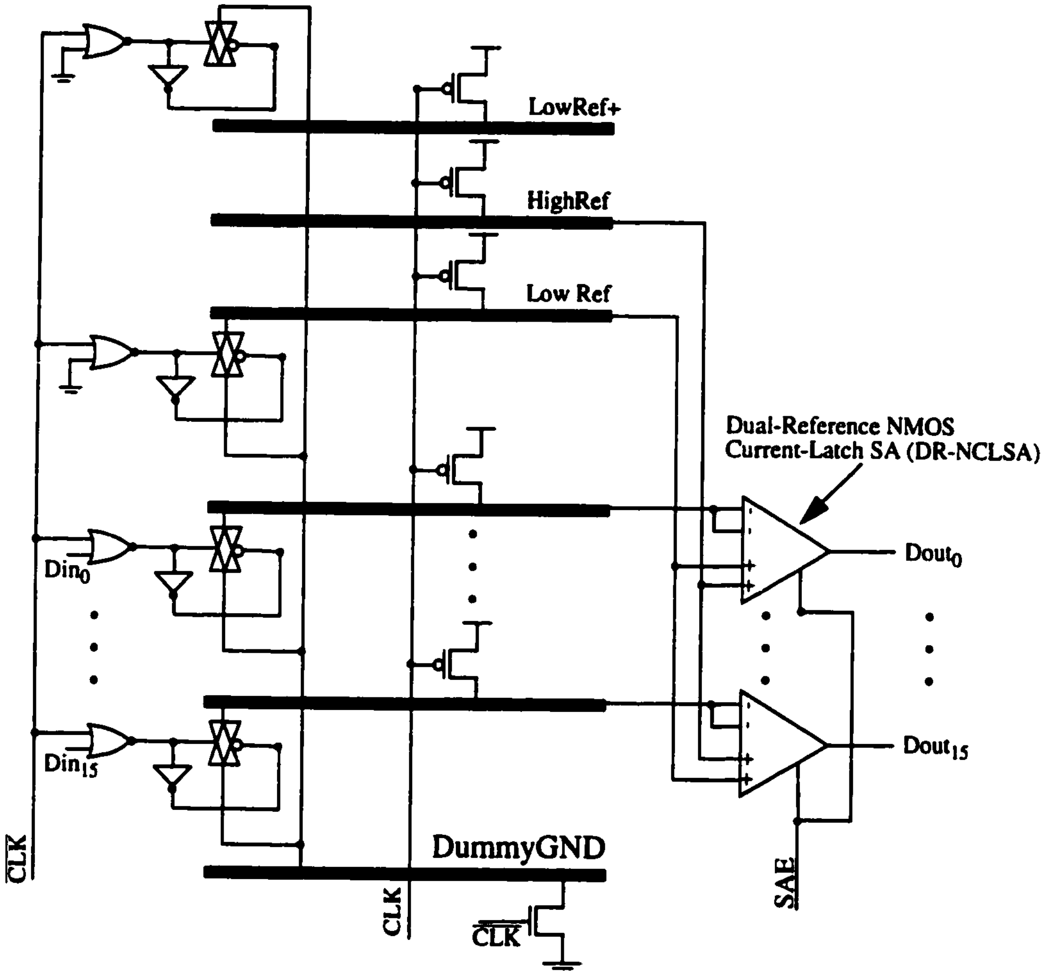


Figure 3.7: Data-dependent bus (DDB) architecture.

As shown in Fig 3.7, a dual-reference NMOS current-latch sense amplifier (similar to DR-PCLSA) is used to restore the transmitted data to CMOS level. Three reference lines are needed in this case. Two of these lines (LowRef and LowRef+) have to charged and discharged every clock cycle. The HighRef line is "sandwiched" between LowRef and LowRef+ to improve the noise immunity of the scheme against crosstalk.

For the design example given in [43] which consists of a 16-bit bus, 2pf per bus line, and V_{dd} = 3.3V, the total swing of the DDB design varies between

$\left[\left(\frac{2pf}{4pf+2pf}\right)3.3\right] \times (0+2) = 1.1V$ for $n = 0$ (only LowRef and LowRef+ will switch), and $\left[\left(\frac{2pf}{32pf+4pf+2pf}\right)3.3\right] \times (16+2) = 3.1V$ for $n = 16$. If on average, half of the bits will swing, then the total swing will be 3.0V. In the HiCapCS scheme using a 150mV per line and assuming a 16-bit bus, 2pf per bus line, and $V_{dd} = 3.3$V, the total swing is 150mVx16 = 2.4V, which is 25% smaller than the DDB architecture.

From speed perspective, the amount of time needed to produce the low-swing in the DDL case in the worst case (all bits switching) is more than that in the HiCapCS case because of the reasons explained earlier in Section 3.2

3.3.3 Implementation Details

A 2-bit bus circuit was laid out in a 0.35 µm 3-metal process in order to perform experimental evaluation of the circuit. Each bus line is about 13mm long and was constructed using metal-1. Because the bus length is much longer than the horizontal or vertical dimension of the chip, a bus line was laid out in a "zigzag" fashion as shown in Fig. 3.8. The dummy capacitor C_A was constructed using polysilicon-to-NWELL capacitance. In designing the bus lines, special care was given to match the capacitance of the differential bus lines B and $\overline{B}$. These lines were separated as much as possible in the layout (as

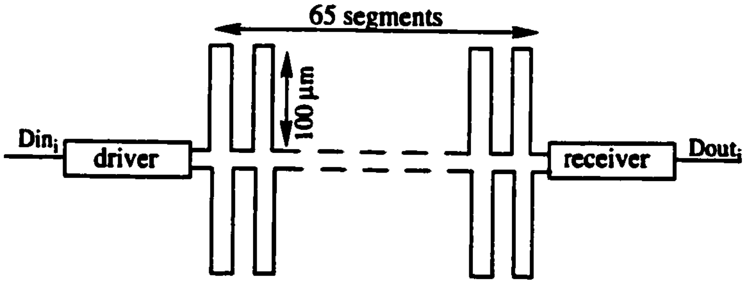


Figure 3.8: Bus layout strategy.

determined by the layout of the transceivers) in order to minimize any coupling (cross talk) between them. The $\overline{SAE}$ signal was generated from *CLK* by using a chain of CMOS inverters in order to obtain the desired delay. The *CLK* net was not allowed to cross over the bus lines to further prevent coupling.

3.3.4 Measurement Results

The photo-micrograph of the 2-bit bus circuit is shown in Fig. 3.9. To verify the chip functionality, random data was used at the bus inputs and then compared to that at the outputs. An IMS-100 tester was used for pattern generation. The *CLK* signal from the input pad is passed to the circuit and also to an output pad as illustrated in Fig 3.10. This allows us to measure the real delay from *CLK* = 0 to valid *Dout* by subtracting the delay of the I/O pads.

An example input and the resultant output waveforms using 1.5V supply voltage are shown in Fig. 3.11. The measured delay versus the supply voltage is given in Fig

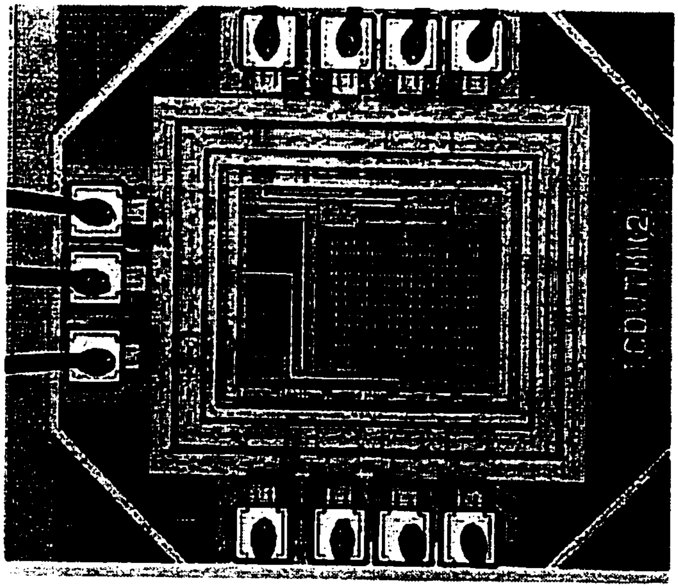


Figure 3.9: Photo-micrograph of the 2-bit HiCapCS bus chip.

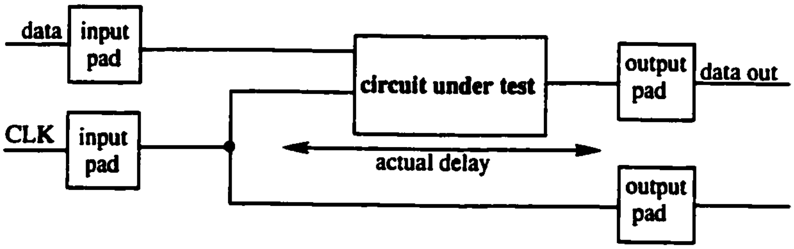


Figure 3.10: Method used for measuring the on-chip circuit delay.

3.12. The figure illustrates the high-performance of HiCapCS down to 2V. The chip was found operational down to 1.2V.

3.4 ROM Bit-Line Design Using HiCapCS

ROMs are important components in many digital systems such as microprocessors, digital filters and digital signal processors. Unlike a Random Access Memory (RAM) where both read and write operations are possible, a ROM is used to store permanent information such as microinstructions in a microprogram memory [46]. A typical ROM

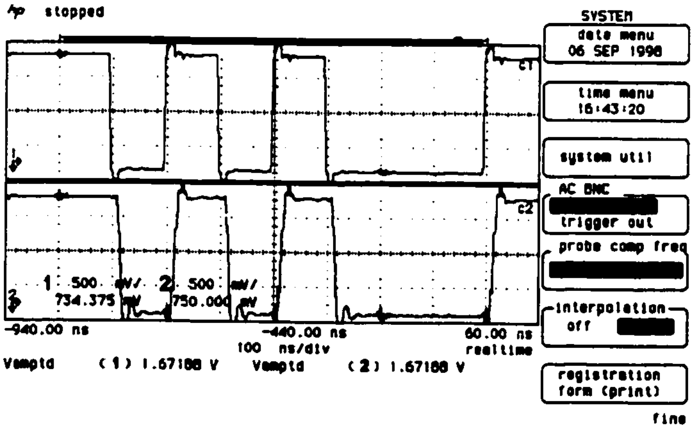


Figure 3.11: Experimental input (top) and output (bottom) waveforms of the internal bus chip.

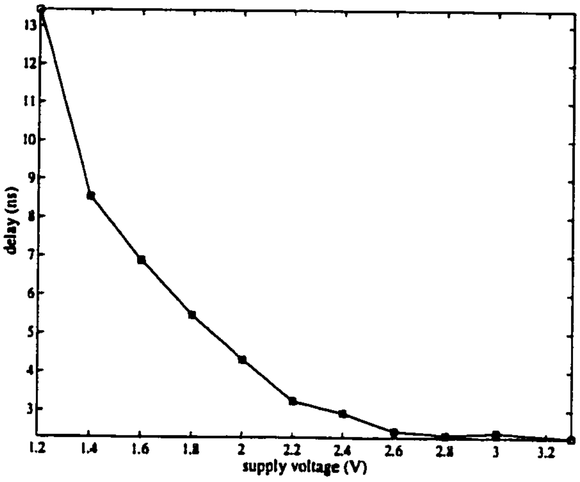
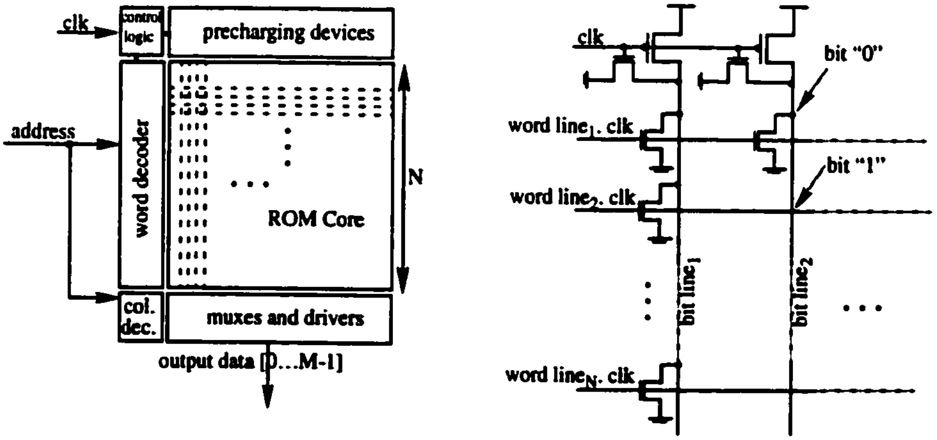


Figure 3.12: Measured delay of HiCapCS bus versus supply voltage.

block diagram is shown in Fig. 3.13a. The word decoder activates one row at a time. Similarly the column decoder and multiplexer select *M* columns (out of the total number of columns in the array) where *M* is the word size. The control logic generates the timing signals such as precharge enable, sense amplifier enable, etc. The precharge devices are used to precharge the bit-lines to "high" during the first half of the cycle. A ROM bit-line can be constructed using either NAND or NOR style. The NAND style is slower



(a) (b)

Figure 3.13: (a) Typical ROM architecture, (b) NOR-based implementation

because it requires serial connections of devices. On the other hand, the NOR style (Fig. 3.13b) consumes more power since it has higher switching activity. In this work, we focus on NOR arrays since they have shorter access time and are the most frequently used [47].

In actual ROM layout, the array can be initially designed with NMOS transistors at *every* row-column intersection. The "1" bits can be then realized (programmed) by omitting the connection between the bit-line and the drain of the corresponding NMOS as illustrated in Fig 3.14a. Since metal-3 is used to construct the bit-lines (in order to reduce their RC delay), this coding approach involves whether or not to add VIA-2 (for "0" bits programming) in the final stage of the fabrication process. This scheme offers short turn-around-time since base process can be completed just before VIA-2 etching and the remaining steps are few [48]. However, it has poor density since diffusion area and contact must be separated for each ROM bit cell.

Another method for ROM programming is shown in Fig, 3.14b. In this approach, bit-line connection can be shared by every adjacent pair of cells. Coding "1" bits is realized by omitting the gate poly and replacing it with a metal-1 line. This technique has higher density than the previous one because contacts are shared. As a result,

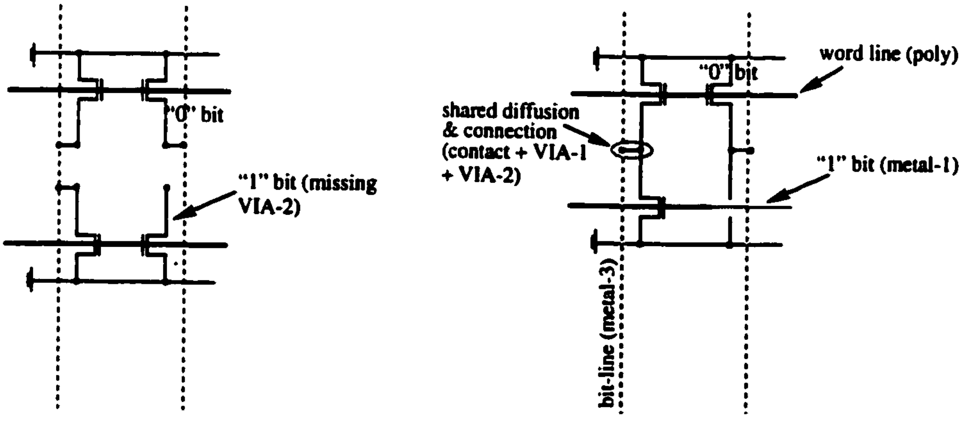


Figure 3.14: (a) VIA-2 programming (b) poly programming

the total parasitic resistance and capacitance of a bit-line using this scheme is potentially less than that using the VIA-2 technique for a random ROM. Regardless of the programming scheme used and in order to save area, the transistors in two adjacent rows are arranged to share a common ground line, and minimum width is usually used for these transistors.

Notice that in a ROM a significant portion of the power is dissipated in the bit-lines. This is because all bit-lines are precharged and then discharged every clock cycle. This power can be saved by using the *selective bit-line precharge* method [49]. With this scheme, only bit-lines which will be accessed are precharged according to the column address decoding. Another advantage of selective precharge is that it eliminates signal coupling to selected bit-lines which can happen in the non-selective case. To demonstrate this, assume that all bit-lines are charged high and that a "1' bit is to be read from bit-line *a*. Further assume that bit-line *a* is "sandwiched" between bit-lines *b* and *c* which are not selected and have "0" bits. During the evaluate phase, the low-to-high transitions on bit-line *b* and *c* can be coupled to the high value on bit-line *a*, resulting in an erroneous output.

Even with the selective precharge method, a (selected) ROM bit-line has to be *fully* charged and discharged every clock cycle. Given the large bit-line capacitance, an excessive amount of energy is consumed as a result. In this research, we use HiCapCS to minimize the swing and so energy of ROM bit-lines. Notice that unlike the bus case, the conventional charge sharing model (Fig. 3.1b) can not be easily extended for implementing the ROM bit-line. The next section describes the new circuit followed by comparisons to conventional dynamic NOR-based ROM designs (Fig. 13.3b). Finally, implementation details and testing results of a prototype 4Kx16b ROM chip in a 0.6 μm process will be described.

3.4.1 Circuit Design

Fig. 3.15 shows how the HiCapCS scheme can be applied to the bit-lines in a ROM. A ROM bit line forms a very large distributed NOR gate. The circuit in Fig. 3 represents a ROM with a number of sub-blocks. Each sub-block has a number of columns (bit lines) and two reference columns. The high reference column (*HRC*) has all transistors con-

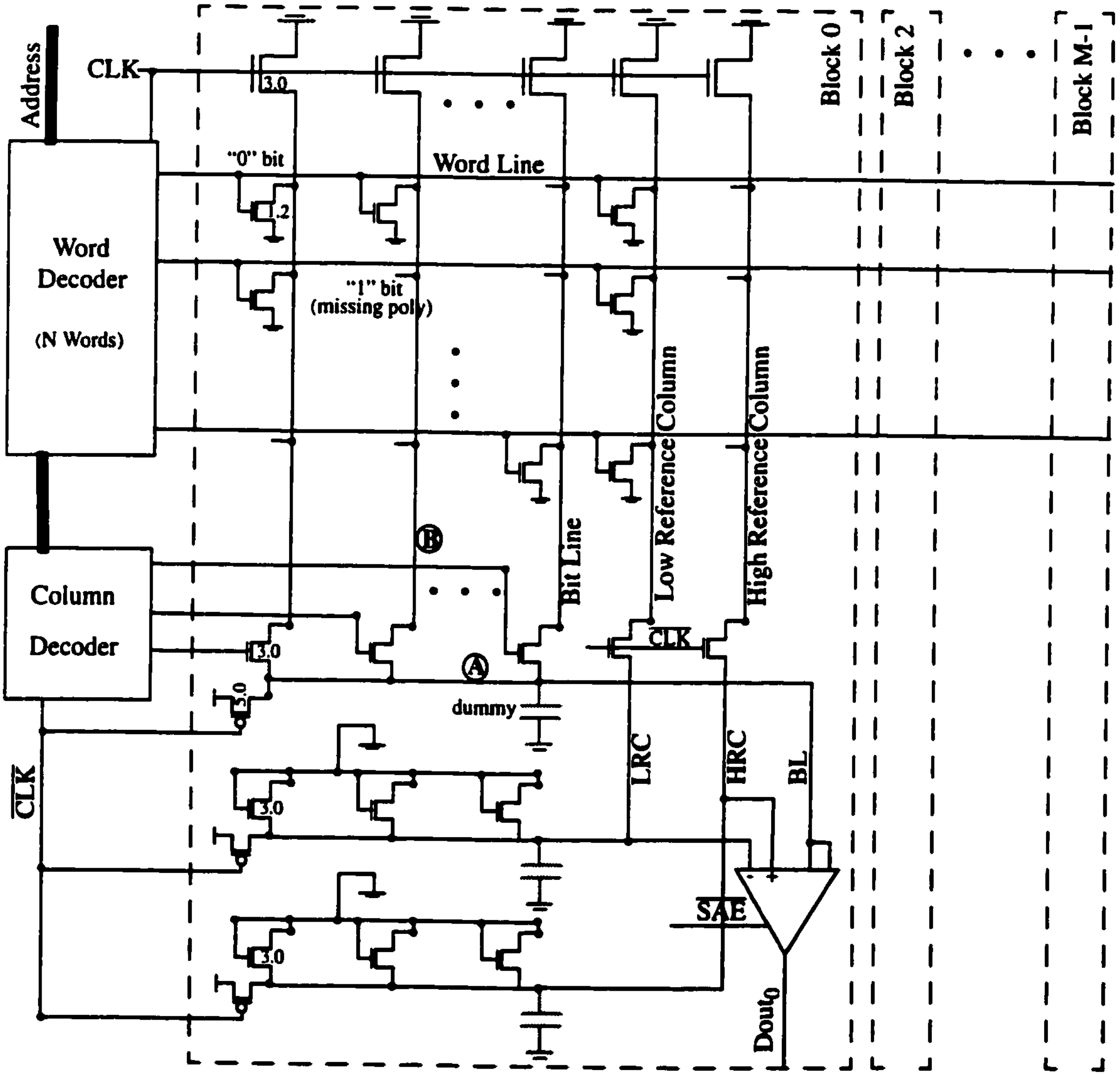


Figure 3.15: ROM bit-line design using HiCapCS

nected to it permanently turned off and the transistors of the low reference column (*LRC*) are all connected to their corresponding word-lines. The capacitance of node *A* (call it C_A) is much smaller than that of the bit-line, and it represents the drain capacitance of the column select transistors plus the wiring capacitance. If needed, additional dummy capacitor can be used to create the required swing on the selected bit line.

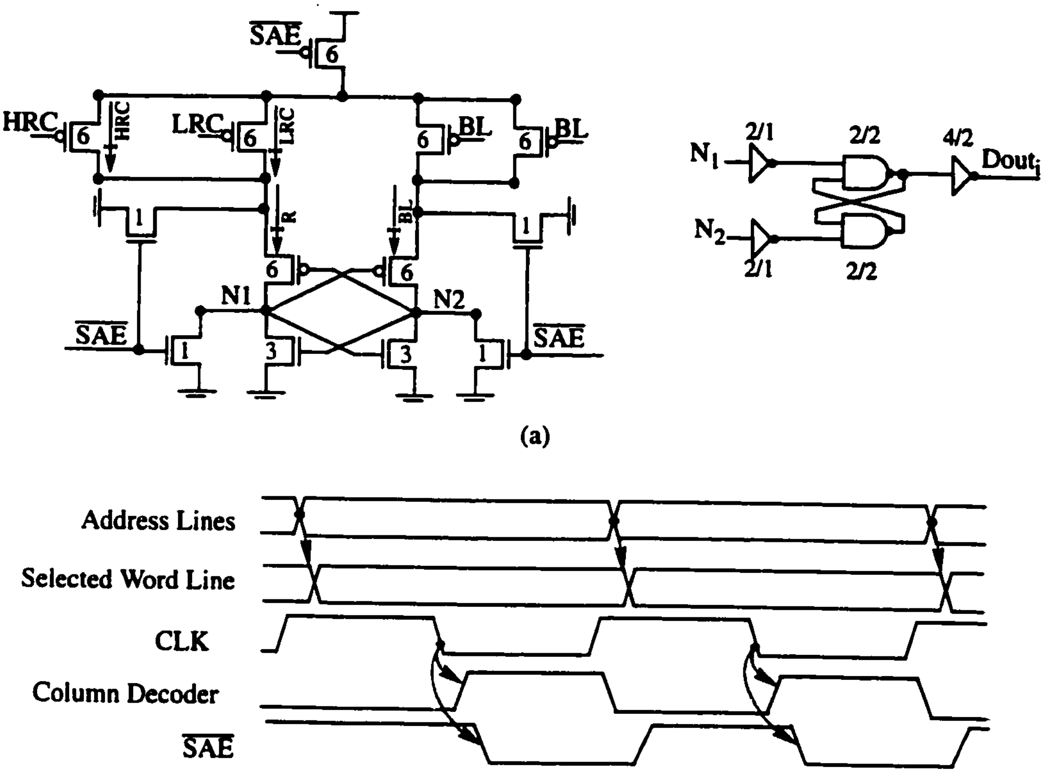
The circuit operates as follows. When *CLK* is high, the bit lines are fully discharged to zero and C_A is charged to V_{dd}. At the same time, the word decoder computes a word address and activates all transistors connected to the corresponding word line. When *CLK* = low, the charge on C_A will be shared with one bit line as selected by the column decoder. If the intersection between this bit line and the enabled word line has an on transistor ("0" bit), the swing on the bit line will be quickly discharged. On the other hand, if a "1" bit is to be read, the swing will be kept. In either case, the *HRC* will have high swing and the *LRC* will discharge the swing.

The ROM data is restored by a dual-reference PMOS current-latch sense (DR-PCLSA) amplifier given in Fig. 3.16a. This amplifier is similar to the one shown in Fig. 3.3 except that it has dual reference lines (instead of one). When $\overline{SAE}$ is high, nodes *N1*-*N4* are pre-discharged to low. When $\overline{SAE}$ falls, the sense amplifier is activated and the value on the bit line is converted to the current difference between I_{BL} and I_R. Here two cases are possible. In case 1, the value to be read is "1" (and so the bit line keeps the swing as a result of charge sharing). Therefore, the following inequality holds:

$$(I_{BL} = 2 \times I_{HRC}) < (I_R = I_{HRC} + I_{LRC}) \tag{3.6}$$

In case 2, the value to be read is "0" (and so swing on the bit line will be discharged through the NMOS transistor turned on by the high word line). This corresponds to:

$$(I_{BL} = 2 \times I_{LRC}) > (I_R = I_{HRC} + I_{LRC}) \tag{3.7}$$



(a)

(b)

Figure 3.16: (a) DR-PCLSA, (b) ROM timing diagram.

The two back-to-back inverters latch will sense this difference in the currents and transfer it into full-wing CMOS values. After it evaluates, the latch prevents any DC power dissipation. The cross-coupled NAND latch keeps the data value after $\overline{SAE}$ turns high again. As in the HiCapCS bus case, the $\overline{SAE}$ signal is a delayed version of the *CLK* signal. The delay required is the amount of time needed for the low-swing to be generated on the bit-line and then discharged (for reading "0" cells). The timing of the new ROM circuit is shown in Fig. 3.16b.

The circuit in Fig. 3.15/3.16 was simulated using HSPICE with a 0.6 μm 3-metal process. The transistors sizes used in the simulation are given in the figures. Metal-3 was used for implementing the bit-line. In the simulation, a typical ROM of 4Kx16b organized as 256 rows by 256 columns (that is, $N = 256$) with a 16-bit per block ($M = 16$) was used. The poly programming approach presented in the previous section was used. With this approach, all bit-lines will have virtually the same capacitance which is important for correct operation of the sense amplifier.

For the above ROM, the extracted bit line capacitance is 523ff and the extracted C_A capacitance is 59ff. The latter represents the lumped drain capacitance of the 16 multiplexor select transistors. With this configuration, the amount of swing generated on the selected bit-line is about 300mv (at V_{dd} = 3.3V) which is more than enough to be detected by the DR-PCLSA. Having a large swing is desirable for the reasons mentioned with the HiCapCS bus. The simulated waveforms at f = 100MHz are shown in Fig 3.17. The simulated power dissipation per bit at the same frequency and V_{dd} = 3.3V is 502µW. In addition to the bit-line power, this number includes the power dissipated by the reference columns and the sense amplifier. As for the delay it has three components: (1) the time it takes to generate the low-swing, (2) the time needed to bring this swing from 300mv to about 200mv (for reading "0" bits), at which moment it is safe to turn on the DR-PCLSA, and (3) the DR-PCLSA delay. The total simulated delay is 1.92ns divided as 0.47ns, 0.79ns, and 0.66ns for the above three parts, respectively.

3.4.2 Comparisons

Energy and delay comparisons between the conventional dynamic NOR-based ROM bit-line [48] (like the one shown in Fig. 3.13b) and the HiCapCS bit-line were performed using HSPICE at V_{dd}= 3.3V. In both cases, a minimum size was used for the bit-line NMOS pull-down transistors. In addition, the parasitic resistance and capacitance of the bit-line (implemented in metal-3) were included in the simulation. For the conventional ROM, the delay is measured as the time it takes the pull-down transistor to

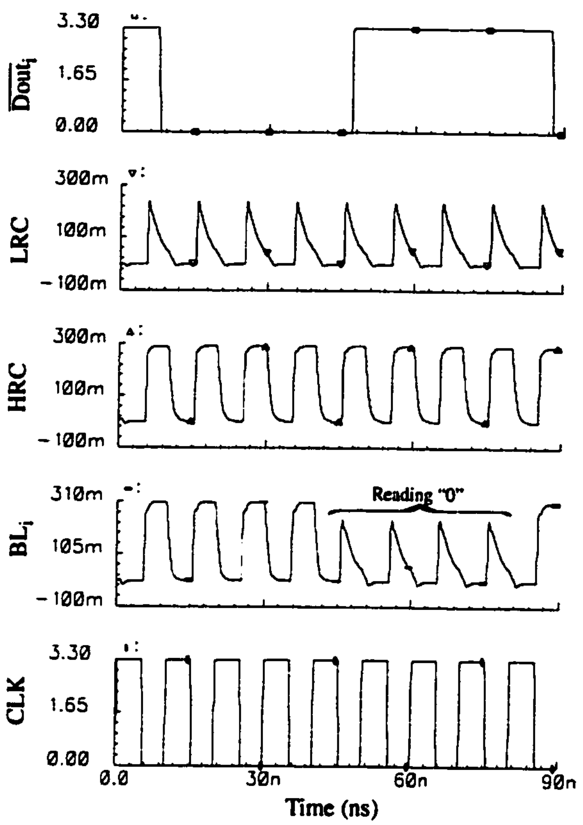
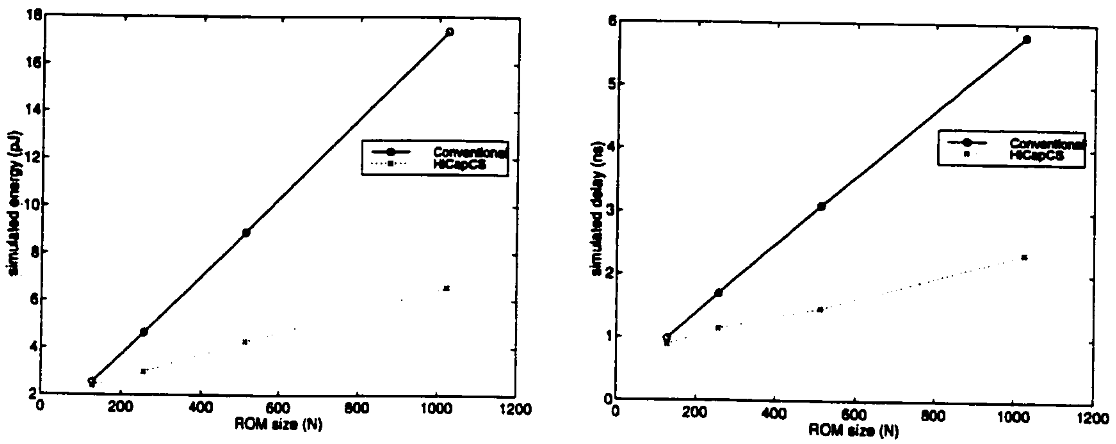


Figure 3.17: HSPICE simulation results for the HiCapCS ROM.

discharge the bit line to $V_{dd}/2$ (for reading "0" bits), and the energy is taken per one bit line (assuming selective precharge). For the proposed circuit, energy and delay were computed as explained previously.

Fig 3.18a and Fig. 3.18b respectively give the energy and delay comparisons using different N values. For the HiCapCS case, the magnitude of capacitor C_A was altered for each memory size in order to produce a low-swing of about 300mv in each case. The figures illustrates that for the range of memory sizes used, the proposed scheme reduces the delay by a factor of 1.1 to 2.5 and energy by 1.1 to 2.7.

Unlike the conventional bus transceiver where the driver/receiver can be properly sized for minimum delay, the ROM pull down transistors have to be minimally



(a)

(b)

Figure 3.18: Comparisons to conventional ROM design: (a) energy, (b) delay.

sized for maximum density. As a result, the delay to discharge the bit-line will rapidly increase in the traditional case, particularly for large N (Fig. 3.18b). Notice also that all word lines in the conventional ROM case has to be low during the first half of the cycle so that the selected bit-lines can be charged to V_{dd} (See Fig 3.13b). In the second half of the cycle, the selected word-line will start rising and as it does it will discharge the bit-lines (in case of reading "0"). This is unlike the HiCapCS circuit where the selected word-line is fully enabled during the first half of the cycle. The extra delay to enable the word-line in the conventional case was not taken into consideration in the above comparisons and if so, it will further worsen the delay of the conventional ROM.

3.4.3 Implementation Details

A 4Kx16b ROM array organized as 256 rows by 256 columns was laid out in a 0.6μm 3-metal process in order to perform experimental evaluation of the circuit. Each bit line is about 1mm long and was constructed using metal-3. The layout of the 64K random ROM cells was generated automatically using a program written in the SKILL language of Cadence design system. To reduce layout area, the memory design rules in the 0.6μm technology were chosen for the layout. These rules are much tighter than the logic design rules.

Conventional static CMOS decoder circuitry was used for both the row and column decoders. Only the last stage of the column decoder was designed using dynamic logic with $\overline{CLK}$ being the control signal. This ensures that the low-swing will be generated at the rising edge of $\overline{CLK}$ (falling edge of *CLK*) as given in Fig. 3.16b. The layout of the row decoder was carefully pitch matched to the ROM cell layout.

Beside using the poly programming approach described in the previous section to match the capacitance of the bit-lines, special care was given to the layout of the DR-PCLSA to minimize any mismatch at its differential inputs. Notice that since only one bit-line is selected at a time, signal coupling between adjacent bit-lines is not possible with this approach. In order to eliminate additional coupling from/to the *LRC* or *HRC*, the layout of these columns were slightly distanced from their adjacent bit-lines.

3.4.4 Measurement Results

The photo-micrograph of the ROM chip is shown in Fig. 3.19. The total core area (not including the decoder) is 1.129mm x 1.335mm. To verify the functionality of the chip, a random sequence of addresses was used and the experimental outputs were acquired and compared to the simulated ones using the same address sequence. In all cases, the experimental waveforms matched the simulated ones. An example output waveform (bottom) using a clock frequency of 100MHz and supply voltage of 2.0V is shown in Fig. 3.20. Fig 3.21 gives the measured delay from $CLK = 0$ to data out using different supply voltages. A similar method as depicted in Fig 3.10 was used for delay measurements. Fig. 3.21 illustrates the high-performance of the proposed circuitry down to 1.2V.

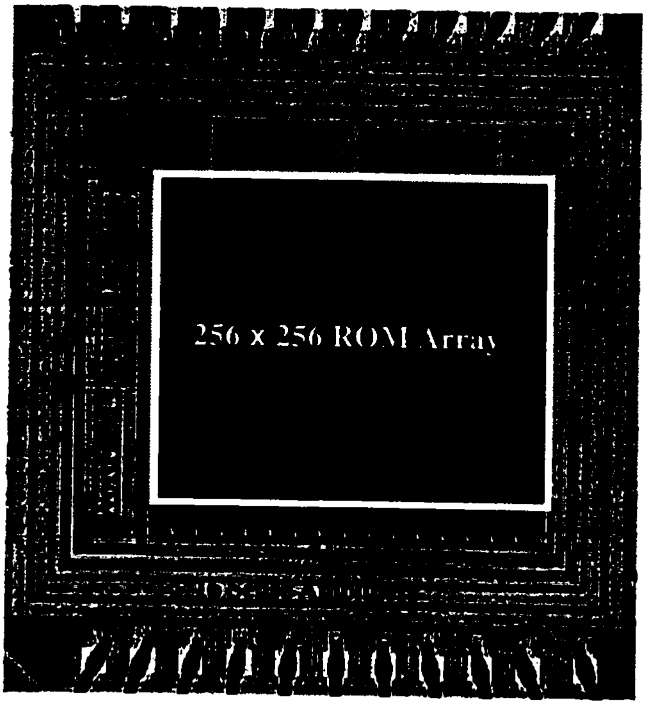
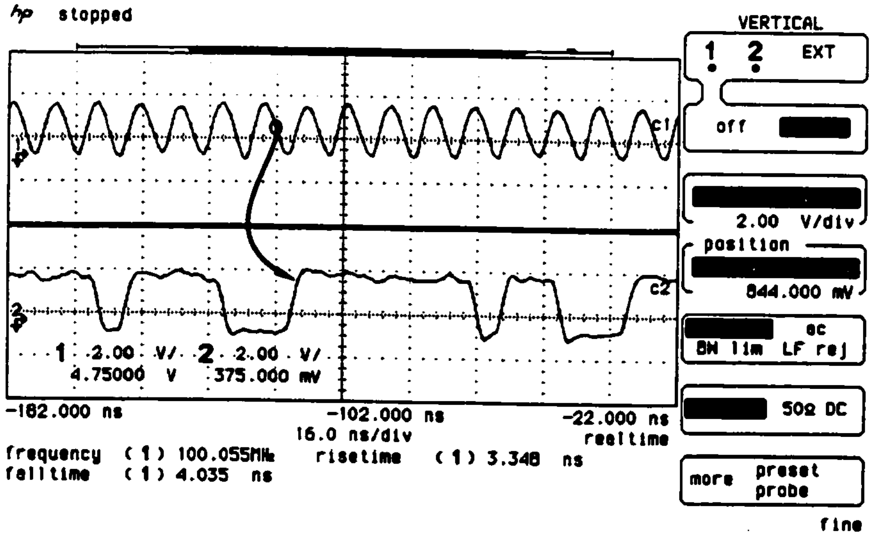


Figure 3.19: Photo-micrograph of the 4Kx16b HiCapCS ROM chip.



```
1  2.00 V/  2  2.00 V/
4.75000 V   375.000 mV
-182.000 ns            -102.000 ns            -22.000 ns
                        16.0 ns/div            realtime
frequency  ( 1 ) 100.055MHz   risetime  ( 1 ) 3.348  ns
falltime   ( 1 ) 4.035  ns
2.00 V/div
844.000 mV
```

Figure 3.20: Experimental waveforms: CLK (top) and ROM data output (bottom).

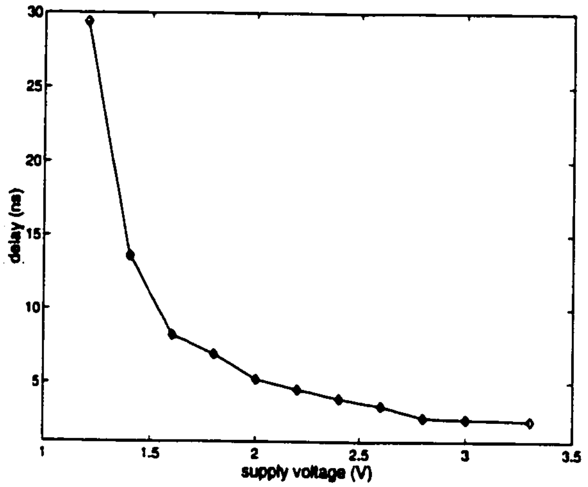


Figure 3.21: Measured delay of HiCapCS ROM versus supply voltage.

3.5 CAM Match-Line Design Using HiCapCS

A Contents-Addressable Memory (CAM) acts like a look-up table in the sense that data is accessed based on its contents rather than on its location (physical address). Applications of CAM include Translation Look-aside Buffer (TLB) which translates from virtual to physical address in a computer system [46], address translation in Asynchronous Transfer Mode (ATM) broadband switches [50], word search in a dictionary processor [51], image pattern recognition in an image processor [52], and data retrieval in a relational database [53].

Fig. 3.22 shows a typical CAM architecture. It consists of a number of memory words (N) with each word having a number of SRAM memory cells (M). Each cell is a conventional 6T SRAM cell plus a 5T comparator. A CAM word stores a tag which can be compared against an input word ($CI_0 \ldots CI_{M-1}$). In a fully associative CAM, all N words are compared (searched) in parallel. A "hit" detection signal is activated if the

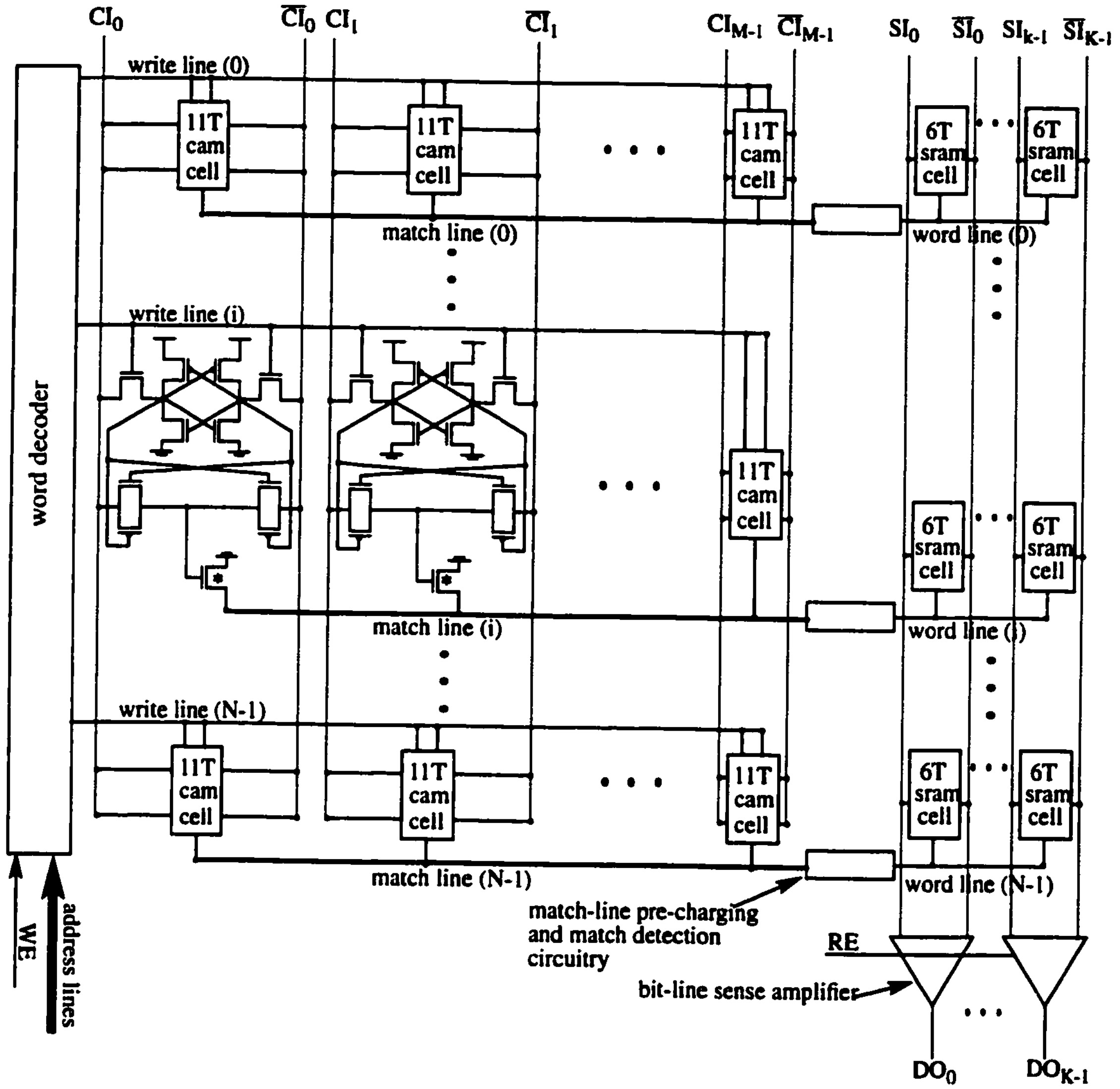


Figure 3.22: Typical CAM Architecture.

data stored in one or more words matches the input data. In case of multiple matches, a match resolution protocol is used in order to select only one match. The match detection signal of the *matched* CAM word is utilized to read an adjacent memory word which is *K* bits wide. In the TLB case, the matched word represents a virtual-address ($CI_0 ... CI_{M-1}$) and the output word is the corresponding physical address ($DO_0 ... DO_{K-1}$).

Similar to a ROM bit-line, a CAM match line is a large distributed NOR gate as illustrated in the Fig 3.22. In case of a match, all transistors marked by (*) in the figure will be turned off, thus enabling the corresponding word line. In the mismatch case, at least one transistor will be turned on and so the associated word line will stay low.

To clarify the above point, Fig. 3.23 gives two detailed implementations of a match line using dynamic CMOS logic [8]. In circuit (a) and during the charging phase (*CLK* = "0"), the bit-lines are predischarged low (that is, $CI_i = \overline{CI}_i$ = "0" for $i = 0,1,\ldots, M\text{-}1$), the match lines are precharged high, and the word lines are driven low by the match detection circuit. During the evaluate phase (*CLK* = "1"), a new data (tag) is applied to the bit-lines and so only the *matched* match line(s) will remain high while the

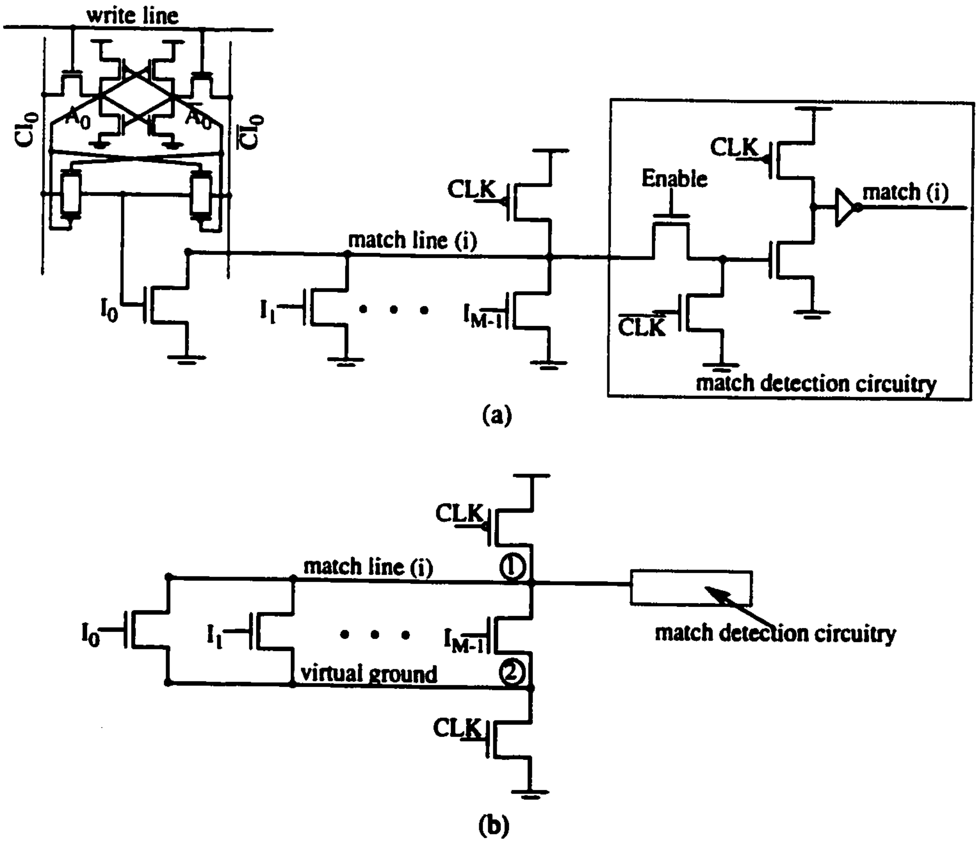


Figure 3.23: Dynamic CMOS implementations of the match line

rest of the lines in the CAM array will be discharged. In the match case, all $A_i = CI_i$ while in the mismatch case, there exists at least one $A_i \neq CI_i$ and so node I_i will be high. After all match lines evaluate and become stable, the *Enable* signal, which is a delayed version of *CLK*, is asserted. Delaying the *Enable* signal is necessary to prevent false glitch at the output.

The gate in Fig 3.23b works in a similar fashion as the one in Fig. 3.23a except that it eliminates the need for bit-line predischarging by using a virtual ground line as shown in the figure. This will however slow down the gate during the evaluation phase. On the other hand and unlike circuit (a), inputting new data to the bit-lines and precharging the match lines are performed simultaneously (when $CLK = 0$) and this might offset the delay introduced by the virtual ground line. From power viewpoint, the virtual ground line has to be charged and discharged to $(V_{dd} - V_{th})$ almost every clock cycle.

Irrespective of the implementation style used, significant energy is consumed per cycle. This is because in a (fully associative) CAM array, N-n match lines have to be completely charged and discharged for each search operation, where n is the number of the matched match lines. In the TLB case, $n = 1$. Reducing the power dissipation of a CAM array is thus vital for low-power designs. As an example, it was found in a recent study that 17% of the total power in the StrongArm processor is consumed by its CAM-based Translation Look-aside Buffer (TLB) [54].

To address the power problem in CAMs, some new techniques have been developed over the last few years. In the selective precharge approach [55], a *small* section of a match line is first charged and then evaluated. The other larger section is charged and later evaluated only if the small section is not discharged. Since this event has low probability for random input data, sizable power savings are possible with this technique. However, the sequencing of two charging and discharging operations increases the worst case delay.

Because the speed of a CAM is the major concern in some applications like in TLBs, novel circuit techniques to reduce match delay were also investigated as in [56] and [57]. Both of these schemes employ differential sensing for fast match detection. The method in [56] suffers from: (1) area increase since it requires two signal lines and one reference line per match word, and (2) high power dissipation since, in addition to the dynamic power consumed by charging (and discharging) the match lines, all *N* sense amplifiers have to be turned on every clock cycle. The technique in [57] improves upon [56] in terms of area by using two lines per match word (instead of three), and in terms of power by only enabling the sense-amplifier of the *matched* match line. However, the power dissipated for match detection is still high since both the signal and the reference lines of each CAM word have to be pulled to large swings of approximately 60 and 40% of V_{dd}, respectively. In addition and similar to [57], a pulsed operation has to be employed to cut DC current in the match lines and the sense-amplifier.

In this thesis, we propose a new match line circuit using the HiCapCS approach. Like the ROM bit-line case, the conventional charge sharing model can not be easily extended for this application. The proposed circuit requires neither a virtual ground line (as in Fig 3.23a) nor bit-line predischarging (as in Fig. 3.23b). Furthermore, it uses a single ended sense amplifier without much penalty in speed. In the following, the proposed HiCapCS circuitry is explained. Comparisons to conventional dynamic NOR-based CAM designs are then presented. Finally, implementation details and measurement results of a prototype 32x32 TLB chip in a 0.6μm process are described.

3.5.1 Circuit Design

The HiCapCS implementation of a CAM match line is shown in Fig. 3.24a. In the figure, transistors M_1-M_3, capacitor C_A, and the pull-down transistors form a dynamic NOR gate. Transistors M_4-M_8 and inverter *I* act as a single-ended amplifier. The match line capacitance *(C_{ML})* represents the drain capacitance of the pull-down transistors plus that of M_2 and M_3, the wiring capacitance of the match line, and the gate capacitance of transistor M_5. The gate works as follows. When CLK = "1" (charging phase), M_1 and M_3 are on, and M_2 is off. As a result, capacitor C_A is charged to V_{dd} and node A is discharged. While C_A is being charged, new input data can be applied to the pull-down NMOSs. At the same time, signal SAE is low making node B high and the output of inverter I low.

When CLK becomes low (evaluation phase), M_1 and M_3 turn off, and M_2 turns on. Two cases are possible; namely match and mismatch. In the match case, none of the pull-down NMOSs are turned on and so the charge on C_A will be *shared* with the match line capacitance C_{ML}. Thus the final voltage on both A and ML will be given by EQ 3.5,

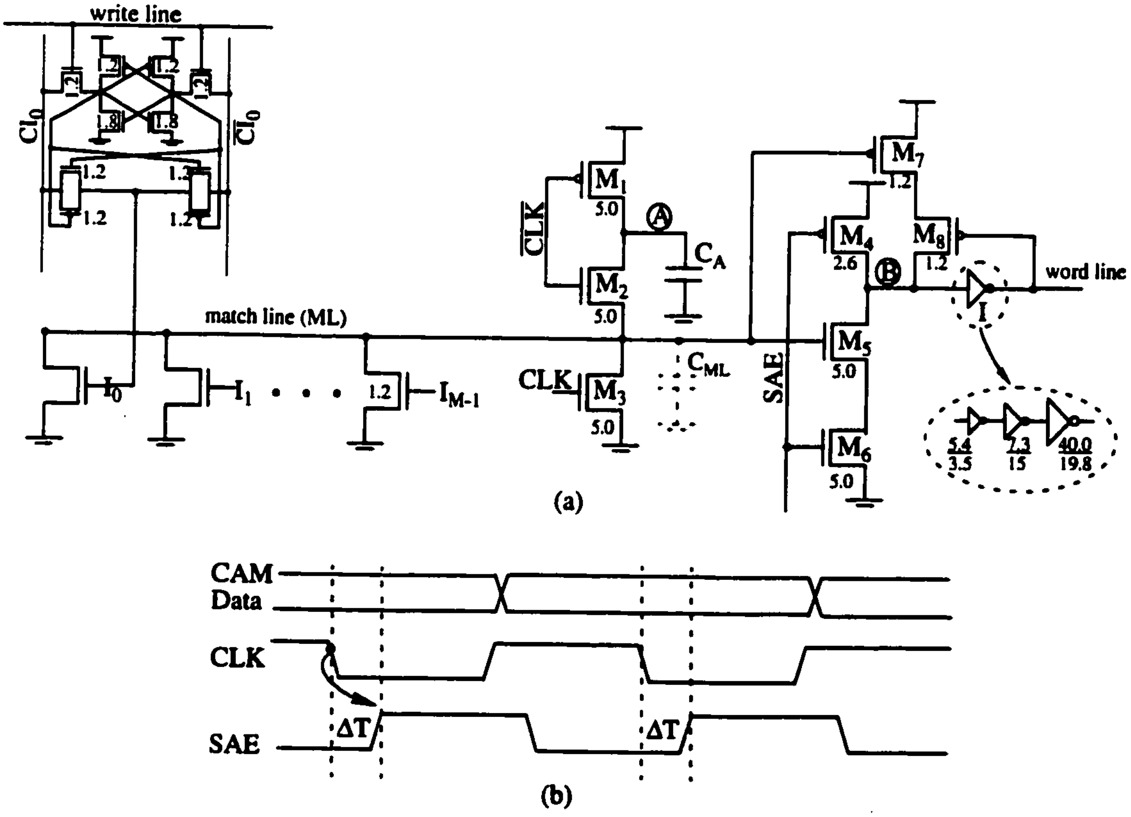


Figure 3.24: (a) CAM match line design using HiCapCS, (b) Timing diagram.

with *LC* represents capacitor C_A (plus the small diffusion capacitance of transistors M_1 and M_2) and *HC* is the total match line capacitance (C_{ML}).

In the mismatch case, at least one of the inputs $I_0, I_1, \ldots, I_{M-1}$ is high. Therefore, C_A will be completely discharged. Since discharging of C_A goes through M_2 and then node *ML*, and in the worst case only one (minimally sized) pull down transistor is turned on, node *ML* can become momentarily high. In order to prevent a false glitch from appearing at the output, signal *SAE* is delayed from *CLK* by ΔT (see Fig. 3.24b).

After *SAE* is activated, node *B* will either get discharged through M_5 and M_6 and the final output will go high for the match case, or it will retain its charge and the output stays low for the mismatch case. In the later case, the feedback transistors, M_7 and M_8, are used to maintain a high-level at node *B* after *SAE* becomes high. With this configuration, only the *matched* match line will dissipate power in the sense amplifier.

Unlike the HiCapCS bus and ROM cases, the value of capacitor C_A has to be carefully selected because a single-ended sense amplifier is used to cut down power dissipation and area overhead as mentioned before. For the match case, the magnitude of C_A determines the voltage at node *ML* as explained before. In order to turn transistor M_5 on, the resultant voltage on the match line which is $\left(\frac{C_A}{C_A + C_{ML}}\right)V_{dd}$ has to be at least as large as the threshold voltage (V_{th}) of transistor M_5. In other words, C_A has to be greater than $\frac{V_{th}}{(V_{dd} - V_{th})}C_M$. Using the 0.6μm CMOS technology with $V_{th} = 0.69$ V and $V_{dd} =$ 3.3 V, C_A is about one fourth of C_{ML}. However, to speed-up discharging of node *B* and to account for positive V_{th} and negative V_{dd} variations, C_A has to be increased above this lower bound. Assuming 10% drop in the supply voltage and 10% increase in the threshold voltage [58], C_A has to be approximately $C_{ML}/3$. A value of $C_{ML}/2$ was used to further enhance the speed.

Notice that in the match case and since $V_{th} < V_{ML} < V_{dd}$, transistor M_5 will operate with a small $(V_{GS} - V_{th})$ and so discharging of node B might be slow. This effect can be lessened by minimizing the charge at node B through reducing the gate capacitance. One way to do that is by using a buffering chain which isolates the possibly large output load from node B (see the onset in Fig. 3.24a). By properly sizing the inverters in the chain, minimum delay can be also achieved.

The power advantage of the proposed gate comes from several sources. First, it employs a low-swing using HiCapCS. Second, neither the logic gate nor the associated sense amplifier dissipates any DC power component. In addition, the gate does not require pre-discharging of the input bit-lines as in Fig. 3.23a. Finally, the elimination of the virtual ground line shown in Fig 3.23b not only reduces power dissipation but also decreases delay.

The circuit in Fig. 3.24 was simulated using HSPICE with a 0.6 µm 3-metal process. The transistor sizes used in the simulation are shown in the figure. Metal-2 was used to implement the match-line. In the simulation, we assumed $N = M = K = 32$. The dummy capacitor C_A was implemented using the drains of 16 NMOS diode connected transistors (that is, for each transistor $V_g = V_s =$ "0"). This will produce a C_A which is half the capacitance of the match line (C_{ML}). Furthermore, capacitor C_A will track any change in C_{ML} due to process or temperature variations.

For the above match-line, the extracted bit line capacitance is 197.8ff and the extracted C_A capacitance is 109.5ff. With this configuration, the amount of swing generated on a match line is about 1.1V (at $V_{dd} = 3.3$V). The simulated waveforms at 100MHz are shown in Fig 3.25. The simulated power dissipation per match line at the same frequency and V_{dd} = 3.3V is 209μW. The delay to activate the word line (associated with the matched match line) is 1.5ns taken from the time $\overline{CLK}$ becomes high.

3.5.2 Comparisons

Energy and delay comparisons between the conventional dynamic NOR-based matchlines in Fig 3.23 and the HiCapCS match line were performed using HSPICE at 3.3V supply. In both cases, a minimum size was used for the match-line NMOS pull-down transistors. In addition, the parasitic resistance and capacitance of the match-line (implemented in metal-2) were included in the simulation. Finally, the worst case delay is measured as the time it takes *one* pull-down transistor to discharge the match line below V_{th}.

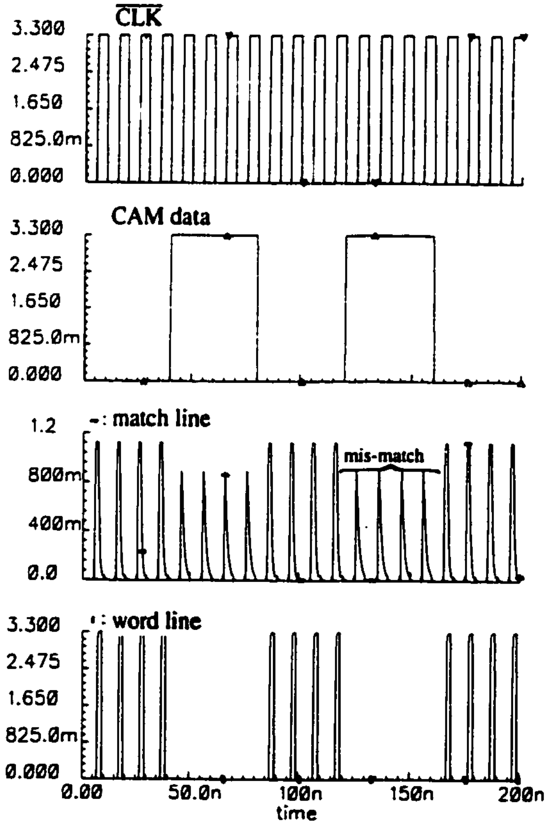
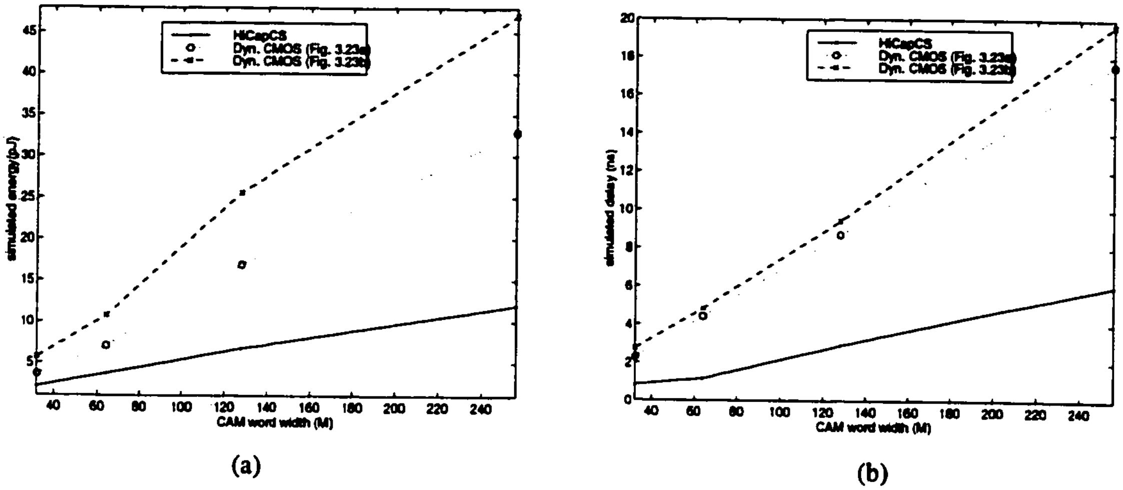


Figure 3.25: HSPICE simulation results for the HiCapCS match line.



(a)

(b)

Figure 3.26: Comparisons to conventional CAM designs: (a) energy, (b) delay.

Fig 3.26a and Fig. 3.26b respectively give the energy and delay comparisons using different M values. For the HiCapCS case, the magnitude of C_A was altered every time in order to make it half of the match line capacitance, as decided in the previous section. The figures illustrates that for the range of memory sizes used, the proposed scheme is superior in terms of both energy and delay as compared to conventional circuits. Note that the energy consumption of the circuit given in Fig. 3.23a is actually worse than is shown by Fig 3.26a since the energy needed to pre-discharge the bit-lines was not included in the simulation.

3.5.3 Implementation Details

A 32x32 TLB chip with $N = M = K = 32$ was laid out in a 0.6μm 3-metal process in order to perform experimental evaluation of the HiCapCS match line. The implemented match line circuit is slightly different than the one shown in Fig. 3.23 and is given in Fig. 3.27. The main difference is in the amplifier part where transistors M_9-M_{12} are added (as shown in **bold**) in order to have a self-resetting word-line. When the *SAE* signal is acti-

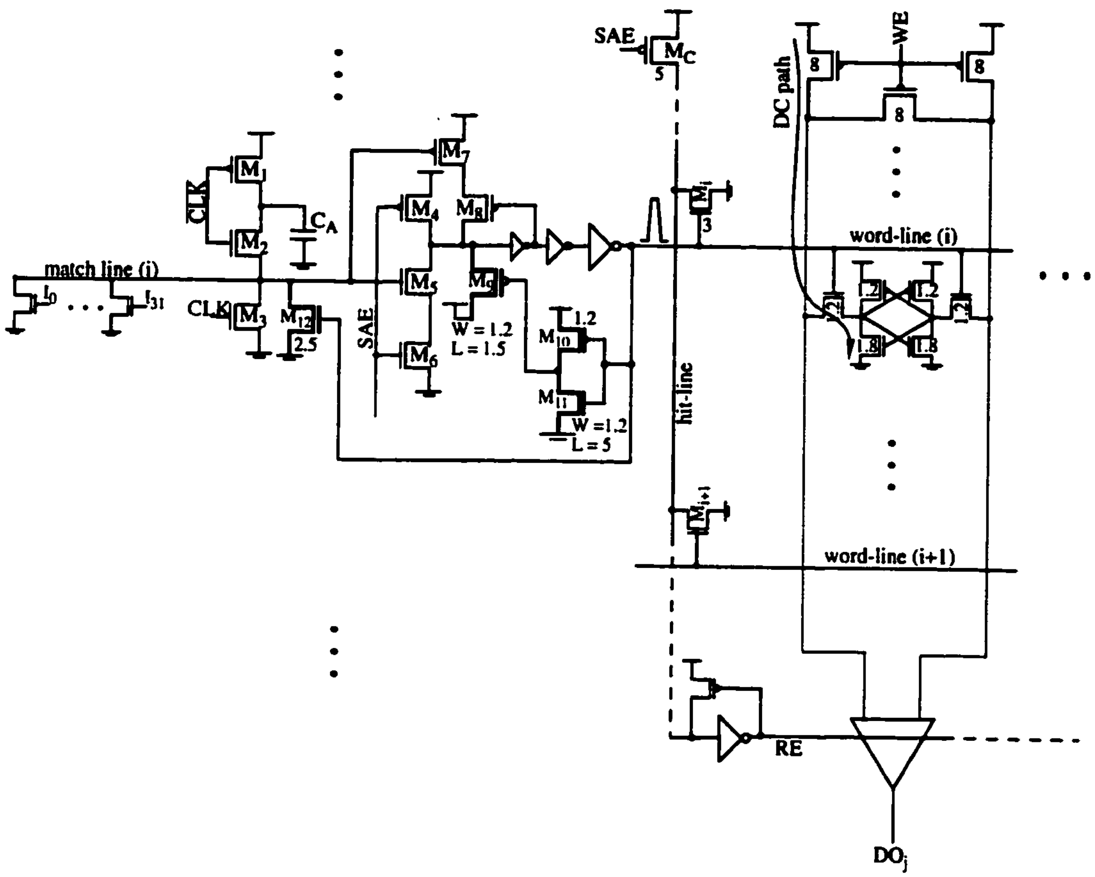


Figure 3.27: Match-line with self-resetting word-line and read enable (RE) circuit.

vated and in case of a match, the corresponding word-line will be enabled. After some delay, the feedback circuitry consisting of the above transistors will reset the word-line to low. In effect, a pulsed word-line is created and this cuts the DC power when reading the associated SRAM word as illustrated in the figure [59][1]. Notice that the lengths of transistors M_{11} and M_9 are increased in order to have a sufficient pulse width.

Fig. 3.27 also shows the method used to enable the SRAM sense amplifiers in case of a match. This is performed by using the distributed dynamic CMOS NOR gate consisting of the charging transistor M_C and the pull-down NMOSs $M_1,\ldots,M_i,\ldots,M_{31}$.

[1] more details on the conventional SRAM read operation are presented in the next chapter.

Only the *matched* match-line will have its word-line enabled and so the associated pull-down NMOS will be turned on. This will consequently discharge the hit-line and hence activate the SRAM sense amplifiers through the *RE* signal. An NMOS current latch sense amplifier (NCLSA) [45] is used to sense the small difference initiated on the SRAM bit-lines. This amplifier works in a similar fashion as the PCLSA described earlier.

The TLB have similar search and write cycles. In addition, both the CAM and the SRAM arrays are written using a single decoder implemented in a standard dynamic NAND-based CMOS. Fig 3.28 gives the timing scenario for the write operation. When a search operation results in no hit, the *WE* signal is activated in the next clock cycle. In addition, new values will be driven on the address lines, the CAM bit-lines and the SRAM bit-lines (See Fig. 3.22). As a result, the write line of the selected CAM word will be enabled hence modifying the CAM word. In the second half of the cycle, the newly written CAM word will produce a match and so the corresponding word line will be activated. This causes a new data to be written in the SRAM word. The *WE* signal is used to deactivate the loads of the SRAM bit-lines during the write operation as Fig. 3.27 shows.

Similar to the ROM chip and to reduce layout area, the memory design rules in the 0.6μm technology were chosen for the layout. In addition, the memory cells in both

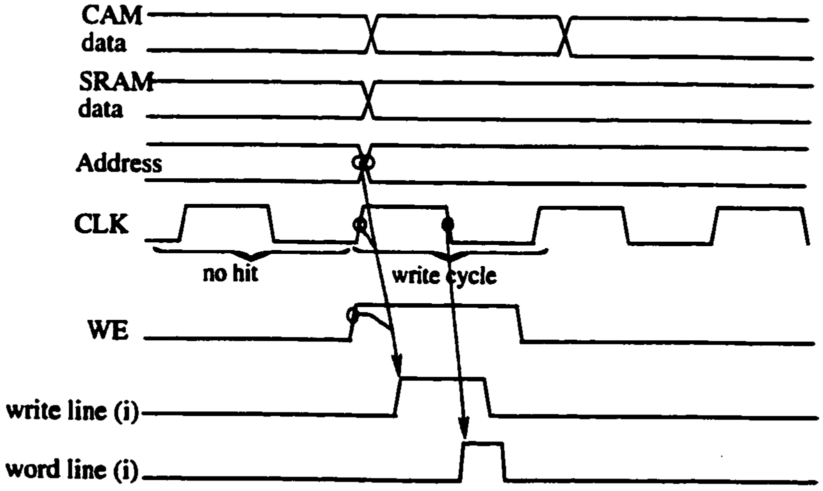


Figure 3.28: Timing diagram for the TLB write operation

the CAM and the SRAM arrays are arranged such that they share common ground and V_{dd} lines in two adjacent rows.

The dummy capacitor C_A was constructed using 16 NMOS diode connected transistors as mentioned before. These transistors were laid out in the first 16 cells of the CAM array with their drains connected using metal-2. Metal-2 was also used for constructing the match line while metal-3 was utilized for the CAM and SRAM bit-lines. In addition, the control signals such as *CLK* and *WE* were distributed using metal-3 since it has reduced RC parasitics. All peripheral circuitries such as: SRAM bit-lines loads and drivers, CAM bit-lines drivers, write-line decoder and drivers, and SRAM sense amplifiers were pitch matched to the CAM/SRAM memory cell. An example layout is given in Fig 3.29 which shows the layout of a CAM cell with a dummy transistor.

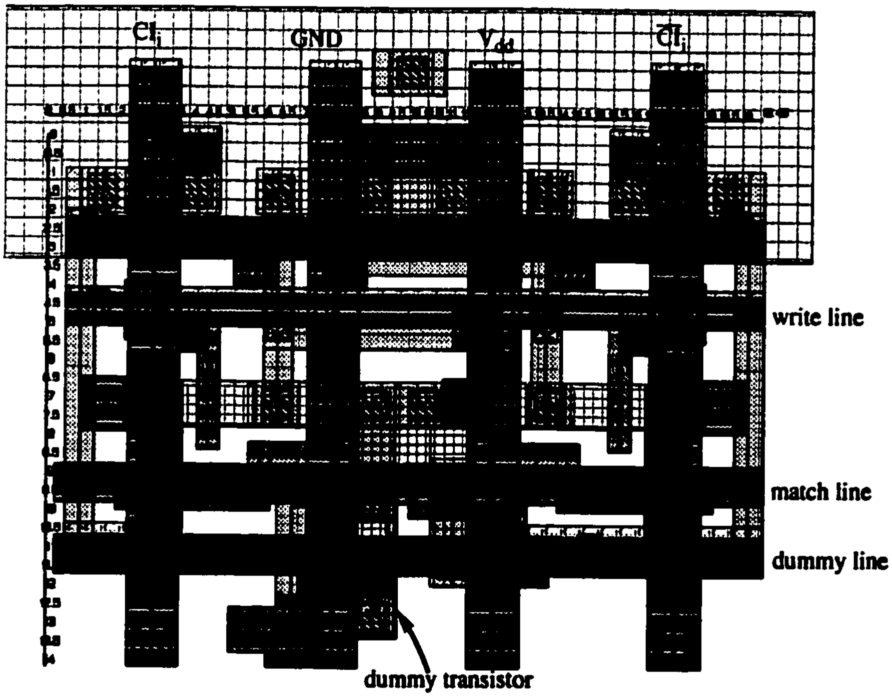


Figure 3.29: Layout of the CAM cell.

3.5.4 Measurement Results

The photo-micrograph of the TLB chip is shown in Fig. 3.30. The total core area is 1.290mm x 0.676mm. To verify the functionality of the chip, the 32 CAM words were initialized (written) using distinct words (virtual addresses). Similarly, unique words (physical addresses) were written into the associated SRAM words. Writing was performed following the scenario described in the previous section. After this initialization step and for each subsequent clock cycle, a random input virtual address $(CI_0, CI_1, ..., CI_{31})$ was driven on the CAM bit-lines and the output physical address $(DO_0, DO_1, ..., DO_{31})$ was acquired. In case of a match, the expected physical addressee was observed and in a mismatch case, there was no change at the outputs.

An example output waveform (bottom) using a clock frequency of 10MHz and a supply voltage of 3.3V is shown in Fig. 3.31a. Fig 3.31b gives the delay at $V_{dd} = 3.3V$

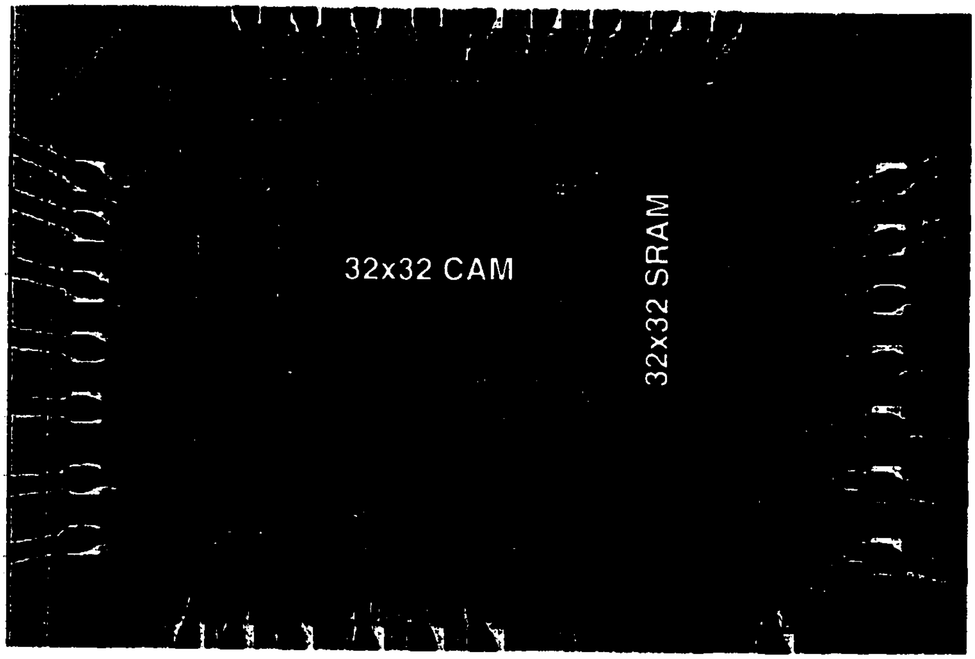


Figure 3.30: Photo-micrograph of the 32x32 TLB chip.

measured from $\overline{CLK}$ = "1" to SRAM output. Fig. 3.32 plots the measured delay (minus I/O pads delay) as a function of the supply voltage. Note that at 3.3V, this delay is about 4.0ns which is comparable to the 4.4ns in [56] and the 2ns reported for the (smaller) 16x32 TLB using a 0.25μm process in [57]. Also Note that unlike the bus and ROM chips, the minimum supply voltage at which the HiCapCS match line can still function correctly is limited due to its single-ended operation.

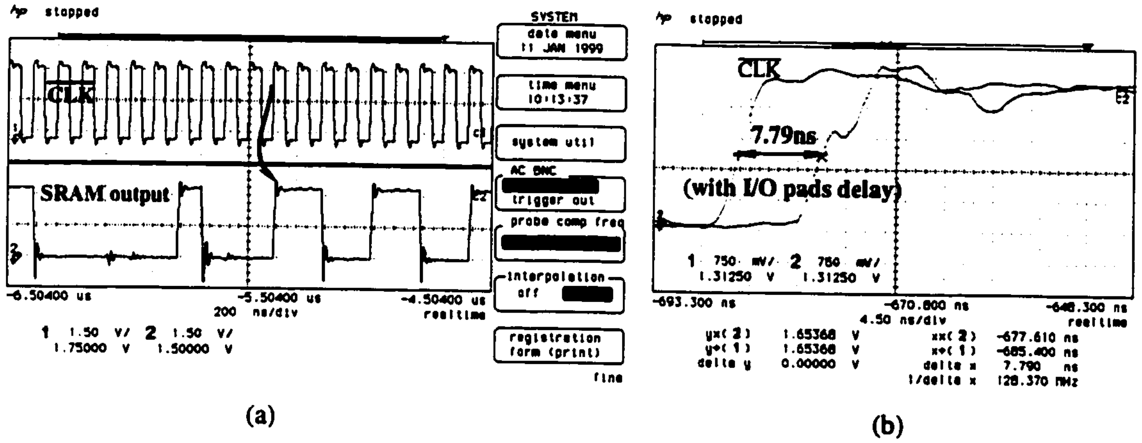


Figure 3.31: (a) Experimental waveforms, (b) Delay from $\overline{CLK}$ to SRAM output.

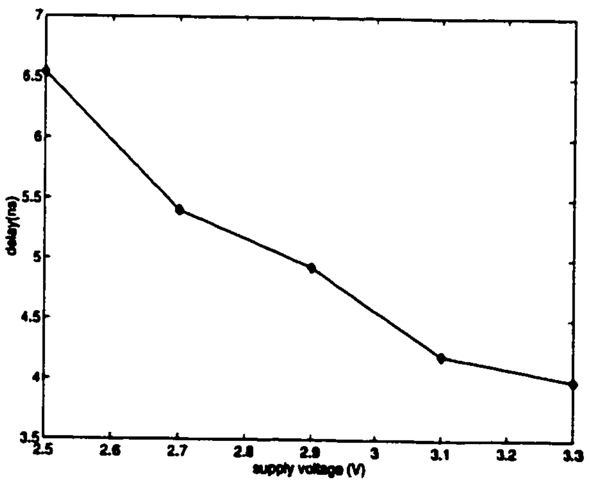


Figure 3.32: Measured delay of the HiCapCS TLB.

3.6 Summary

In this chapter, we presented new circuit techniques for low-power and high-performance design of a class of dynamic digital circuits with high capacitive loads. These techniques are all based on a new charge sharing model, called HiCapCS. With this model, it is possible to reduce the swing of signals running on a heavily loaded node instead of using full-swing as in conventional CMOS styles. To convert the small swing to CMOS signals, single- or double-ended sense amplifiers are used. The proposed scheme is applied to internal bus lines, ROM bit lines, and CAM match lines. HSPICE circuit simulations showed significant improvement over conventional dynamic CMOC circuits in terms of both speed and energy. The functionality and performance of the proposed method were verified through three chips: a 2-bit internal bus, a 4Kx16b ROM, and a 32x32 TLB.

In addition to its performance and power advantages, HiCapCS reduces the noise generated on the supply lines because it requires a small dummy capacitor to be charged every clock cycle as compared to a much larger one in the CMOS case. Furthermore, the approach is simple since it is capable of creating a small swing without using an extra reference voltage generator or a pulse generator. In addition, it does not dissipate any DC power.

On the negative side, the design of a HiCapCS circuit has to be carefully performed in order to ensure that the required swing is produced under process and temperature variations. This is particularly true for the CAM match-line because of its single ended nature. In addition, special care has to be taken to minimize any mismatches that may result in the midst of the layout stage. For example, balanced layout should be employed in the case of the ROM and bus designs in order to reduce the mismatch at the sense amplifier inputs.

The HiCapCS approach requires that node *HC* to be floating for a short period of time before the *SAE* signal is activated. Thus the circuit is susceptible to noise injection during this time. This will further requires more attention during the layout in order to prevent any cross talks from neighboring global lines with full-swing. This problem is not a major concern in the bus and ROM cases, however, because of their differential operation. In addition, the reduced noise immunity might be tolerable within the (ROM/ CAM) memory arrays, where the noise conditions and signal interfaces can be carefully controlled [60].

Chapter 4

Circuit Techniques for LP HP Multi-port SRAMs

4.1 Introduction

Multi-port SRAM memories allow simultaneous read and write operations of several memory words. They are key components in many digital systems that demand high data bandwidth, shorter cycle time and/or multiple operations per cycle. Examples of multi-port memories include register files in microprocessors [46] and buffer memories in digital telecommunication chips [61].

In a microprocessor system, a register file resides in the critical path and so it is essential to improve its speed. On the other hand, the ever increasing system requirements and cooling costs put high demands for low-power. This is notably true in the register file case because it is accessed very frequently during a program execution time. Approaches to improve the speed of register-files via expensive technologies such as

BiCMOS [62] or multi-V_t CMOS [63] have been proposed. Fast read and write operations can be also achieved through complex multi-port cell design as in [64]. Another key factor to high operating frequencies is proper pipelining using a single clock signal [65].

From power viewpoint, the power consumed during a read operation can be reduced via a pulsed word-line which cuts the DC power from the bit-lines loads. A large number of embedded memories manufactured today employ this technique with some variations [59]. However, less effort has been spent to decrease write power and delay. Conventional voltage-mode write operation requires full charging and discharging of the selected bit-lines. This approach not only consumes large energy but also does not meet the performance requirements of today's systems. In addition, it increases peak power dissipation and noise coupling to adjacent bit-lines.

In this thesis, we propose a new method for fast and low-power memory write operation. The basic idea is to initially bias the back-to-back CMOS inverters of the memory cell in the transient region to increase their voltage gain. Then by creating an extremely small swing (~100mV) on the bit-lines, a differential current is injected into the cell. This current difference will in turn be converted to full CMOS voltage levels by the memory cell which exhibits a high gain. The proposed method, henceforth, called Current-Injection Write (CIW) comes at a small increase in the cell area. However, we will show that by using this technique in multi-port SRAMs, the area penalty is actually reduced. Furthermore, CIW incurs no degradation in terms of the cell noise immunity.

In addition, a sense amplifier for low-power and high-speed memory reads is also described. This amplifier works in a similar fashion as the CIW circuitry. These techniques are used in the design of a 3-port 32x64b register file fabricated in a standard 0.35μm CMOS process. Other key approaches to improve the speed such as using a single-phase clock signal for control and fast address decoding utilizing Source Coupled Logic (SCL) [66] are also presented.

The rest of this chapter is organized as follows. In the next section, the basic read and write operations of a conventional SRAM are described. A review of the state-of-the-art in low-power SRAM write techniques are given in section 4.3. The new current-injection write and read circuits are present in section 4.5 and 4.6, respectively. The design and simulation of a 3-port 32x64b register file that utilizes the proposed techniques are given in section 4.6. This is followed by experimental results of the fabricated register file chip and a summary at the end.

4.2 Conventional SRAM Read/Write Operation

Fig. 4.1a gives a generic block diagram of an SRAM memory with $N.M.2^S$ bits. The memory is organized into $N.2^S$ words and each word is M bits wide. Both horizontal (word) and vertical (column) address decoding is used to have a feasible design with acceptable aspect ratio. The S parameter represents the column multiplexing factor. The precharging devices pull the bit-lines to "high" and the sense amplifiers amplifies the small voltage difference generated on the (column) selected bit-lines during a read operation.

Fig. 4.1b gives a more detailed picture of the memory operation. Each storage element is a standard 6T memory cell. The transistors in the memory cell are made as small as possible to increase layout density. Special care has to be taken however when sizing of the access transistor M_{a1} (M_{a2}) relative to the pull-down transistor M_{n1} (M_{n2}) and the pull-up transistor M_{p1} (M_{p2}) in order to guarantee proper read and write operations with sufficient static noise margin.

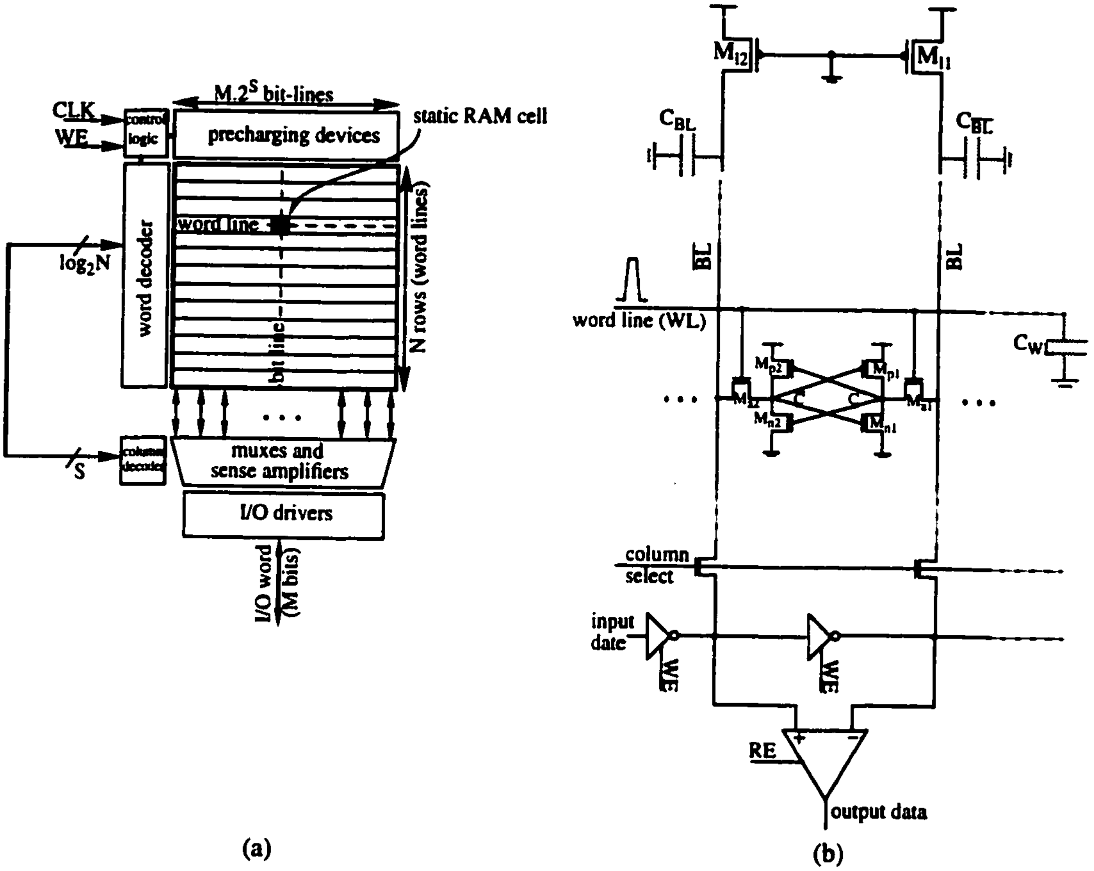


Figure 4.1: Conventional SRAM: (a) block diagram, (b) Read/Write operations.

Referring to Fig. 4.1 and during a read operation (WE = "0"), the asserted word line will turn on the access transistors. Assuming a low value at node C, a DC path will exist through transistors M_{l1}, M_{a1}, and M_{n1}. Since the load transistor M_{l1} has much smaller impedance than the M_{a1} and M_{n1} combination, a small voltage drop will develop on BL. The differential voltage between BL and $\overline{BL}$ is then restored to a full-swing level by the sense amplifier. This differential operation provides fast read access.

Notice that node C will rise by a ΔV value when the word line is turned on. As long as this ΔV is smaller then the threshold voltage (V_{th}) of transistor M_{n2}, the read operation is safe. To satisfy this condition, the cell ratio r defined as $\left(\frac{W}{L}\right)_{M_{n1}} / \left(\frac{W}{L}\right)_{M_{a1}}$ has to be larger than unity. In other words, transistor M_{n1} has to be stronger than transistor M_{a1}. The lower limit of this ratio can be derived by solving the current equations of transistors M_{n1} and M_{a1} such that the above ΔV condition is met. Interested reader is referred to [67] for further analysis.

Because an SRAM cell is designed with read stability and speed in mind, writing a new value to the cell requires almost a *full-swing* on BL (or $\overline{BL}$). Assume that we want to store a "1" on node C (which originally has a "0"), then the necessary condition in this case is that transistor M_{p2} to be weaker than M_{a2}. This is needed in order to pull node $\overline{C}$ to below the threshold voltage of transistor M_{n1} and so turn M_{n1} off. Again detailed analysis is provided in [67].

Note that during a read operation, *all* $M.2^S$ memory cells connected to the selected word-line will dissipate DC current with only M cells consume useful current. A pulsed word line is usually used to cut down this large DC current as mentioned before [59]. The pulse has to be wide enough to allow a sufficient swing (~0.1V-0.3V) to be developed on the bit-lines and then sensed by the differential amplifier.

Finally note that for a large memory (> 64Kb) the parasitic capacitance and resistance associated with the bit- and word- lines will dramatically increase. This will adversely affect both speed and power. Therefore, a large memory is partitioned into smaller blocks and hierarchical word-line (HWL) [68] and/or hierarchical bit-line (HBL) [69] should be employed.

4.3 Low-Power SRAM Write Techniques

In this section a quick review of low-power write techniques is presented. Two of these approaches are illustrated in Fig. 4.2. The variable impedance bit-line load [70] shown in Fig. 4.2a makes all the bit-lines load impedance high in the write cycle. This is achieved by using the write enable signal (*WE*) to cut off the large load transistors; namely M_{l1} and M_{l2}. With this technique, the DC current from the loads to the bit-lines write drivers is significantly reduced since transistors M_{l3} and M_{l4} are much smaller than M_{l1} and M_{l2}. During a read cycle, all load transistors are turned on and so the bit lines will have low impedance, allowing fast precharge (or bit line recovery) for the next cycle. Although effective in cutting the DC power, this scheme still requires full swing on the bit-lines during writes.

Another low-power write technique is the Driving Source Line (DSL) [71] shown in Fig. 4.2b. In DSL, the source nodes of the pull down transistors (of the

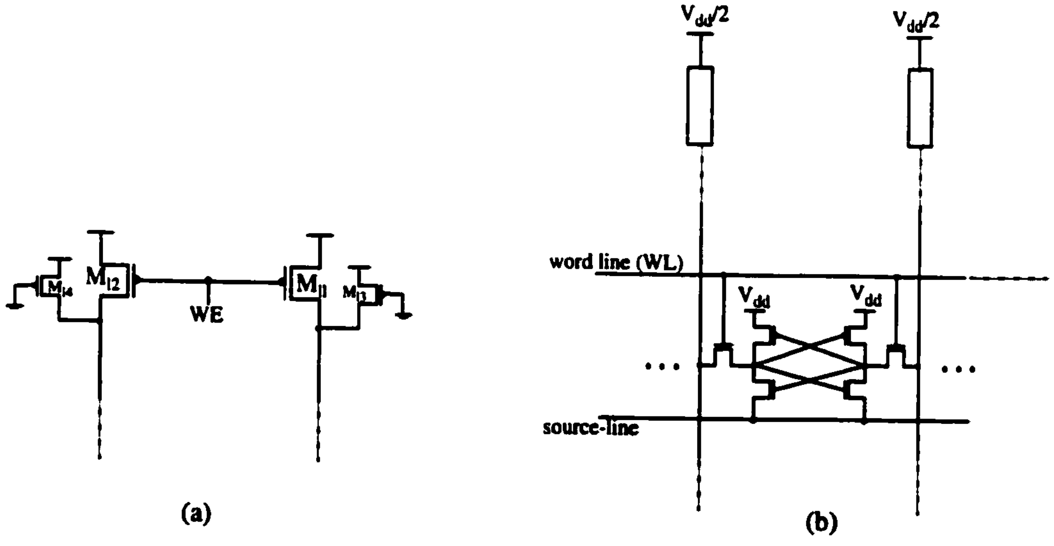


Figure 4.2: Low-power write techniques: (a) variable impedance load, (b) driving source-line (DSL).

selected memory word) are all made floating (Hi-Z) at the start of a write cycle. This will make it easy to release the cell potentials and so a small differential voltage on the bit-lines is adequate to change the stored data. At the end of the cycle, the source line is switched back to "0" which brings the cell potentials to full rail-to-rail values. This low-swing is also largely attributed to the fact that the bit-lines are precharged to $V_{dd}/2$ which realizes a larger V_{GS}-V_T in the access transistors. However, intermediate voltage precharging is disadvantageous from noise point of view. This is because the high-level node of the enabled memory cell can become equal to the bit-line voltage during a read operation. This reduces the cell immunity particularly at low V_{dd} [8]. Another drawback of the DSL technique is that two lines (the word and the source lines) have to be asserted during a write operation.

Other low-swing approaches based on DSL but without half-supply bit-line precharging was proposed in [72] and [73]. In [72] and to enable low bit-line swing, the access transistors had to have zero threshold voltage. In [73], low-swing is achieved by word-line voltage boosting.

A different low-swing write technique was presented in [74]. The idea is to use an almost equal pass and pull-down transistors in the memory cell. This will facilitate writing a new data in the cell with low swing of about 0.5V on the selected bit-lines. This approach suffers from dramatic decrease in noise immunity since (1) it reduces the cell ratio r (defined in the previous section), and (2) the bit-lines are precharged to an intermediate voltage level (0.5V at V_{dd} = 5V) which, like the original DSL method, results in low noise margin.

4.4 Current-Injection Write (CIW) Circuit

4.4.1 Cell Operation

In this section, the CIW approach to high-speed and low-power SRAM cell write is described. The basic idea is illustrated in Fig. 4.3. The figure shows a 7T SRAM cell with separate read and write ports. The write port consists of transistor M_5 along with the *WBL* and $\overline{WBL}$ lines. These lines are shared by all cells in the same memory column. Transistors M_6 and M_7 operate in the linear region and are used to pull *WBL* and $\overline{WBL}$ to V_{dd}, respectively. Transistor M_8 is always turned on and is used to minimize the voltage

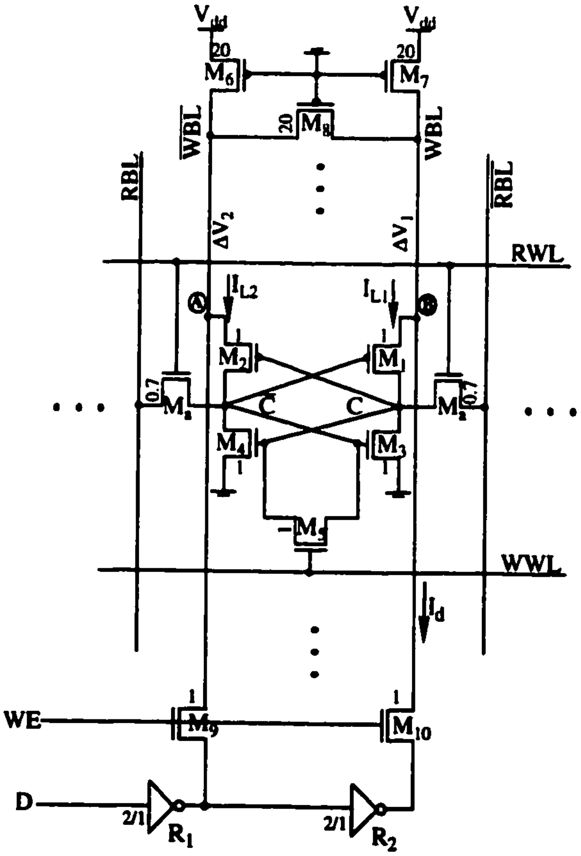


Figure 4.3: Current injection write (CIW).

difference between *WBL* and $\overline{WBL}$. Notice that In a conventional SRAM nodes *A* and *B* (Fig. 4.3) are connected to a common horizontal power supply line.

Writing is performed by injecting a current difference into the memory cell. This is done in two phases: *precharge* and *evaluate*. During the precharge phase, M_5 is turned on by driving *WWL* high. As a result, the voltages on *C* and $\overline{C}$ are equalized to an intermediate level. This in effect will bias the two back-to-back CMOS inverters of the memory cell in the transient region which increases their voltage gain. Furthermore, this equalization will create two equal DC currents (I_{L1} and I_{L2}) along the write bit-lines, thus *both* *WBL* and $\overline{WBL}$ will be pulled *slightly* below V_{dd}.

During the precharge phase also, signal *WE* is high and so depending on the input data (*D*) an additional current I_d is dissipated in one of the data drivers, R_1 or R_2, through the source node of M_9 or M_{10}, respectively. Assuming a low value at *D* and without loss of generality, the NMOS transistor of inverter R_2 will then sink this current via transistor M_{10}, as shown in the figure. This current will *further* pull down the *WBL* node below V_{dd}. Thus a small voltage difference equal to (ΔV_1 - ΔV_2) is created between the source nodes of transistors M_1 and M_2. This, however, should not affect the equality relation between $I_{L1} = I_{L2}$ as long as M_5 is turned on.

During the evaluate phase, *WWL* becomes low thus turning off M_5. As a result, M_1 and M_2 will source different amounts of current since they see different (V_g-V_s) values, where V_s is the corresponding bit-line voltage. Thus a current difference of magnitude (I_{L2}-I_{L1}) is *injected* in the memory cell and this gives rise to a small voltage deviation between *C* and $\overline{C}$. This difference will be quickly amplified to CMOS levels by the two back-to-back inverters of the memory cell which acts as a positive feed-back amplifier with high gain. At the end of the cycle, *WE* is negated to cut off the DC current flowing through M_{10}.

When writing a particular cell, it is important to assure that the stability of other cells in the same column is not affected since they share the same write bit-lines. Any voltage drop on the write bit-lines will be passed to all cells in the same column through their $M_1(M_2)$ transistors. As long as this drop is less than the threshold voltage (V_{th}) of the corresponding M_2 (M_1), then these cells will not lose their stored values.

In the above design, this is achieved in part by making the equivalent impedances of transistors M_1-M_5, M_9-M_{10}, and inverters R_1-R_2 much larger than those of the PMOS loads M_6 and M_7. HSPICE simulations, performed using a 0.35μm CMOS technology with transistors widths shown in the figure and for various memory sizes N (number of rows) and supply voltages, showed a less than $V_{dd}/30$ drop on the *WBL* (or $\overline{WBL}$) as a result of the proposed approach. This drop is much smaller than V_{th} in contemporary processes which is about $V_{dd}/5$, thus ensuring stability of other cells.

In addition, any voltage drop that appears on either *WBL* or $\overline{WBL}$ will turn on transistor M_8 which tries to minimize the voltage difference on the two lines. This will further guarantee that voltage disturbances induced on a write bit-line during the write operation will not affect the stability of other cells. Notice that the "always-on" transistor M_8 is also used at the end of a write operation to equalize the voltages on *WBL* and $\overline{WBL}$ and so prevent any current mismatch during the next write cycle.

Fig. 4.4 gives an example HSPICE simulation results using a V_{dd} = 3.3V, and N = 32. Parasitic resistance and capacitance extracted from the layout were included in the simulations. Notice that the maximum voltage drop on the write bit-lines is less than 0.1V. Further notice that the *WE* pulse has to be slightly wider than the *WWL* pulse for correct write operation. The last point will be further clarified in Section 4.6.

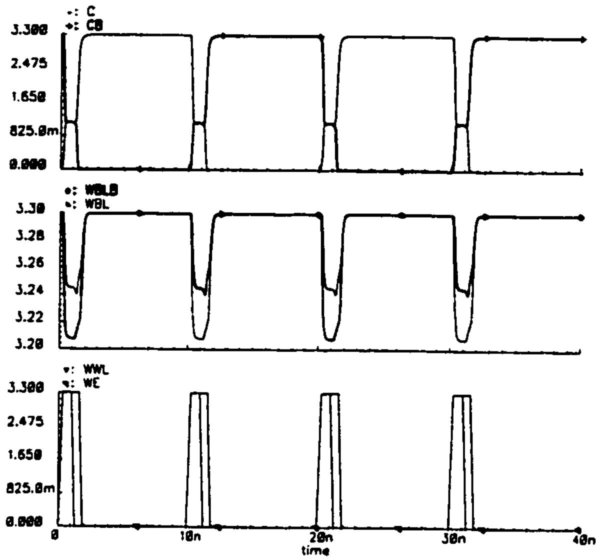


Figure 4.4: HSPICE simulation results of the CIW approach.

4.4.2 Comparisons

As compared to conventional SRAM write operation, the proposed CIW approach does not require (an almost) full charging and discharging of the bit-lines. Hence, this approach greatly reduces both write energy and speed. HPSICE simulations to compare the CIW scheme with the conventional approach for various memory sizes were performed. The conventional approach is similar to the one shown in Fig.4.1 except that a variable load impedance (as in Fig 4.2a) is used to reduce DC power. In addition, large bit-line data drivers were utilized for fast bit-line discharging. For both the conventional and the CIW techniques and to further reduce DC current, minimum width *WE* and *WWL* pulses needed to complete the write operation were employed. Finally, the same transistor sizes were used for both cases: $W_{M1\text{-}M4}$ = 1μm, W_{Ma} = 0.7μm, $W_{M6\text{-}M7}$ = 20μm, and for the proposed scheme W_{M5} = 1.0 μm.

The simulated energy and speed curves for *N* ranging from 32 to 1K words are shown in Fig. 4.5a and 4.5b, respectively. In Fig. 4.5a, each value represents the total

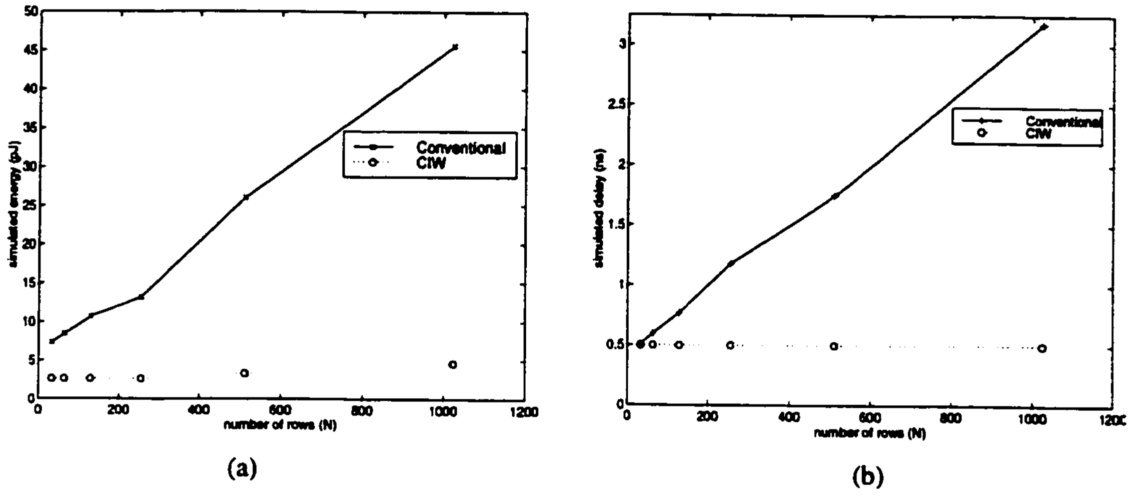


Figure 4.5: Comparisons to conventional write: (a) energy and (b) speed.

energy consumed during writing a single cell. The delay in Fig. 4.5b represents the minimum time needed to write the cell. For the proposed scheme, this time includes the equalization time. As the figure illustrates, the CIW approach reduces energy by 2.8 to 9.9 times and improves the speed by a factor of 1.02 to 6.36. The figures illustrates that, for the memory range shown, the delay and energy of the CIW is almost independent of the memory size. Notice that the speed of both schemes are comparable at small N values. However, the numbers in Fig. 4.5b do not include the *recovery time*, defined as the time needed to pull the bit-lines back to high before the next cycle starts. This time is much worse in the conventional case because the bit-lines are almost fully discharged during writes. In the CIW case, however, transistor M_8 is *always* turned on and so it will quickly bring the bit-lines (which already have small swing) to full V_{dd}. Therefore, if this time is accounted for, the CIW scheme will always be much faster than the conventional.

Similar to the HiCapCS approach described in Chapter 3, an attractive spin-off of the CIW scheme will be the reduction of current spikes and supply voltage noise. HSPICE simulations have shown peak current reduction by a factor of 10 on the average. Further more, not charging and discharging the large bit-line capacitance during writes reduces the noise coupling to adjacent bit-lines.

Unlike the DSL approach [71] described earlier, the CIW scheme requires the assertion of only one horizontal line (the *WWL*). However, this comes at the expense of adding one extra transistor (M_5) and two extra bit-lines (*WBL* and $\overline{WBL}$). Another disadvantage of this technique, which is also common to the DSL, is that all cells in the same row have to be written since they will loose their stored values during the pre-charge phase. Thus the CIW is most suitable to multi-port buffer memories with large word size as in register files, memory based ATM switches, and search CAMs. Extension to a general purpose (that column decoded) SRAM is possible by using multiple write wordlines, which is analogous to multiple source lines in [71].

4.4.3 Cell Design

In addition to the stability requirements described in section 4.4.1, the sizing of the new memory cell is affected by other factors. Referring again to Fig. 4.3, when M_5 is turned on, the equalized voltage on C and $\overline{C}$, call it V_M, will depend on the sizing of M_1 and M_3 (M_2 and M_4). It is desirable to have $V_M = V_{dd}/2$ to bias the CMOS inverters of the memory cell in the mid-point of their transient region and so increase their voltage gain. When M_5 is on, both M_1 and M_3 operate in the saturation region and have equal currents as given in EQ (4.1):

$$\mu_p C_{ox}\left(\frac{W}{L}\right)_{M_1}(V_{dd}-V_M-\Delta V_1-|V_{thp}|)^2 = \mu_n C_{ox}\left(\frac{W}{L}\right)_{M_3}(V_M-V_{thn})^2 \qquad (4.1)$$

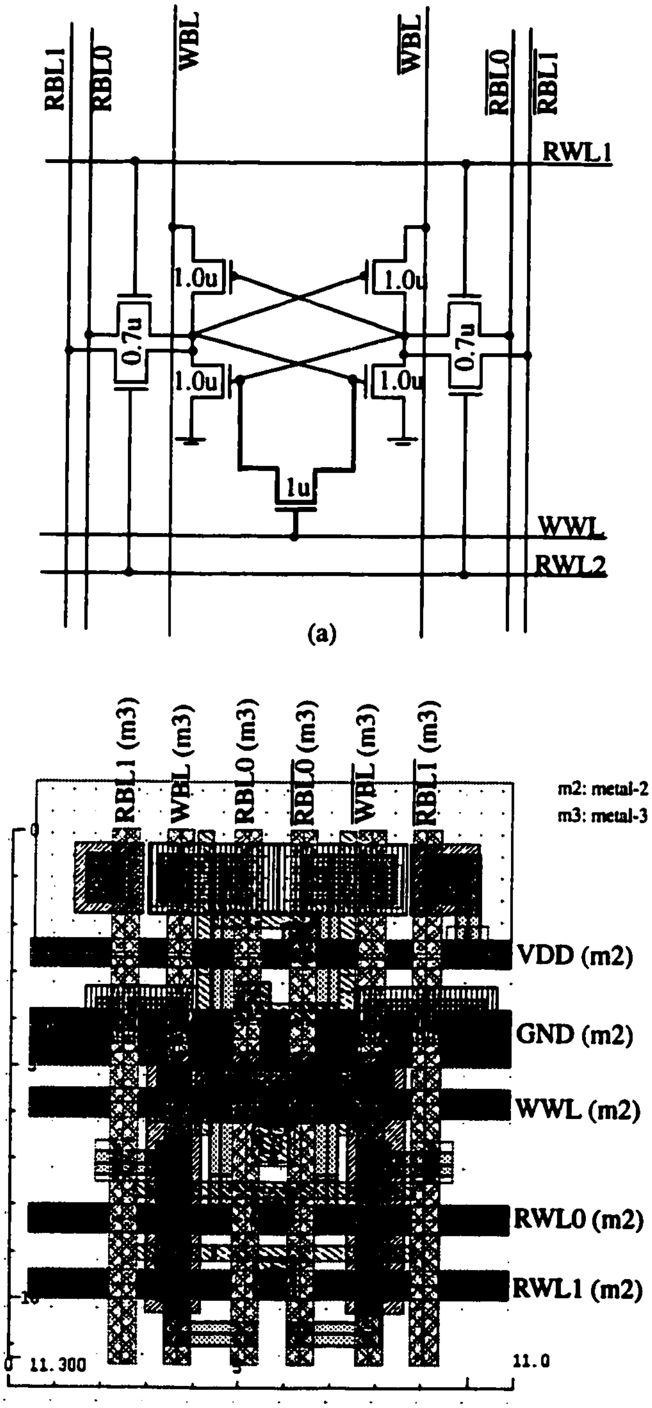
solving the above equation for $V_M = V_{dd}/2$, assuming $V_{thn} = V_{thp}$ and equal transistor length L, gives $W_{M_1}/W_{M_3} \approx \mu_n/\mu_p$. Since $\mu_n \approx (2\text{-}3)\mu_p$ in a typical CMOS process, then W_{M1} should be about (2-3) W_{M3} which will enlarge the cell area. Fortunately, the write operation is fast enough (as was shown in the previous section) and so this condition can be relaxed by making W_{M1} equal to W_{M3}. As a result, V_M will be given by:

$$V_M = \frac{V_{dd} - \Delta V_1 + V_{th}\left(\sqrt{\mu_n/\mu_p} - 1\right)}{\sqrt{\mu_n/\mu_p} + 1} \qquad (4.2)$$

which is about $V_{dd}/2.5$ using typical technology parameters.

Another requirement is that transistor M_5 has to be stronger than M_1 (M_2) in order to be able to pull the high node of the memory cell down to V_M. This can be also achieved by having $W_{M5} = W_{M1}$ because of the difference in mobility as stated above. Finally, similar to conventional SRAM cell design, the size of transistor M_a is made smaller than M_3 (M_4) for noise margin requirement during reads as explained earlier.

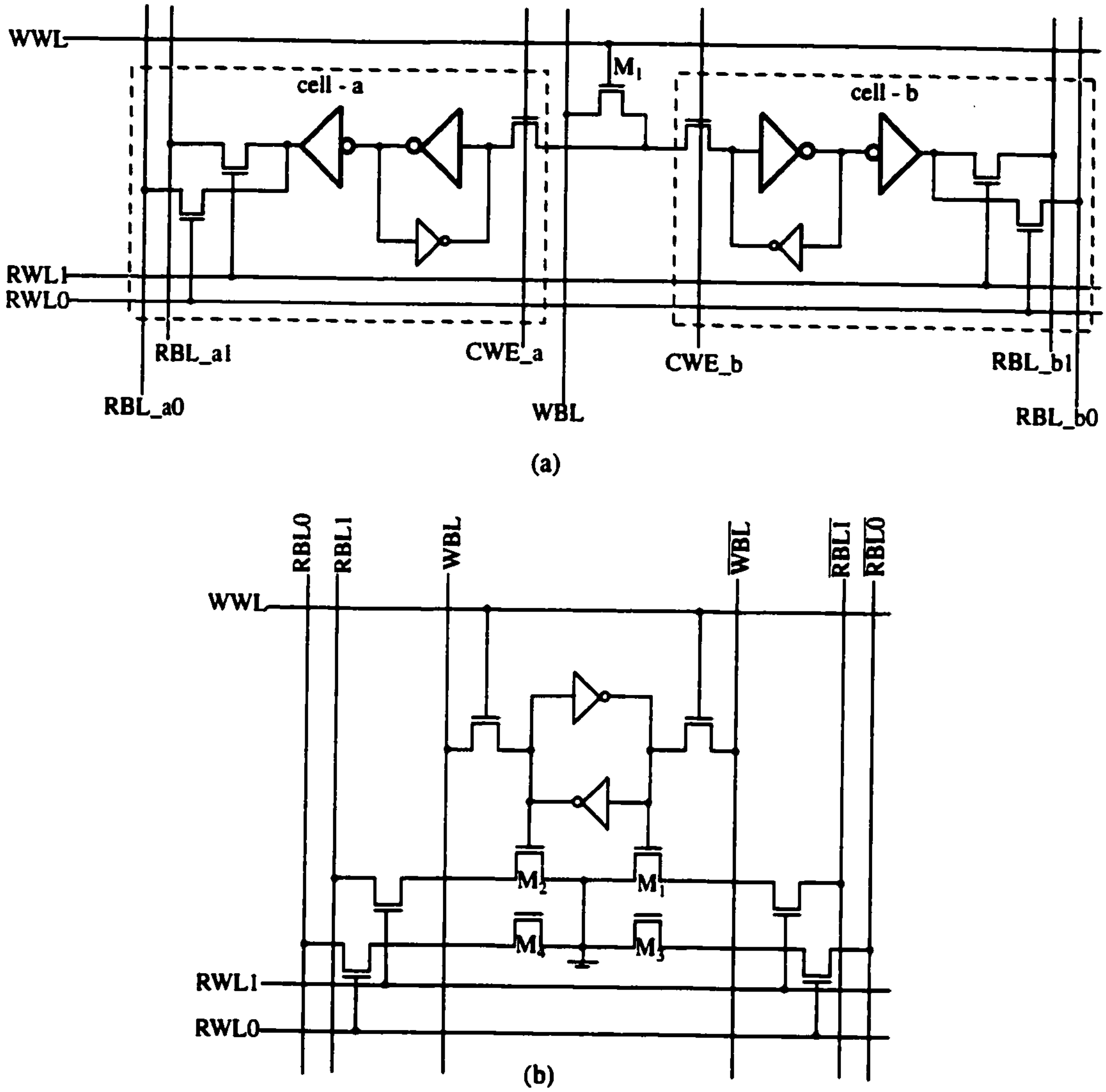
Fig. 4.6a gives the schematic of a 3-port (1W-2R) of a 9T memory cell using the proposed approach. The transistor sizes shown are based on a 3-metal 0.35 μm technology. The layout of the memory cell is shown in Fig 4.6b. The total cell area is about 125 μm^2. The size of the cell is expected to be smaller than conventional multi-port SRAM cells. Two of these cells are depicted in Fig 4.7. The cell in Fig 4.7a is based on a single-ended read and write techniques. As a result, both operations are slow and require full swing on the bit-lines. In this scheme, two adjacent cells *a* and *b* shares a common write bit-line thus the total number of transistors per cell is 9.5, which is comparable to the CIW scheme. However, some of these transistors (indicated by **bold**) have large sizes as required by the single-ended read operation. The cell in Fig. 4.7b is based on a differential read operation and so is fast. The write operation is also differential but follows the conventional full swing scheme. Transistors M_1-M_4 are used to isolate the cell latch from the noise on the read bit-lines to enhance reliability.



(a)

(b)

Figure 4.6: (a) Schematic of the proposed 3-port cell, (b) its layout



(a)

(b)

Figure 4.7: Some conventional 3-port SRAM cells: (a) single-ended [75], (b) double-ended [63].

4.5 Current-Injection Read (CIR) Circuit

Similar to the write operation, the current-injection technique can be applied to the read operation. A CIR sense amplifier is shown in Fig. 4.8. The 7T cell in the figure is similar to the one given in Fig. 4.3 with the write bit-lines not shown for simplicity. As in the CIW technique, read proceeds in two stages: equalize and evaluate. When the sense-amplifier enable (*SAE*) signal is high (equalize phase), M_5 is on and so the voltages on nodes A and $\overline{A}$ are equalized. During this time, RWL is asserted thus turning on the access transistors (M_a in Fig. 4.3) of the memory cell. Because of the opposite polarities on the memory cell inverters, one inverter will sink additional load current, I_d, as shown in the figure. During this phase, the bit-lines current I_{L1} and I_{L2} are equalized. When $SAE = 0$ (evaluate phase), M_5 is turned off and due to the small voltage difference on the bit-lines, M_2 will sink less current than M_1 thus forcing $\overline{A}$ to low and A to high (for the example given in the figure). The amplifier shuts the DC power once its outputs reaches complementary full swings. The non-inverting active-low single-phase dynamic latch

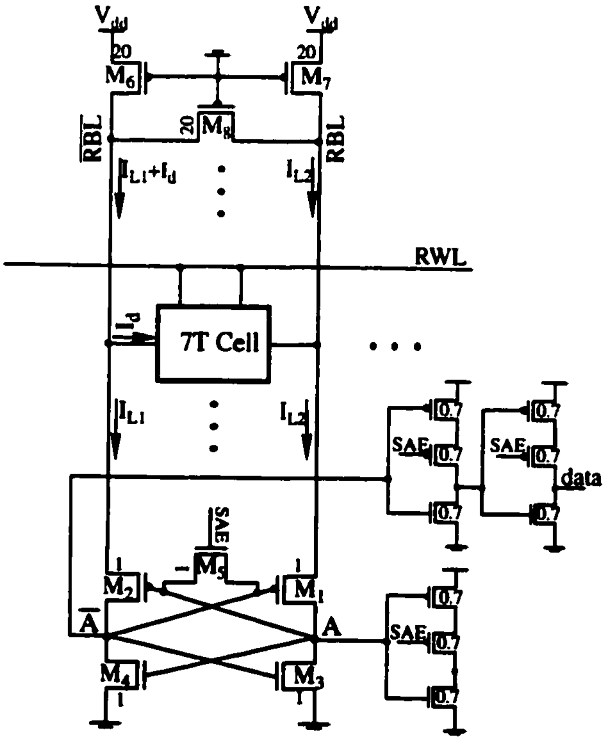


Figure 4.8: Current injection read (CIR).

will hold the newly read value before *SAE* goes back to high. The read cycle ends when *RWL* goes back to low. Notice that for correct read operation, the *RWL* has to be reset slightly after the *SAE* signal is deactivated. This is comparable to the timing requirement between the *WE* and the *WWL* signals in the write cycle.

Sense amplifiers based on current operation have been used before for fast read operation. The amplifier shown in Fig. 4.8 can be though of as combination of those in [76] and [77]. It uses a pair of back-to-back inverters as in [76] and a load equalizing transistor (M_8) as in [77]. Unlike the amplifier in [76], the CIR amplifier does not clamp the bit-lines to a low-voltage level. As explained earlier, this improves the stability of the cell during reads. In addition, the proposed scheme simplifies the read operation since a low swing *RWL* is required in [76] in order not to disturb the stored cell values. As compared to [77], the CIR amplifier does not need a bias voltage generator.

Simulations of the CIR circuit have been carried out for various memory sizes. For a memory with 32 rows, read operation was simulated at a rate of 1 GHz (without taking decoding delay into consideration). This is somewhat slower than the write speed for the same memory size (1.6 GHz) due to the limited speed of the dynamic latch.

4.6 Register File

In order to evaluate the performance of the above techniques, a three-port (2 reads and 1 writes) 32x64b register file was designed and fabricated using a 0.35 μm CMOS technology. This file can be used for applications with continuous read and write requests. In order to operate at high frequencies, two techniques are used: (1) pipelining, and (2) using a single-phase clock for control. The first allows the register file to operate at clock cycles shorter than the access time, while the second eliminates the race problems associated with multiple clock signals. In addition to the above, fast address decoding using Source-Coupled Logic [66] was adopted.

The timing diagram of the pipelined operation is shown in Fig. 4.9. This scheme allows one (or two) read operations and one write (to a different word) every clock cycle as follows. A new address is entered and is allowed to settle during the low-phase of *CLK*. Therefore, address decoding is performed when *CLK* becomes high. When this happens, the decoder will activate the *WWL* of the selected word (write precharge phase). In addition, new data is inputted to the write bit-lines (through transistor M_9 or M_{10} in Fig. 4.3). At the same time, read decoding is activated as well and so the *RWL*(s) of the (other) selected word(s) will be enabled. When *CLK* goes back to low, the *WWL* is disabled and writing is performed during the dashed period shown in Fig. 4.9. During this period, it is important that the new data is stable and signal *WE* remains high.

Similarly for the ongoing read operation, the *RWL* is deactivated and the read sense amplifier proceeds to the evaluate mode. Since the *SAE* signal in Fig. 4.8 is active high, we can substitute this signal with *CLK* to turn off transistor M_5 (and turn on the dynamic latch). Hence only two control signals are needed in this file: *CLK* and *WE*. As for the write operation, reading is performed during the dashed period shown in Fig 4.9. The *RWL* should stay high enough after *CLK* becomes low in order to allow the sense

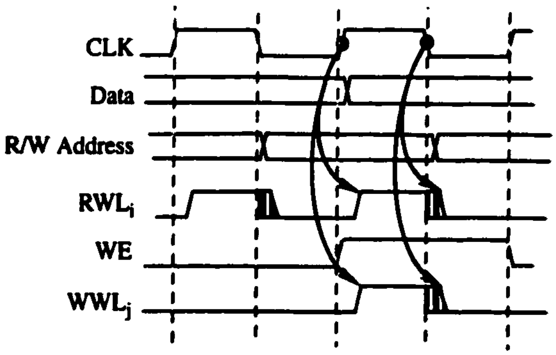


Figure 4.9: Pipelined operation using a single clock.

amplifier to read the correct data. Enough delay between the two signals can be introduced by the decoder circuitry.

In order to operate at high-frequencies, fast address decoding is needed. Source Coupled Logic (SCL) is one of the fastest address decoding schemes ever reported [66]. The SCL logic is shown in Fig. 4.10. When *CLK* is low, nodes *A* and *B* are charged to V_{dd}. When *CLK* is high and depending on the address lines, node *A* will either remain high or get discharged. In the first case (word selected), node *B* will get discharged through M_2 and M_3 and the word line will rise. In the later case, the word line remains low. The weak cross-coupled PMOS feedback transistors enhance the noise immunity of this dynamic circuit by keeping the appropriate charge after *CLK* becomes high. The operation of the SCL fits the behavior of both the *RWL* and the *WWL* as described before. Note that the sizing of the driving buffer is slightly skewed to delay *RWL* from *CLK*, as explained above.

In the above scenario, we assumed a high-frequency single-phase clock signal with 50% duty cycle. At high-frequencies (for example ≥ 500MHz), the width of the high level portion of the clock is very short, thus minimizing the DC current during both reads and writes. For the general case, a pulsed operation can be used in which the *RWL*, *WWL*, and the *SAE* lines are all pulsed with a single positive *CLK* pulse. This pulse can be generated from Address-Transition Detection (ATD) circuitry as in some low-power SRAMs [59].

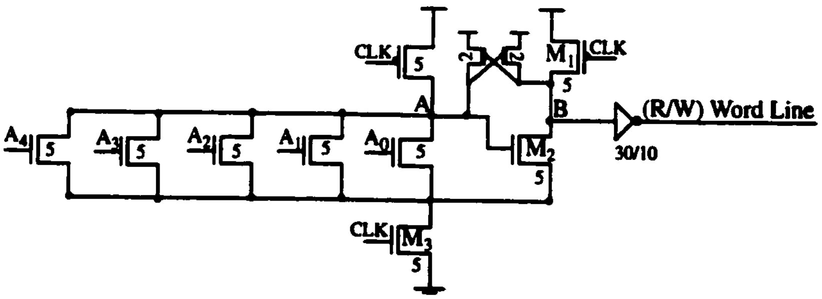


Figure 4.10: Source-Coupled Logic (SCL) for read/write word line decoding.

Using a 0.35 μm CMOS technology with $V_{dd} = 3.3$, and $f = 750$ MHz, the register file (including decoder circuitry) was simulated with HSPICE. Extracted parasitic capacitance and resistance from the actual layout were included. Results from the simulation are given in Fig. 4.11. The simulations show one write followed by two consecutive reads of the same memory cell. The cell potentials are given by C and $\overline{C}$ in the figure. The intermediate voltage V_M is about 1.2V. Note that the output read data follows the written input.

Since both read and write operations are based on current-injection techniques, this register file can still operate at very high frequencies even under low supply voltages. Fig. 4.12 plots the simulated maximum operating frequency for a range of supply

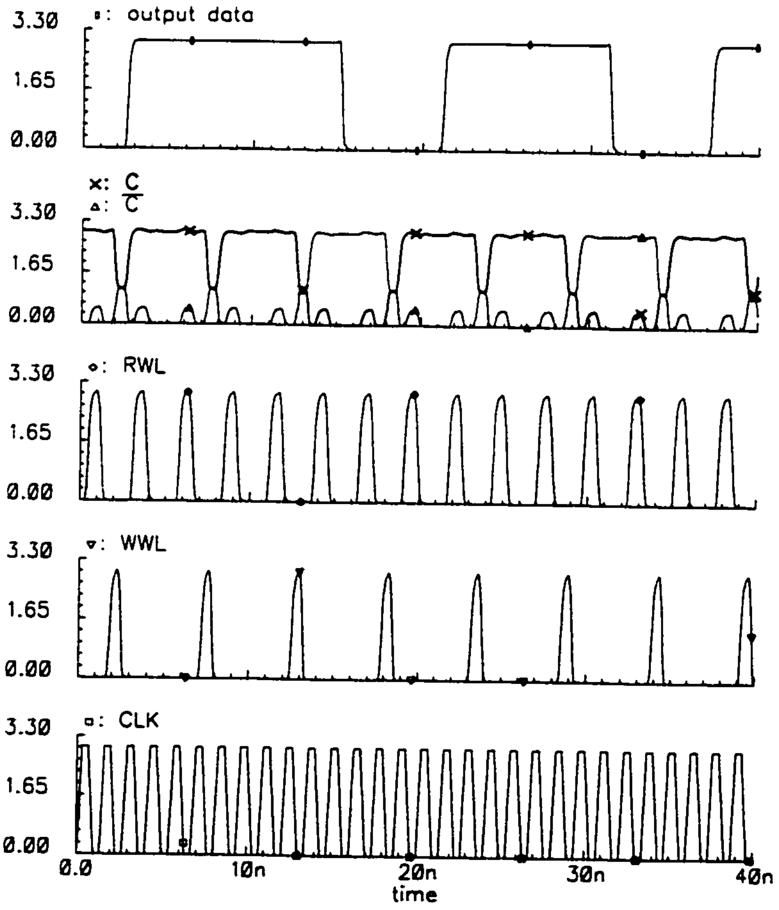


Figure 4.11: HSPICE simulations of the 3-port 32x64 register file.

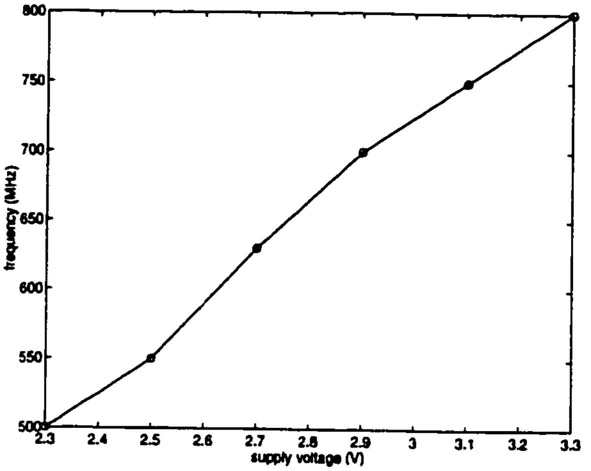


Figure 4.12: Simulated results: effect of supply voltage scaling on speed.

voltages. At a low voltage of 2.3V, the maximum frequency is 500MHz. The minimum supply voltage under which the file can function correctly is determined by the equalizing transistor (M_5 in Fig. 4.3 and Fig. 4.8). This transistor can equalize its source and drain nodes as long as its gate voltage is larger than twice its threshold voltage, where the later is subject to body effect. To work at even lower voltages, low-threshold M_5 can be used if advanced multi-V_{th} process is available. Another alternative is to boost the gate voltage of M_5 and use a low V_{dd} for the rest of the circuit.

4.7 Experimental Results

The 32x64b 3-port register file was fabricated using a 0.35 µm CMOS technology. The layout of the complete chip is shown in Fig. 4.13. The total core area is 604µm x 900µm. The figure clearly shows the memory array and three decoder sections. The bitline loads and the write drivers were placed on top of the array, while the sense amplifiers and output drivers were located at the bottom.

Measurements were carried out using an IMS-100 Logic Tester for input pattern generation and an oscilloscope for waveform acquisition. Functionality was verified by changing the write address from 0 to 7 and writing different data to the corresponding words. *Simultaneous* reading of words 1,2,...,7, 0 and 2,3, ...,0,1 was performed and the read data matched the written one. Fig 4.14 shows some of the measured waveforms. In Fig 4.14a, the waveforms of the *CLK* signal, one address line, one input line and one output line are given. Notice that the address changes when *CLK* is low, while the input data varies when *CLK* is high, as explained in the previous section. The input pattern used was 0010011101001100 and repeat. The figure illustrates that the output data matches the input. The same output is depicted in Fig. 4.14b but using a supply voltage of 2.3V. Finally, Fig. 11c shows the same output at frequency of 100 MHz and V_{dd} =

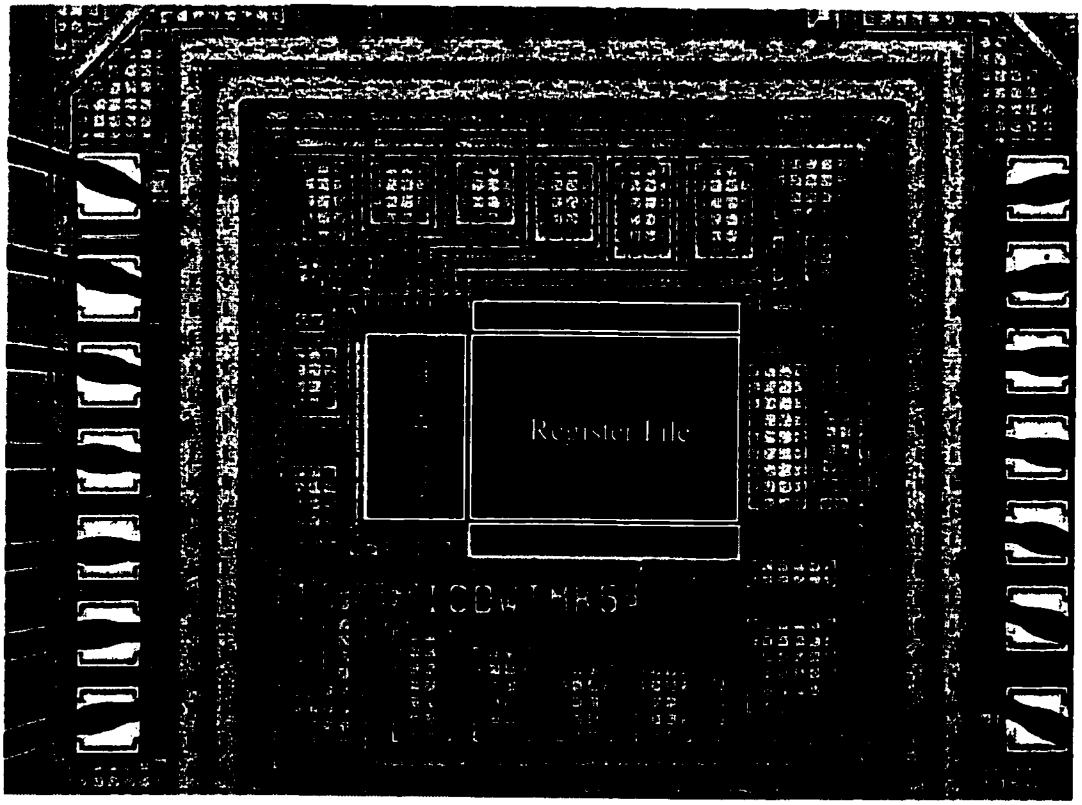
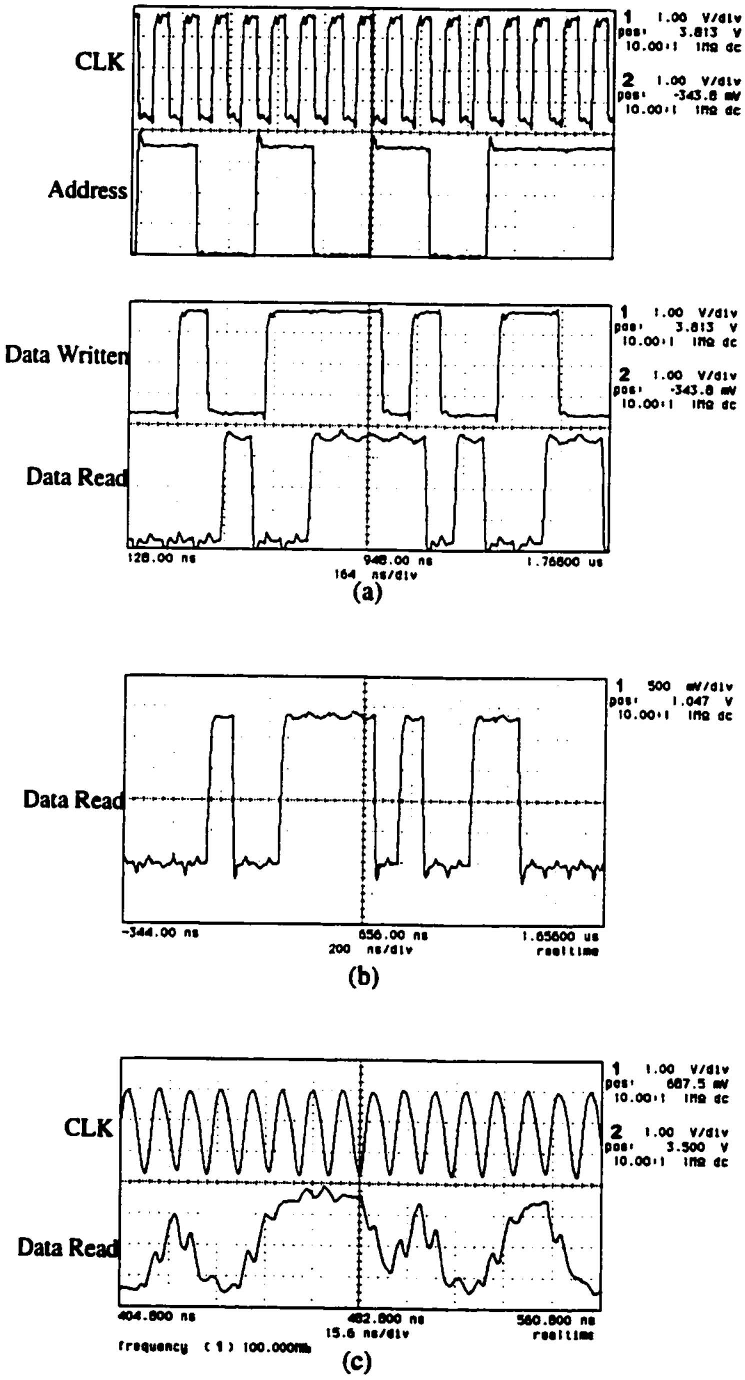


Figure 4.13: photo micrograph of the 32x64 3-port register file test chip.



(a)

(b)

(c)

Figure 4.14: Measured waveforms: (a) At V_{dd}=3.3V, (b) At V_{dd} = 2.3V, and (c) At V_{dd} = 3.3V and f = 100MHz.

3.3V. The distortion of the output can be partially attributed to the noise introduced by the available low-frequency package.

Fig. 4.15 plot the effect of supply voltage scaling on the measured delay. The delay shown is taken from the falling edge of the *CLK* signal to the time the output data becomes available. This delay includes that of the I/O pads and so taking that into consideration, the measured results in Fig. 4.15 are in good agreement with the simulated ones in Fig. 4.12. With respect to power dissipation, measured power is not available as both the I/O pads and the chip core use common supply pads.

4.8 Summary

We described novel circuit techniques based on current-injection to enable fast and low-power write and read operations in multi-port SRAMs. The design of a pipelined 32x64 3-port register file that demonstrates the use of the proposed techniques was also presented. Measurement results from the fabricated register file chip have confirmed the feasibility of the approach.

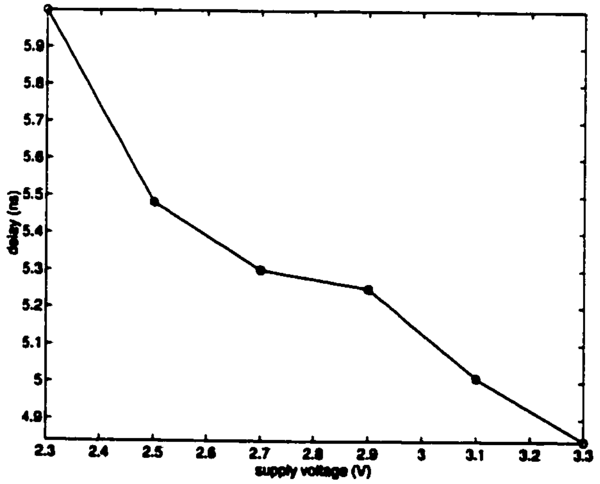


Figure 4.15: Measured results: effect of varying V_{dd} on the delay.

Chapter 5

Power Minimization Using Dual-V_{dd} and Dual-V_{th} (DVDV)

5.1 Introduction

In a well-designed CMOS circuit, power consumption is predominated by the dynamic (charging) power which is relative to the square of the supply voltage (V_{dd}). Scaling down V_{dd} is, therefore, one of the most effective means to cut down the power. Lowering V_{dd} is also a requirement in submicron technologies to prevent hot-carrier effects.

However, reducing V_{dd} alone comes at the expense of increasing the delay because the current driving capability of a MOS transistor decreases. Reducing V_{dd} and (relatively so) the threshold voltage (V_{th}) can maintain improvement in circuit performance. Lowering V_{th}, on the other hand, results in exponential increase in the subthreshold (leakage) power. At very low V_{th} and with the current trend in system-on-a-chip, where millions of transistors are integrated on a single die, the leakage power can actually dominate.

In this work, the Dual-V_{dd} Dual-V_{th} (DVDV) methodology is proposed as a means for *low-power* design of *high-performance* CMOS logic circuits. DVDV employs two supply voltages HV_{dd} and LV_{dd}, and two threshold voltages HV_{th} and LV_{th}, where H and L stands for high and low, respectively. With this technique, it is possible to operate a logic circuit at very high speeds because of the availability of HV_{dd} gates. These gates are mapped to nodes on the critical path(s) of the circuit. On the other hand, nodes off the critical path are made to operate at LV_{dd} for dynamic power saving. Depending on the synthesized circuit and the required speed, the low supply voltage (LV_{dd}) can be made very small (as compared to HV_{dd}) thanks to the availability of the low threshold voltage (LV_{th}). Finally, leakage power is substantially reduced by using HV_{th} gates whenever possible.

After this brief introduction, the delay and power d models for a (deep) submicron CMOS gate are presented. Section 5.3 discusses the merits and shortcomings of supply voltage scaling followed by a review of the state-of-the-art methods targeting the low-supply/low-threshold voltage paradigm. The DVDV technique is presented in Section 5.5. Following that, a Depth-First Search (DFS) based heuristic for minimizing the total power dissipation of a logic circuit under a delay constraint and using the DVDV methodology is described. Section 5.7 provides results from applying our new scheme on a number of ISCAS and MCNC benchmark circuits. A summary of the chapter is given in Section 5.8.

5.2 Delay and Power Models

5.2.1 Delay Model

The basic CMOS delay model (presented in Chapter 2) where the delay of a CMOS gate is inversely propotional to $(V_{dd} - V_{th})^2$ is not valid for short (sub micro-meter) channel devices. The main reason for this deviation is the *velocity saturation* [60]. This term refers to the fact that by shortening the transistor channel, the carriers velocity will saturate when the electrical field along the channel reaches a critical value, E_{sat}. For a *p*-type silicon, electrons reach saturation at an electrical field of about 1.5V/µm. In a 1 µm NMOS device, this is translated to a drain-source voltage (V_{ds}) of 1.5V after which velocity saturation occurs. In a 0.25µm process, this voltage reduces to 0.375V which can be easily met.

Several models were proposed in the past years to account for velocity saturation and other short-channel effects such as carrier mobility degradation. An often cited model is the alpha-power law [78] which replaces the square factor above by $\bar{\alpha}$ that can take any value between one, for full velocity saturation, and two, for no velocity saturation, as in long channel devices. An average value of $\bar{\alpha}$ between 1.3 to 1.5 is often used [58] [79] [80]. Thus the delay equation of a CMOS gate can be modified to:

$$T_d \approx \frac{C_L \cdot V_{dd}}{K \cdot \left(\frac{W}{L}\right)(V_{dd} - V_{th})^{\bar{\alpha}}} \qquad (5.1)$$

where $K = \mu C_{ox}$ and $C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}}$ is the gate oxide capacitance per unit area with t_{ox} being the oxide thickness. The mobility factor µ is not constant and depends on the applied electric field among other factors. Finally, *W* and *L* are the width and length of the transistor, respectively.

Notice that for short-channel devices and since $\bar{\alpha}$ is less than two, reducing the supply voltage (up to a certain extent) does not have a very significant impact on the delay as in long-channel transistors. For example, moving from 5V supply to 3.3V results between 10% to 30% degradation in the delay (depending on the threshold voltage) and yields a 60% reduction in power.

5.2.2 Power Model

As discussed in Chapter 2, the total power dissipation (ignoring short-circuit power) of a CMOS circuit with *N* gates is given by:

$$P_{total} \approx P_{dyn} + P_{leakage} = \left(V_{dd}^2 \cdot f \cdot A\right) \cdot \sum_i^N C_L^i + \sum_i^N \left(I_{leak}^i \cdot V_{dd}\right) \tag{5.2}$$

Where $C_L^i = \alpha_i C_i$ is the switched capacitance of gate *i* with α_i being the switching activity (as defined in Chapter 2) and C_i is the total lumped capacitance seen at the output of the gate.The factor *A* defines the fraction of time the circuit is active (i.e. doing useful computations) over an extended period of time, which can be possibly obtained from system-level simulations.

The second term of EQ 5.2 considers the total leakage power, with I_{leak}^i being the sub-threshold current of gate *i*. This current accounts for the weak-inversion conduction of the carriers when the gate to source voltage (V_{gs}) of the transistor drops below the threshold voltage (V_{th}), and is given by [81]:

$$I_{leakage} = I_0\left[1 - e^{-V_{ds}/V_T}\right]e^{(V_{gs} - V_{th})/(nV_T)} \tag{5.3}$$

where $I_0 = \mu_0 C_{ox}(W_{eff}/L_{eff})\, V_T^2 e^{1.8}$, μ_0 is the zero bias mobility, $V_T = kT/q$ is the thermal voltage which is about 26mV at T = 300K, and n is the subthreshold swing coefficient given by $1 + \frac{C_d}{C_{ox}}$ with C_d being the depletion layer capacitance of the source/drain junctions.

Note that the leakage current is *exponentially* propotional to (V_{gs}-V_{th}), as seen in Fig. 5.1. The figure plots I_{ds} vs. V_{gs} for a 0.25μm process for two V_{th} values. As V_{th} is lowered, the leakage current exponentially increases, thus contributing a significant amount to the total power dissipation in CMOS circuits. This is particularly true if the circuit has to operate at low frequency (*f*) and/or under low activity (*A*), as given in EQ

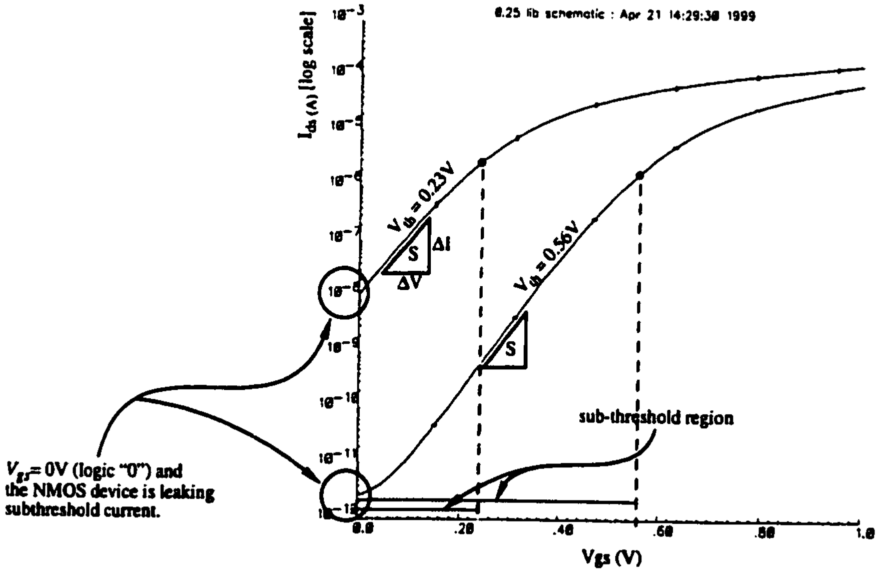


Figure 5.1: Sub-threshold current in a 0.25μm NMOS (W = 1μm, V_{ds} = 1V).

5.2. Under these circumstances, the leakage power can overtop the total power. This issue will be further tackled in the next section.

The sub-threshold behavior of the MOSFET transistor is also often characterized by the inverse subthreshold slope (S) defined as the amount of gate voltage required to drop the leakage current by one decade (see Fig. 5.1). This amount can be determined by taking the ratio of two points in the sub-threshold region using EQ 5.3 which gives $S = nV_T\ln(10)$ or $nV_T = S/\ln(10)$. Using this result and V_{gs}= 0V (device off), EQ 5.3 can be rewritten as:

$$I_{leakage} = I_0\left[1 - e^{-V_{ds}/V_T}\right]10^{-V_{th}/S} \qquad (5.4)$$

It is desirable to have S as small as possible in order to turn off the device as close to V_{th} as possible [6]. Unfortunately, S increases with temperature rise and has a minimum value (for a subthreshold swing coefficient n of unity) of $2.3V_T$ which is about 60mV/decade at room temperature. In reality, n is greater than one and is largely invariant with scaling [9]. This makes a typical value of S in the proximity of 100mV which means that for every 100mV reduction in V_{th}, the leakage current is increased by one order of magnitude.

5.3 Supply Voltage Scaling

Supply voltage scaling is a key methodology for low-power design and is believed to yield the maximum reduction in power among other approaches [3]. Another force behind low-voltage design is the high demand for portable products which are powered by batteries with limited supply voltage. Moreover, reducing the supply voltage is also a must in high-performance and Ultra Large Scale Integration (ULSI) systems to cut down cooling costs and maintain manageable power consumption per chip area.

From a reliability viewpoint, lowering the supply voltage is essential to prevent long term device degradation and eventually malfunction due to high electric fields. As miniaturization of CMOS devices progresses without reducing the supply voltage, the electric field (measured in V/m) along the channel increases and this can trigger hot-carrier mechanisms. By increasing the electric field, the electrons can acquire very high velocity (or become hot) and so may depart the silicon and tunnel into the gate oxide. This can result in modifying the device properties (for example, its threshold voltage) and finally device failure.

Unfortunately, lowering the supply voltage comes at the expense of increasing the delay. In [9], two voltage scaling scenarios are described: one optimized for the highest speed without impacting the long term device reliability and one optimized for power. In both cases, the threshold voltage is not fully scaled to maintain acceptable leakage current. It was found that for the *same process generation* (feature length), the high-performance scenario requires higher supply voltage than the low-power one by a factor of about 1.5 on average. For example, the high-performance supply voltage for a 0.25μm process is about 2.5V while the low-power one is set to 1.5V.

In this section, we present a simple explanation as to why the above two scenarios differ. Assume that we have a circuit designed in a certain process and is operating at a high supply voltage (HV_{dd}) and under a high threshold voltage (HV_{th}). Assume further that we want to scale down both the supply and threshold voltages to LV_{dd} and LV_{th}, respectively such as the delay (speed) is maintained while keeping the sizes of transistors (or area) the same. If EQ 5.1 is used before and after scaling, then the following equality has to be satisfied:

$$\frac{C_L \cdot HV_{dd}}{K\left(\frac{W}{L}\right)(HV_{dd}-HV_{th})^{\bar{\alpha}}} = \frac{C_L \cdot LV_{dd}}{K\left(\frac{W}{L}\right)(LV_{dd}-LV_{th})^{\bar{\alpha}}} \quad (5.5)$$

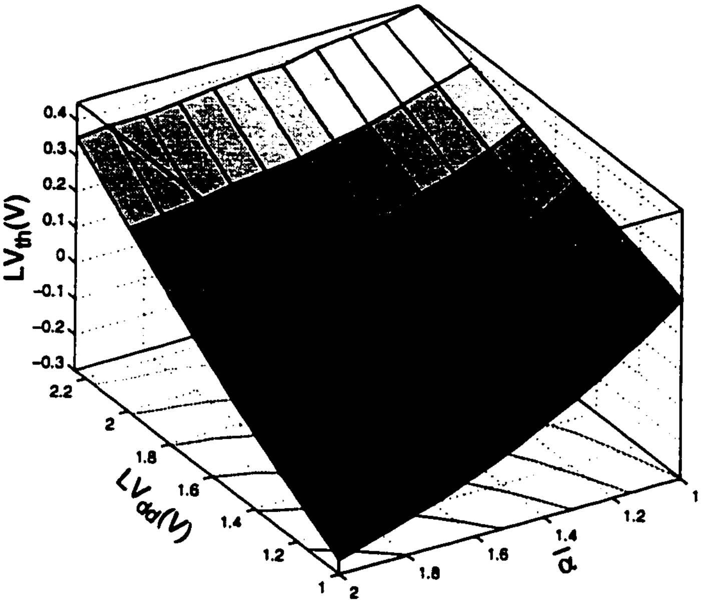


Figure 5.2: Effect of velocity-saturation factor ($\bar{\alpha}$) and LV_{dd} on LV_{th}.

Solving for LV_{th} gives:

$$LV_{th} = LV_{dd} - (HV_{dd} - HV_{th})\left(\frac{LV_{dd}}{HV_{dd}}\right)^{1/\bar{\alpha}} \quad (5.6)$$

EQ 5.6 tells us that LV_{th} has to be reduced as LV_{dd} decreases or $\bar{\alpha}$ increases as illustrated in Fig. 5.2. The figure plots LV_{th} as a function of LV_{dd} and $\bar{\alpha}$, assuming a HV_{dd} = 2.5V and HV_{th} = 0.5V. The contour curves show constant LV_{th} versus LV_{dd} and $\bar{\alpha}$. Notice that at $\bar{\alpha}$ = 1 (full velocity saturation), $LV_{th} = LV_{dd}/5$. However, in contemporary submicron processes and down to 0.1μm technologies [79] and possibly less, $\bar{\alpha}$ is unlikely to be this small. Therefore, much lower LV_{th} is needed at low LV_{dd} (low-power scenario in [9]) to maintain the same delay, as the figure illustrates. For example, for $\bar{\alpha}$ = 1.3-1.5 and at a low supply voltage of 1V, LV_{th} drops below zero (for an NMOS device).

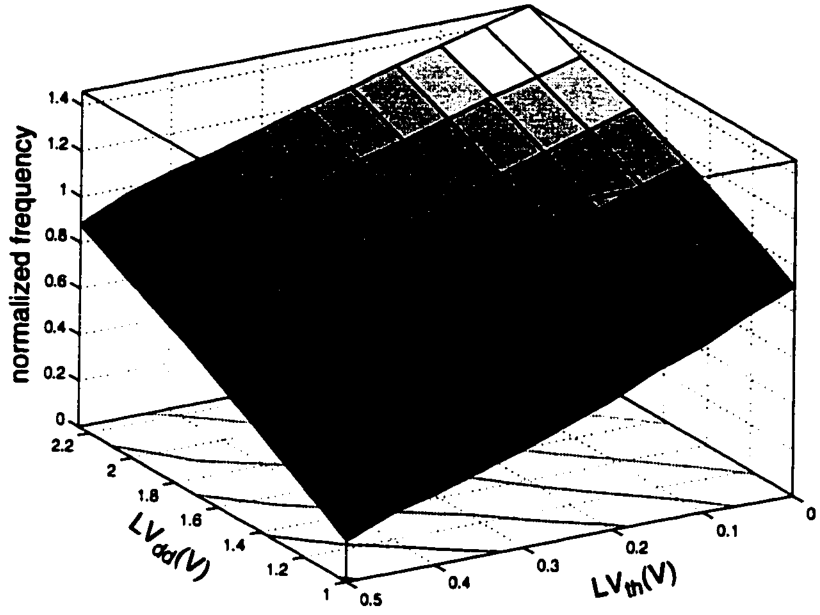


Figure 5.3: Oscillation frequency of a 31-stage ring oscillator in a 0.25μm process.

To verify the above analytical result, a benchmark circuit representing a ring oscillator of 31 stages was simulated using HSPICE with a 0.25 μm process. We varied LV_{th} by modifying the *DVTH* parameter in the HSPICE models. Fig. 5.3 shows the oscillation frequency versus LV_{dd} and LV_{th}, normalized with respect to that using HV_{dd} = 2.5V, and HV_{th} =0.56V. Note that at LV_{dd} = 1V, and even with LV_{th} = 0V, the speed is only about 90% of the original. Solving for $\bar{\alpha}$ using the above simulation points gives a value of 1.54, as expected. Notice that the results in Fig 5.3 is in good agreement with those given by EQ 5.6. As an example, for LV_{dd} = 1.5V and LV_{th} = 0.1V, the oscillation frequency is almost equal to the original (within 4%) as shown in Fig 5.3. Using EQ 5.6 with $\bar{\alpha}$ = 1.54, HV_{dd} = 2.5V, and LV_{th} = 0.56, gives a LV_{th} of about 0.108V which is close to the simulated one (0.1V).

Another example that illustrates the effect of voltage scaling on delay is given in Fig. 5.4. The figure (taken from [9]) plots the delay of a 2-input NAND gate for different

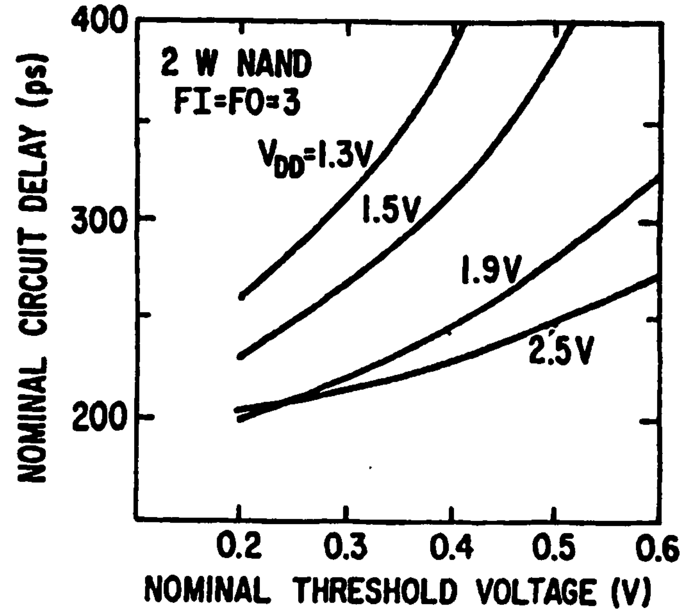


Figure 5.4: 0.25μm CMOS delay versus V_{th} and $V_{dd.}$ [9].

V_{th} and V_{dd} values. At a V_{dd} = 1.3V and V_{th} = 0.26, the delay is about 18% worse than at V_{dd} = 2.5 and V_{th} = 0.5, although V_{dd}/V_{th} = 5 in both cases. The figure predicts that the delay gets even worse as V_{dd} is reduced.

The proceeding discussion suggests that it is theoretically possible to achieve the same performance before and after V_{dd} scaling if V_{th} is scaled proportionally as dictated by EQ. 5.5. However, two main factors prevent full V_{dd} scaling. The first is the increase in the leakage current. Remember from before (section 5.2.2) that as V_{th} decreases, a hefty penalty in the leakage current has to be paid since the later is exponentially propotional to V_{th}. As an example, Fig 5.5 plots the ratio of the leakage power, simulated using HSPICE, to the total power (EQ. 5.2) as a function of LV_{dd} and LV_{th}. This was done for the above ring oscillator benchmark (without the ring feedback) using a fixed input frequency of 50MHz. It is clear that at low LV_{th}, the leakage power becomes considerable. For example, at LV_{dd} = 1.0 and LV_{th} = 0, the leakage power is more than 50%

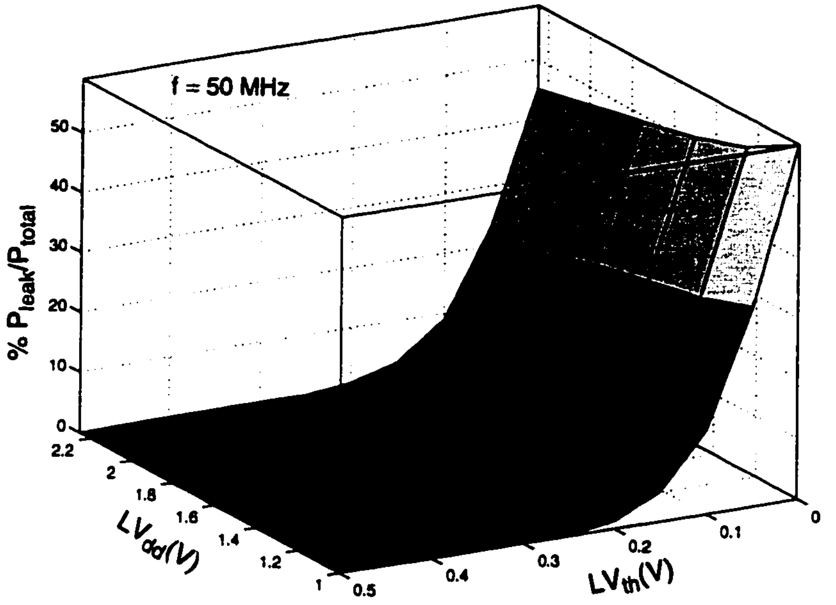


Figure 5.5: % P_{leak}/P_{total} as a function of LV_{th} and LV_{dd}.

of the total power. This contribution is, however, much smaller at higher frequencies at which point the dynamic component of the power becomes more significant.

The second element that prevents extreme V_{dd} scaling is the increased sensitivity of the delay to fluctuations in V_{th} caused by process variations. Note that the threshold voltage of transistors within a chip or a single transistor across chips varies randomly. This can be attributed to nonuniform fabrication conditions in elements like the channel length, impurity concentration densities, oxide thicknesses, and diffusion depths [60]. For achieving a high yield, the chip with the worst case (highest) V_{th} determines the speed specification of a design. Consequently, minimizing V_{th} variations is a key issue that has to be addressed in future ultra high-performance chips.

In [58], it was analytically shown that the delay sensitivity to V_{th} variations worsens as V_{dd} is reduced. Assume that ΔV_{th} and ΔT_d stand for the variations in the threshold voltage and delay, respectively. Then by differentiating the delay given by EQ 5.1 with respect to V_{th}, the following relation is obtained:

$$\frac{\Delta T_d}{\Delta V_{th}} \propto \frac{1}{(V_{dd} - V_{th})^{\bar{\alpha}+1}} \tag{5.7}$$

The above result was derived in [58] for $\bar{\alpha} = 1.5$ and is rewritten above and further investigated next for the general case. Note that as $\bar{\alpha}$ becomes small and approach unity, the exponent of the denominator gets smaller. This means that under this condition (small $\bar{\alpha}$), reducing V_{dd} makes the delay more sensitive to V_{th} variations (large $\Delta T_d/\Delta V_{th}$). This conclusion is true even if V_{th} is scaled relatively for equal T_d as illustrated in Fig. 5.6. The figure plots $\Delta T_d/\Delta V_{th}$ as a function of $\bar{\alpha}$ for three V_{dd} values: 2.5V, 1.5V, and 1.0V. Only the lowest curve (V_{dd} = 2.5V) has a fixed V_{th} of 0.5V, while the other two curves assume different V_{th} for different $\bar{\alpha}$. The exact V_{th} value at each point on the top two curves were obtained from EQ 5.6 for equal delay. It is clear that $\Delta T_d/\Delta V_{th}$ is not a

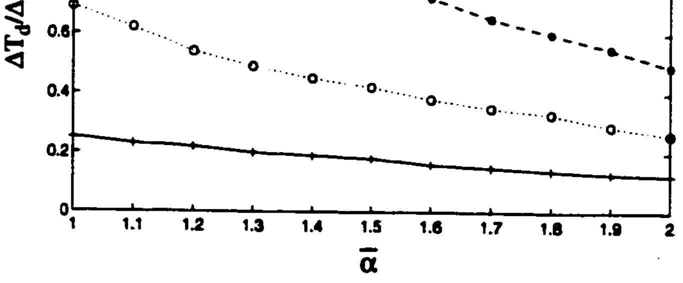


Figure 5.6: Delay variations for different V_{dd}'s and $\bar{\alpha}$'s.

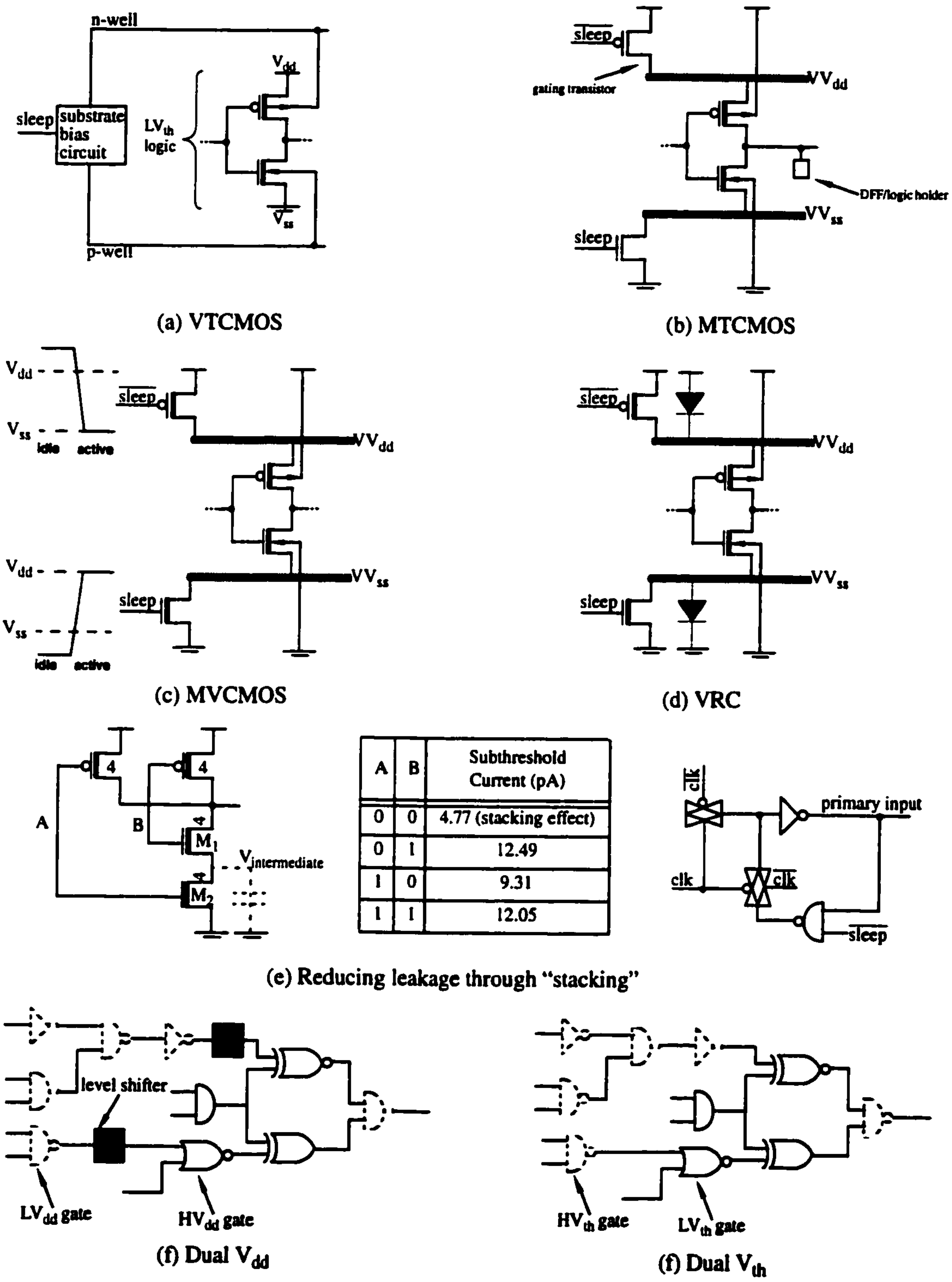
big issue at high V_{dd}. As V_{dd} is lowered, this ratio is adversely affected particularly as devices become more velocity saturated (small $\bar{\alpha}$).

In summary, scaling the supply voltage is essential for reliable device operation and to enable battery-operated systems. However, it was shown that without a relative decrease in the threshold voltage, the delay increases rapidly, particularly at very low supplies. An important relation (EQ. 5.6) that expresses the new low threshold voltage in terms of the new scaled supply, the old high supply and threshold voltages, and the velocity saturation factor $\bar{\alpha}$ was derived. Scaling V_{th} according to that relation for equal delay results in a very high penalty in terms of the leakage current. Furthermore, it was shown that the delay sensitivity to V_{th} variations is exacerbated at low V_{dd}, especially for velocity saturated devices. In the next section, a review of the state-of-the-art design techniques that address the problems associated with V_{dd} scaling will be presented.

5.4 Low-V_{dd} Circuit Techniques

In general, low-V_{dd} circuit approaches can be classified into two categories: *dynamic* and *static*. In the dynamic case, a control signal is commonly employed to turn off devices in the stand-by (sleep) mode to minimize leakage power. This mechanism is similar to disabling the clock feeding various parts of a system during idle periods in order to cut off the dynamic power dissipation of these sections (since $f = 0$) [40].

Parts (a)-(e) of Fig. 5.7 show different dynamic low-V_{dd} circuit techniques that were proposed over the last few years. In the Variable Threshold CMOS (VTCMOS) approach [82], the threshold voltage is controlled dynamically through varying the substrate bias voltage. In this scheme, all transistors have low V_{th} (LV_{th}) and the substrate bias voltage is altered so as to: (1) compensate for V_{th} fluctuations in the active mode



(a) VTCMOS

(b) MTCMOS

(c) MVCMOS

(d) VRC

A	B	Subthreshold Current (pA)
0	0	4.77 (stacking effect)
0	1	12.49
1	0	9.31
1	1	12.05

(e) Reducing leakage through "stacking"

(f) Dual V_{dd}

(f) Dual V_{th}

Figure 5.7: Low V_{dd} circuit techniques: dynamic (a)-(e), static (f)-(g)

and so minimize delay variations, and (2) reduce the leakage current in the stand-by mode. Notice that V_{th} is propotional to the square root of the substrate voltage (EQ. 3.3) and so a large change in the later is required to change V_{th} by effective values. Another problem associated with VTCMOS is that it requires a triple-well structure and a charge-pump circuit. A similar technique to achieve dynamically variable V_{th} but using Silicon On Insulator with Active Substrate (SOIAS) was investigated in [41].

Another commonly used configuration, called the Multi-Threshold CMOS (MTCMOS) [83], is given in Fig. 5.7b. The MTCMOS employs both low and high V_{th} transistors. The logic gates are implemented using LV_{th} transistors and are connected to virtual ground (VV_{ss}) and virtual supply (VV_{dd}) lines. These lines are linked to the main supply and ground rails through high V_{th} (HV_{th}) transistors with large sizes. Using LV_{th} transistors permits operating at low supply value during the active mode. In the stand-by mode, the *sleep* signal is activated to turn off the HV_{th} devices. This will cause the VV_{ss} and the VV_{dd} to float and so limit the leakage current to that of the HV_{th} transistors, which is very small.

To preserve the data during the sleep mode, a conventional static latch implemented using HV_{th} can be used. Other more faster latch circuitry like the "balloon" scheme was also proposed [84]. Note that the configuration shown in Fig. 5.7b is particularly useful if a latch has to be used at the output. In this case, both HV_{th} PMOS and NMOS are necessary to prevent any leakage through the latch. In pure combinational circuits where storing the state is not required, only one HV_{th} transistor (either a PMOS or an NMOS) can be used.

There are several drawbacks associated with the MTCMOS scheme. First, both LV_{th} and HV_{th} devices have to be available. Second, the HV_{th} gating devices can eventually limit how low the supply voltage can be scaled. Third, it requires extra overhead to store the state in idle periods. Finally, the gating devices have to be made sufficiently large to (1) reduce delay, and (2) minimize noise bouncing on the virtual rails. Note that the virtual lines have much higher impedance than the true rails and will unavoidably bounce. This will adversely affect both noise margins and delay. The latter is influenced because of reduced effective supply voltage and increased V_{th} due to body effect. A technique for sizing MTCMOS gating transistors for minimizing delay based on mutual exclusive discharge pattern was proposed in [85].

Fig. 5.7c and Fig 5.7d show two recently proposed configurations that address some of the MTCMOS problems. The Multi-Voltage CMOS (MVCMOS) [86] scheme in Fig 5.7c do not use HV_{th} devices to reduce leakage power. Instead, it employs LV_{th} gating transistors whose gate voltages are driven in the sleep mode to larger than V_{dd} and smaller than V_{ss} for the PMOS and NMOS, respectively. This creates a negative gate-to-source biasing (V_{gs}) and so reduce leakage current substantially (see EQ 5.3). The elimination of the HV_{th} transistors allow the supply voltage to get much smaller than the MTCMOS technique. Also, since the gating transistors are LV_{th} devices, they can be sized smaller than the HV_{th} MTCMOS transistors for the same current drive. This scheme still however suffers from the supply bounce problem and, in addition, it requires positive and/or negative charge pump(s).

The Virtual Rails Clamp (VRC) [87] circuit in Fig 5.7d solves the data holding problem during the stand-by mode. Similar to the MVCMOS, the gating transistors are LV_{th} devices. These transistors are turned off in idle periods and so the virtual rails will float. Due to leakage current, the VV_{dd} line will start dropping below V_{dd} and the VV_{ss} line will start rising above V_{ss}. Once the voltage on the VV_{dd} line drops by an amount equivalent to the built-in potential (Φ_0) of the clamping diode (~0.5V), the diode is turned on, hence preventing any further discharge. Similarly, any voltage increase on the

VV_{ss} line will be compensated by the other clamping diode. The difference between the VV_{dd} and the VV_{ss} rails which is about V_{dd} -2Φ_0 makes it possible to hold data during the sleep mode. In that mode, the leakage current is reduced because the voltage drop/ increase on the virtual rails will (1) cause an increase in the bulk-to-source voltages of internal transistors which in turn increases their threshold voltages (body effect), and (2) negatively bias the gate-to-source voltages of internal transistors. However, unlike the MTCMOS scheme, the VRC approach suffers from leakage current between the true and virtual rails due to the use of LV_{th} gating transistors. In addition, it sets a lower limit on V_{dd} in order to have a sufficient logic swing (V_{dd} - 2Φ_0) to guarantee data preservation.

Note that the VRC scheme saves leakage power in the stand-by mode due to the "stacking" of turned-off transistors which in this case occurs between the gating transistors and the internal transistors. In general, having more than one turned-off transistor in a stack creates an intermediate voltage on the parasitic capacitance between transistors (see Fig. 5.7e) which will cause an increase in the effective V_{th} of transistors in the stack. This happens due to (1) establishing a body effect (for M_1 in Fig 5.7e) and (2) reducing the drain-to-source voltage (of M_2 in Fig 5.7e) which lessens the Drain-Induced-Barrier-Lowering (DIBL) effect and so forces V_{th} to increase. Furthermore, the gate-to-source voltage (of M_1) is negatively biased, farther decreasing the subthreshold current.

Due to the stacking effect and differences in the leakage of PMOS and NMOS devices, the leakage current of a CMOS gate can vary broadly with input combinations. An example is shown in Fig. 5.7e for a 2-input NAND gate in a 0.25μm process. This fact can be used to save leakage power in a module (like an adder) if the primary inputs of the module are set to the *vector* that best minimizes power in the sleep mode [88-90].

Fig 5.7e shows an example of a latch that can be used to force a primary input to logic "0". Searching for the "best" vector can be done through circuit simulations controlled using random [88] or genetic techniques [89]. It was illustrated in both [88] and [89] that the stacking effect is not very effective in circuits with deep logic since on average the greater the number of logic levels, the more insensitive the leakage power to primary input combinations. Furthermore, the power of a CMOS inverter is only slightly affected by its input since it does not possess any stacking effect.

In summary, dynamic low V_{dd} circuit techniques requires the use of special technology (as in VTCMOS [82] and SOIAS [41]), suffer from delay and noise degradation due to employing virtual rails which require special sizing of the gating transistors as in MTCMOS [83], MVCMOS [86], and VRC [87], need special circuitry or techniques for preserving data in idle periods, and might not be very effective in saving leakage power in large circuits as in the "stacking" techniques. Furthermore, common to all dynamic low V_{dd} circuits (except possibly those based on "stacking") is the power cost incurred when switching the devices from idle to active states and vice versa. It was shown in [41] that large savings in the *total* power of a system are possible with dynamic techniques if the system spends most of its time in the idle state (for example, 98%). This is necessary in order to offset the power penalty paid in turning devices on and off (or in varying their threshold dynamically as in [82-41]).

Unlike dynamic techniques, static approaches do not employ any control signal for power savings. Parts (f) and (g) of Fig 5.7 give two examples of static techniques. In the dual V_{dd} [91] approach, two supply voltages are provided: high V_{dd} (HV_{dd}) and low V_{dd} (LV_{dd}). Gates with large slacks are made to operate at LV_{dd} thus reducing dynamic power. On the other hand, those along the critical paths, which represent a small portion, remain as HV_{dd} nodes. On the positive side, the availability of HV_{dd} permits operating at high frequencies. Furthermore, this approach does not suffer from leakage current increase since *no* low threshold devices are used. However, the amount of power savings is limited since LV_{dd} can not be greatly lowered than HV_{dd}, particularly at high frequencies. In addition, the dual V_{dd} technique requires inserting a level-shifter whenever a LV_{dd} gate fanouts to a HV_{dd} one in order to stop any short-circuit (DC) current in the HV_{dd} gate. Using level-shifters increases the area penalty of the circuit. Finally, efficient DC-DC converter has to be used in order to generate a low supply from the main supply (HV_{dd}).

Another example of a static low V_{dd} scheme is the dual V_{th} approach shown in Fig 5.7g [92]. In this case a very low supply (LV_{dd}) is used along with two threshold voltages: HV_{th} and LV_{th}. Gates off the critical paths are made to operate at HV_{th} thus minimizing leakage power while those which are critical remain as LV_{th} nodes. While effective in reducing both dynamic and leakage power, the dual V_{th} suffers from speed degradation. As the results in the previous section have shown, this is due to the fact that a very low (infeasible) value of LV_{th} is needed otherwise to achieve high-performance at low LV_{dd} (for example, 1V). Therefore, the dual V_{th} scheme is a good choice for low-power applications running at low-to-moderate frequencies.

Note that the static schemes do not have the (previously summarized) drawbacks associated with dynamic approaches. On the other hand, leakage power is not almost completely cut off in the standby mode like in the MTCMOS technique. This is because some gates, as in the dual V_{th} approach, have to operate at low V_{th} all the time. In the next section, the dual V_{dd} dual V_{th} (DVDV) approach, introduced at the beginning of the chapter, will be described. This scheme tries to capture the advantages of both the dual V_{dd} and the dual V_{th} scheme; namely high-performance, low dynamic power and low leakage power.

5.5 LP HP Design Using the DVDV Approach

The dual V_{dd} dual V_{th} (DVDV) approach is proposed as a means for ***low-power*** design of ***high-performance*** CMOS circuits. It comes in a mid-point between the dual V_{dd} and the dual V_{th} schemes described in the previous section. In DVDV, each node in a network can assume any of the three gate implementations listed in Table 5.1 under the DVDV column. Nodes with zero or very small slacks are kept as (HV_{dd}, HV_{th}) gates. On the other hand, nodes with relatively larger slacks are re-mapped to (LV_{dd}, LV_{th}) gates. This will considerably reduce dynamic power dissipation as compared to the dual V_{dd} scheme. In that case, re-mapping is done using the slower (LV_{dd}, HV_{th}) gates and so a *higher* LV_{dd} has to be used to maintain performance. Using LV_{th} gates incurs high penalty in the leakage current, as mentioned before. In DVDV and like the dual V_{th} scheme, this power can be significantly reduced by using (LV_{dd}, HV_{th}) gates whenever possible. The advantages of the DVDV scheme can be summarized as:

Table 5.1: Spectrum of V_{dd} and V_{th} scaling.

No Scaling	dual-V_{dd} [91]	DVDV (This Work)	dual-V_{th} [92]	only V_{dd} scaling
HV_{dd}, HV_{th}	HV_{dd}, HV_{th}	HV_{dd}, HV_{th}	LV_{dd}, LV_{th}	LV_{dd}, HV_{th}
	LV_{dd}, HV_{th}	LV_{dd}, LV_{th}	LV_{dd}, HV_{th}	
		LV_{dd}, HV_{th}		

- it allows high-performance designs due to the availability of the (HV_{dd}, HV_{th}) gates.
- it reduces dynamic power dissipation because LV_{dd} can be scaled deeply, owing to having LV_{th} devices.

- it reduces leakage power by using HV_{th} transistors whenever possible.

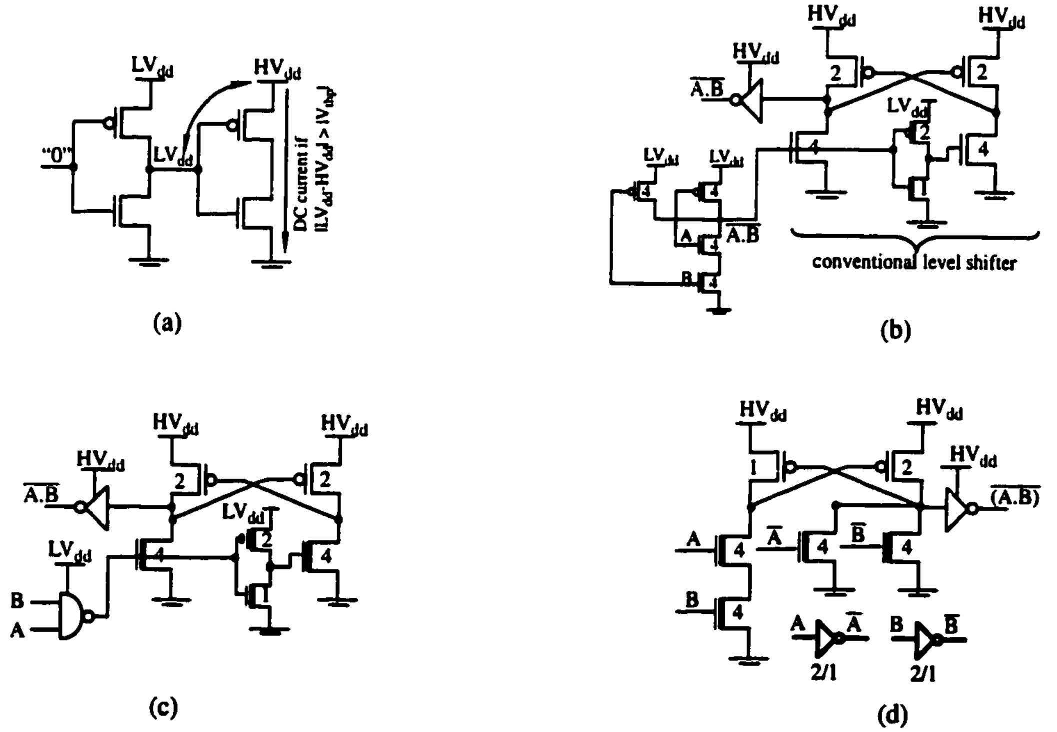
Notice that having gates operating at HV_{dd} is also advantageous from delay sensitivity point of view. Recall from Section 5.3 that the delay sensitivity of a CMOS gate to V_{th} variations increases as the supply voltage is scaled down, particularly for velocity saturated devices. The most sensitive nodes in a logic circuit are those on the critical path(s) since they have zero slacks and so any slight increase in their delays will affect the speed specification of the design. In the DVDV scheme, if these nodes are implemented using (HV_{dd}, HV_{th}) gates, then their sensitivity to V_{th} variations will be greatly suppressed (see Fig. 5.6). Fortunately, mapping these nodes to (HV_{dd}, HV_{th}) gates is the same means for achieving high-performance designs, as outlined before.

This however might not solve the whole problem since some nodes are implemented using LV_{dd} gates. The sensitivity of those nodes will be affected as a result, forcing them to be critical. A trade-off between power and delay sensitivity has perhaps to be made then by re-mapping some of the nodes implemented using LV_{dd} to HV_{dd} gates. Although not exploited in this work, the availability of different types of cells in the DVDV approach gives us the flexibility to do so, which is a definite advantage over other approaches. Note that having to worry about the delay sensitivity in the low voltage era adds a *fourth* dimension to the (by now) conventional three dimensional design space of area, speed and power. Other solutions to the sensitivity problem are: (1) reducing ΔT_{th} in EQ 5.7, which is a definite challenge to device engineering [58], or (2) dynamically varying the threshold voltage as in VTCMOS scheme [82] [93], and this requires triple well process.

When using two supply voltages (HV_{dd} and LV_{dd}) in a logic circuit, a level shifter (*LS*) has to be inserted whenever a LV_{dd} gate fanouts to a HV_{dd} one. If otherwise connected directly, a large DC current may flow in the HV_{dd} gate whenever the LV_{dd}

gate outputs high. In that case, the PMOS transistor of the *HV_{dd}* gate is not completely turned off, as illustrated in Fig. 5.8a. The conventional *LS* employed in the dual *V_{dd}* approach [91] uses a Cascode Voltage Swing Logic (CVSL) inverter, shown in Fig. 5.8b. At low *LV_{dd}*, inverting of the cross coupled PMOS transistors becomes very slow which leads to large short-circuit power.

This problem can be circumvented by using *LV_{th}* transistors since they are available in the DVDV scheme, as demonstrated in Fig. 5.8c. This will allow faster switching of the cross coupled PMOSs leading to less short-circuit current. For NAND gates, it is possible to further improve the speed by embedding the *LS* within the gate as in Fig.



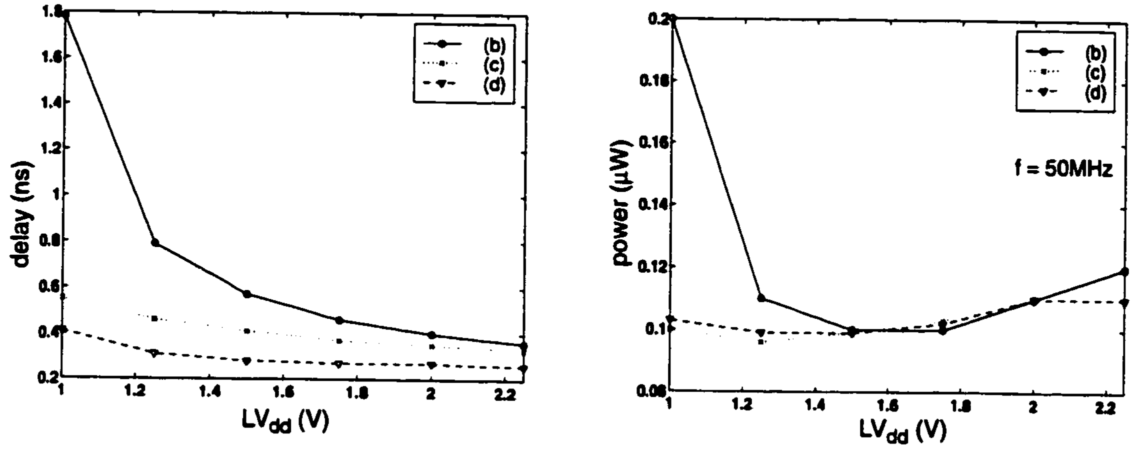
(a)

(b)

(c)

(d)

Figure 5.8: (a) Direct connection between LV_{dd} and HV_{dd} gates, (b) NAND gate with conventional level shifter [91], (c) NAND gate with improved level shifter, (d) NAND gate with embedded level shifter.



(a) (b)

Figure 5.9: Comparing the level shifters in fig. 5.8 in terms of: (a) delay, and (b) power dissipation.

5.8d. The circuits in Fig 5.8 (parts b, c, and d) were simulated in HSPICE using a 0.25µm process and the delay and power comparisons are shown in Fig 5.9a and 5.9b, respectively. For each LV_{dd} value, LV_{th} was set to 0.2xLV_{dd}. Fig 5.9 shows that the embedded configuration (Fig. 5.8d) is the fastest and consumes similar total power as the one in Fig 5.8c. Simulations have shown that the circuits in Fig. 5.8c and Fig 5.8d consume more leakage power than the one in Fig. 5.8b because they employ LV_{th} devices. This power is only significant at very low frequency and/or under low activity (see EQ. 5.2). Note that using level shifters between gates incurs some area penalty. However, as the results at the end of the chapter will show, this penalty is relatively small.

In addition to the level shifters overhead, the DVDV scheme requires the use of a dual V_{th} process, similar to the MTCMOS [83] and the dual V_{th} [92] approaches. Note however, unlike the VTCMOS [82] which requires triple well process and SOIAS [41] which demands dual gated SOI, having dual V_{th} can be easily implemented with an extra implant step to create the high V_{th} devices. Recently, several dual V_{th} MOSFET processes were developed as in [94], making implementing logic circuits using the DVDV approach more feasible.

Another concern, which is common to the dual V_{dd}, is how to place and route (P&R) the cells using standard P&R tools. The problem arises because some cells are supplied from the HV_{dd} rail while others have to be connected to the LV_{dd} supply. An efficient layout strategy which causes less than 3% area penalty per cell was proposed in [95], and is shown in Fig. 5.10. Each cell has two power lines (HV_{dd} and LV_{dd}), but only one can be connected to the interior of the cell depending on its type. Owing to the ability to butt the two types of cells arbitrarily, existing P&R tools can be used without any modification.

Finally, similar to the dual V_{dd} approach, an efficient DC-DC converter is needed in order to generate the low supply voltage from the high one without wasting too much energy in the conversion. The energy efficiency of the converter (η) can be loosely

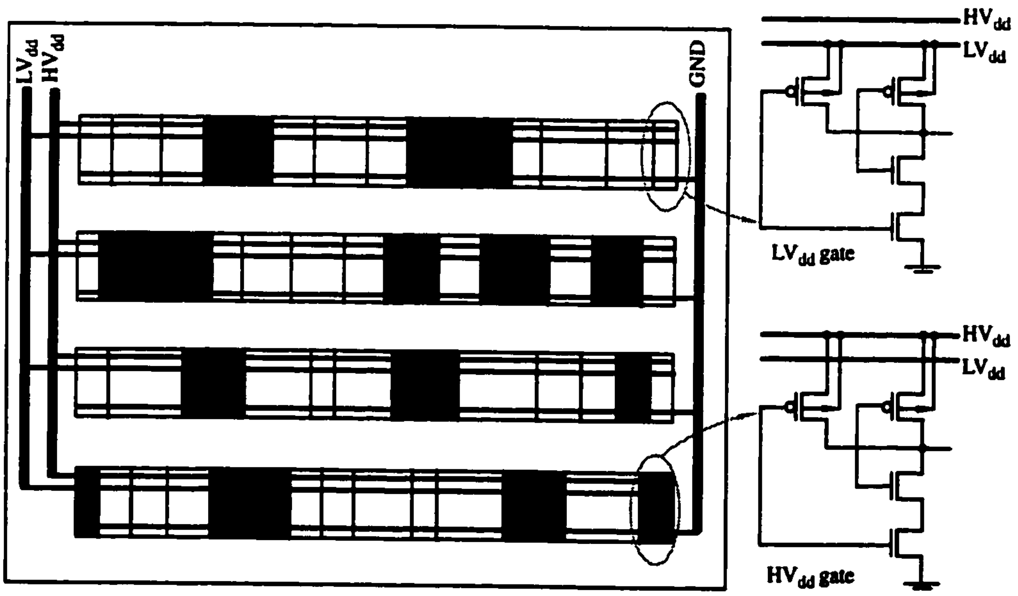


Figure 5.10: Layout architecture using dual supply voltages [95].

defined as the ratio between the energy supplied by the converter to the energy it consumes, with an ideal value of unity. In [96], a first order analysis was presented to determine how much DC-DC efficiency is needed to make the dual V_{dd} approach viable. The following result that reveals the system energy savings (as compared to using HV_{dd} alone) was obtained:

$$\%\text{ savings} = \beta\left(1 - \frac{LV_{dd}^2}{HV_{dd}^2 \cdot \eta}\right) \tag{5.8}$$

where β is the percentage of logic gates that will run at LV_{dd}. Note that in the ideal case (η = 1), EQ 5.8 reduces to the energy savings obtained when ignoring the DC-DC conversion. The break even point (0% savings) occurs when $\eta = \frac{LV_{dd}^2}{HV_{dd}^2}$. For example, if HV_{dd} = 2.5V and LV_{dd}= 1.5V, then η has to be at least 36% to avoid losing energy. This limit is far exceeded by some newly designed DC-DC converters as in [97], which achieves more than 90% efficiency for 6V to 1.5V conversion.

In the next section, a Depth-First-Search (DFS) based heuristic that applies the DVDV technique to a CMOS logic circuit is described. The procedure attempts to minimize the total power dissipation of a circuit while meeting a delay constraint.

5.6 Algorithm

5.6.1 Preliminaries

A combinational logic circuit is represented as a directed acyclic graph $G(V,E)$. Each node $x \in V$, which is neither a primary input nor a primary output, maps to one gate in a given library. An edge $e = (x_1, x_2) \in E$ if the output of node x_1 connects to an input of x_2. The delay of a node x, $d(x)$, is defined as the maximum delay from any of the inputs of x to its output.

Following the terminology used in [98], each node $x \in V$ is labeled with two values: the arrival time, $a(x)$, defining the delay of the slowest path from the primary inputs to x, and the required time, $r(x)$, which is the latest time the signal has to be available at the output of x. The arrival time of a primary input (PI) is taken to be zero while the required time at a primary output (PO) is given by:

$$r(v) = M \cdot T_{cycle} \quad \forall (v \in PO) \tag{5.9}$$

where T_{cycle} is the delay of the critical path(s) of the circuit as determined from an initial mapping, and M is a multiplication factor that can be set by the user and has a default value of 1. Given the above, $a(x)$ and $r(x)$ of a node x can be mathematically defined in EQ 5.10, and EQ 5.11, respectively:

$$a(x) = \max_{y \in FI(x)} (a(y) + d(x)) \tag{5.10}$$

$$r(x) = \min_{y \in FO(x)} (r(y) - d(y)) \tag{5.11}$$

where $FI(x)$ and $FO(x)$ are the sets of fanin and fanout nodes of node x, respectively. Assuming the above notions of $a(x)$ and $r(x)$, the slack of x is defined as the amount of time by which node x can be slowed down, or:

$$s(x) = r(x) - a(x) \tag{5.12}$$

Note that in order not to violate the required timing constraint set by EQ 5.9, each node $x \in V$ should have its $s(x) \geq 0$. Nodes with zero slacks are those on the critical path(s).

Assuming the above definitions, we now describe a procedure of *two phases* to implement the DVDV technique.

5.6.2 DFS Heuristic

Given: (1) an initially mapped logic network, *G*, using (HV_{dd}, HV_{th}) gates only, (2) a timing constraint as specified in EQ 5.9, and (3) a DVDV library with all possible combinations given in Table 5.1, the following heuristic attempts to minimize the total power of the network while satisfying the required timing constraint.

As an initial step, the nodes of the (mapped) network are traversed in a DFS manner starting from the *PO*s. The result of this traversal is stored in an array ζ such that each node appears somewhere before all of its transitive fanin nodes.

In phase I of DVDV, the nodes in ζ are processed in order one at a time, and an attempt is made to replace each node with a functionally (and size-wise) equivalent, but slower, (LV_{dd}, LV_{th}) gate. This replacement is committed if and only if the slack of x after replacement, $s^*(x)$, remains non-negative. Note that $s^*(x)$ is given by $r(x)$ - $a^*(x)$, where:

$$a^*(x) = \max_{y \in FI(x)} (a(y) + d^*(x)) \tag{5.13}$$

and $d^*(x)$ is the delay of node x if x is to be implemented using (LV_{dd}, LV_{th}) gate. From EQ 5.1, $d^*(x)$ can be simply obtained by scaling $d(x)$ as follows:

$$d^*(x) = \frac{LV_{dd}(HV_{dd} - HV_{th})^{\bar{\alpha}}}{HV_{dd}(LV_{dd} - LV_{th})^{\bar{\alpha}}} \cdot d(x) \tag{5.14}$$

If the above calculations give a positive $s^*(x)$, then it is safe to assign (LV_{dd}, LV_{th}) to node x, otherwise x is not replaced. If x is replaced, then the slacks of the affected nodes in the network are updated (using EQ 5.10-5.12) before proceeding to the next node in ξ.

Notice that before attempting to replace a candidate node x, a check is made to see if the output of x has a fanout to a (HV_{dd}, HV_{th}) gate. If so a *LS* is needed to prevent DC current dissipation as discussed before. This *LS* is inserted at the output of node x, and its delay (obtained from a look-up table) is summed to that in EQ 5.14 before calculating $s^*(x)$. As in [91], the *PO*s of the circuit at hand are assumed to supply HV_{dd} signals. Therefore, a *LS* is also inserted at the output of node x if this output happens to be a *PO*, and x is a candidate for replacement. The last condition is not necessary and can be eliminated.

When a *LS* is added, it is difficult to check right away whether its power overhead will be offset by other (unvisited) nodes in its transitive fanin. Therefor, a check is made at the end of phase I to remove all redundant level-shifters. A redundant *LS* is a one used at the output of a LV_{dd} node x, such as all the fanin nodes of x are mapped to (HV_{dd}, HV_{th}) gates. If such a *LS* is found, its power dissipation is not outweighed and so x is replaced back to a (HV_{dd}, HV_{th}) implementation and timing information is updated.

Phase II of DVDV is similar to phase I, except that only nodes assigned to (LV_{dd}, LV_{th}) in phase I are re-visited in this phase in an effort to substitute them with the even slower (LV_{dd}, HV_{th}) gates to minimize leakage power. Delay recalculation, slack checking and timing information updating proceed in a similar fashion as in phase I. At the end of phase II, the total power dissipation is calculated by:

$$P_{total} \approx (f \cdot A) \cdot \sum_{i}^{N}\left(C_L^i \times V_i^2\right) + \sum_{i}^{N}\left(V_i \times I_{leak}^i\left[V_{th}^i\right]\right) + \text{power of level shifters} \quad (5.15)$$

which is a modified version of EQ 5.2. In the above equation, V_i represents the supply voltage of gate *i* which can be HV_{dd} or LV_{dd}, and V_{th}^{i} is the threshold voltage of gate *i* with HV_{th} or LV_{th} as possible values. The total power in EQ 5.15 is compared to the original power (EQ 5.2) and the modified solution is accepted if it reduces the power. Note that in both EQ 5.2 and EQ 5.15, *f* is taken as $1/(M.T_{cycle})$.

In the above description of the heuristic, it was assumed that both LV_{dd} and a related LV_{th} values are given. Alternatively, the procedure can be used to search for the (LV_{dd}, LV_{th}) combination that can "best" minimize the total power without violating the required cycle time. This usage will be illustrated in the next section.

5.7 Results and Discussions

The proposed heuristic was implemented in the *C* programming language under the SIS-1.2 CMOS logic synthesis environment [99]. To simplify the technology-mapping step, a simple library containing BUFFER, INV, NOR, and NAND gates was used. HSPICE simulations was carried out using a model file of a 0.25 μm process. Simulations to extract the delay and capacitance parameters needed by the SIS routines as well as the leakage current of a gate were performed. The latter was found as the average over all possible input combinations.

In the following experiments, HV_{dd} and HV_{th} were fixed at 2.5, and 0.5, respectively. Logic optimization and technology mapping for each circuit was performed in SIS using the script *script.delay* which optimizes the circuit for minimum delay. Only (HV_{dd}, HV_{th}) gates were allowed at this step. T_{cycle} of a circuit was taken as the delay of the critical path(s) obtained from SIS after mapping. Using the above 0.25 μm process,

T_{cycle} ranged from 0.32ns to 4.43ns for a set of ISCAS and MCNC benchmarks. For dynamic power calculation, the SIS's gate-level *power_estimate* command with random primary inputs transition probability was employed to extract C_L^i for each gate i. The later was used for power calculations in EQ 5.2 and EQ 5.15. To find the LV_{dd} value that results in the largest power savings, the heuristic was repeated six times for LV_{dd} of 2.25V, 2.0V, 1.75V, 1.5V, 1.25V, and 1.0V. For each case, LV_{th} was taken as $LV_{dd}/5$.

To compare the power savings of the DVDV approach to the dual V_{dd} scheme, we performed two runs. In the first run, we used a library having all the three DVDV gate combinations (Table 5.1) and the modified *LS*s (Fig. 5.8c and 5.8d). For each circuit, the (LV_{dd}, LV_{th}) combination that best minimized the total power for a given timing constraint was chosen. In the second run, a library having the two dual V_{dd} combinations (i.e, no LV_{th} is allowed in this case) and the conventional *LS* (Fig 5.8b) was used. For fairness, the LV_{dd} value that resulted in the minimum power dissipation was selected. Note that in this case, only a single phase is needed to re-map nodes to (LV_{dd}, HV_{th}) gates whenever possible. Further note that EQ 5.14 is slightly modified since LV_{th} is not used.

Fig. 5.11 shows an example of the results for circuit C3540. For different values of M (timing constraints), the figure gives the total power dissipation obtained using a single HV_{dd}, the dual V_{dd} scheme, and the DVDV approach. For this example, the dual V_{dd} saves 41% on average of the original power (using a single HV_{dd}). The DVDV scheme further improves upon the dual V_{dd} by 42%. A summary of the results for all circuits is given in Table 5.2. For each M value, the corresponding entry shows the percentage of power savings using the DVDV scheme over the dual V_{dd} approach, or:

$$\frac{dual\,V_{dd} - DVDV}{dual\,V_{dd}} \times 100\% \qquad (5.16)$$

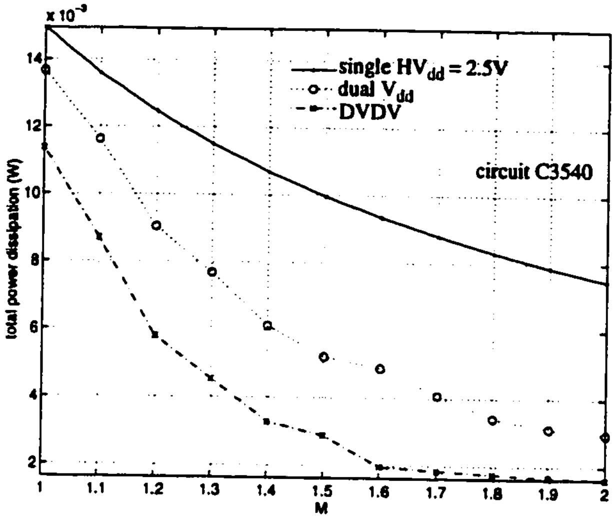


Figure 5.11: Total power dissipation using different approaches and values of M.

The last row of Table 5.2 gives the average savings over all circuits for each *M* value. It is clear that the DVDV scheme always outperforms the dual V_{dd}. This is because DVDV utilizes the faster (LV_{dd}, LV_{th}) gates and the modified *LS*s, thus allowing a lower "optimal" LV_{dd}. Also note that at low values of *M* (very tight timing constraints), not many nodes can be re-mapped to LV_{dd} gates and/or LV_{dd} can not be scaled profoundly, resulting in low to moderate savings. As we relax the timing requirements, better savings are possible as the table shows.

The above observations are clarified in Fig. 5.12 and Fig. 5.13. Fig 5.12 plots the "optimal" LV_{dd} averaged over all circuits as a function of *M*. The figure clearly shows that DVDV allows deeper scaling of the supply voltage than the dual V_{dd} scheme, thanks to the availability of LV_{th}. It further illustrates that at low *M*, LV_{dd} is not significantly reduced on the average. Since the benchmark circuits have divergent initial T_{cycle} (see the third column of Table 5.2), large error bars (deviations) are produced at small values of *M*. This tells us that the "optimal" LV_{dd} (and so the related LV_{th} in the DVDV case) varies largely depending on the circuit and the required operating frequency. How-

Table 5.2: % savings of DVDV over dual V_{dd}.

Circuit	No. of Gates	T_{cycle} (ns)	A = 1.0 and M =										
			1.0	1. 1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
C1355	788	2.37	0.0	21.1	33.5	41.4	36.9	49.3	47.3	61.2	55.9	38.3	38.3
C17	9	0.32	0.0	0.0	0.0	0.0	14.4	26.3	39.0	48.9	47.6	42.7	39.5
C1908	863	3.01	0.0	16.1	35.7	38.9	45.1	48.1	49.7	59.0	50.5	46.0	45.7
C3540	1572	4.43	16.9	25.2	35.9	41.0	46.3	44.6	59.7	54.8	48.7	46.5	45.2
C432	257	3.42	0.0	16.4	19.2	26.7	30.5	36.2	32.0	40.9	40.9	32.1	30.6
C499	640	1.96	0.0	16.1	37.6	44.2	40.8	52.2	42.9	59.8	59.8	34.0	34.1
C5315	3119	4.13	23.5	21.3	35.5	42.6	47.8	50.2	53.5	51.7	46.4	45.6	45.6
C880	566	3.03	18.0	26.6	33.6	39.5	45.2	46.3	51.1	50.3	47.6	43.4	42.0
i1	51	0.77	9.1	20.0	22.6	23.6	32.8	34.6	28.6	39.2	36.1	30.1	25.6
i2	279	1.88	33.7	32.0	39.6	47.1	56.7	54.0	56.0	60.1	58.7	58.6	55.0
i3	258	0.65	0.3	0.3	13.8	46.3	56.8	39.2	38.9	43.2	60.0	48.0	47.9
i4	232	1.75	12.0	11.7	38.3	46.3	51.9	51.3	58.3	62.8	58.2	53.6	51.6
i5	420	1.57	13.2	18.8	15.6	41.8	12.8	17.6	16.3	10.4	4.7	7.6	3.7
i6	543	0.58	0.0	0.0	24.3	16.8	41.0	48.0	45.9	52.7	52.2	51.9	54.0
i7	704	0.60	0.0	0.0	24.1	35.1	48.4	45.7	56.4	51.4	43.9	59.9	59.9
i8	1435	1.76	13.2	23.9	35.3	41.7	46.1	52.8	51.2	49.1	46.4	44.8	43.6
i9	745	1.50	0.0	29.6	49.2	47.5	52.6	50.0	45.1	45.3	44.8	39.6	32.9
i10	3030	4.11	18.6	19.4	28.0	34.8	40.1	39.7	39.5	37.3	33.2	31.0	30.9
Average			**8.8**	**16.6**	**29.0**	**36.4**	**41.5**	**43.7**	**45.1**	**48.8**	**46.4**	**41.9**	**40.3**

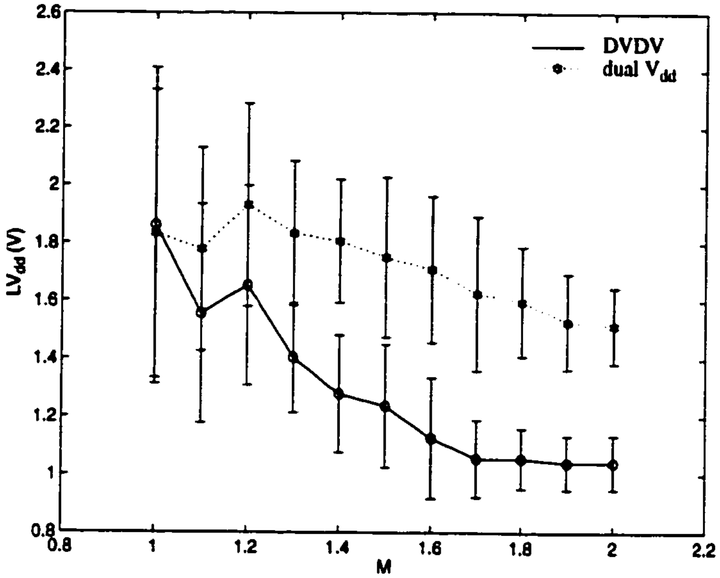


Figure 5.12: "Optimal" LV_{dd} using DVDV and dual V_{dd} for various M values.

ever, as *M* increases, most circuits can operate at the same supply voltage, which in the DVDV case equal to 1V.

Fig. 5.13 gives the breakdown of the gates replaced using the DVDV approach for circuit (i8). As *M* increases, it is possible to operate more gates at a low supply voltage (LV_{dd}) using either HV_{th} or LV_{th}. In fact, at some point ($M > 1.8$, for this example), *all* gates can operate at LV_{dd}. At this stage, the DVDV scheme converges to the dual V_{th} approach [92][1]. This confirms our early observation that the dual V_{th} is a good choice for low-power but not very high-speed designs.

In the previous simulations, it was assumed that the activity factor (*A*) (EQ 5.2 and EQ 5.15) is one. This might not be the case in systems which are not continuously operational, or in other words, even-driven [41]. In such environments where the hard-

[1] The few level shifters shown in the figure at high values of *M* are those used at the primary outputs. They can be eliminated if not needed as mentioned before.

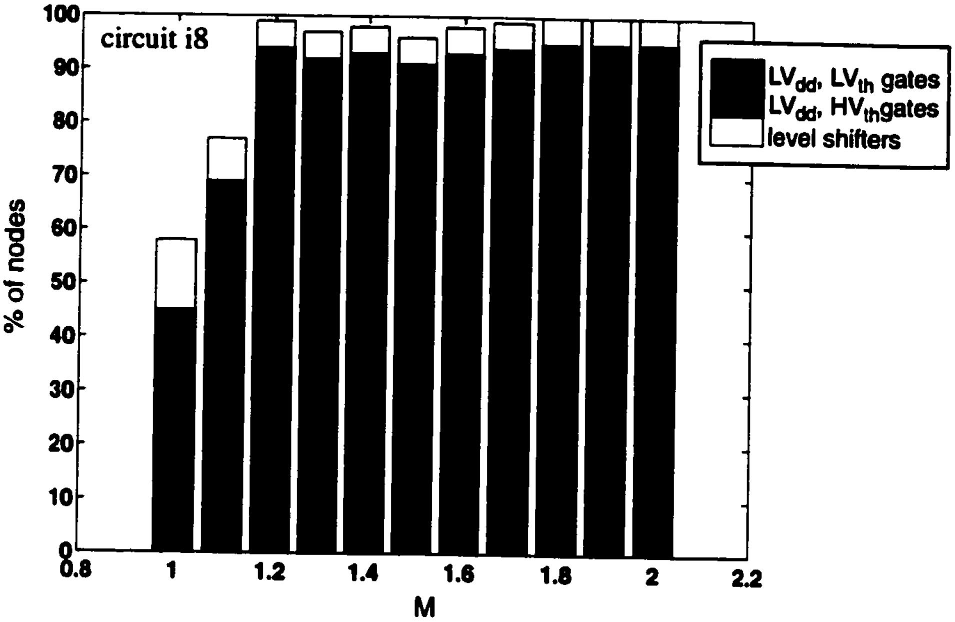


Figure 5.13: Breakdown of replaced nodes using the DVDV method.

ware can be idle for long periods of time, leakage power comes into respect. To illustrate this, Fig. 5.14 plots the same figure-of-merit used in Table 5.3 for circuit C880 under various activity factors and for three delay constraints. Note that unlike *M*, the

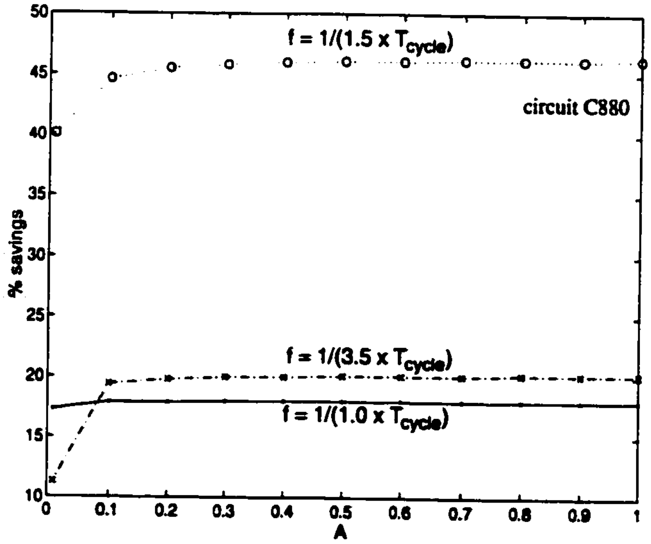


Figure 5.14: Effect of frequency (f) and activity factor (A) on DVDV savings.

factor (A) is not considered when making replacement decisions and is only used at the end of the procedure to calculate the final power. Fig. 5.14 shows that at high frequencies ($M = 1$ and 1.5) and almost independent from A, the DVDV approach is always better than the dual V_{dd}. As (A) becomes small and drops to 1% and below, the power savings of DVDV is only slightly decreased. However, as the operating frequency is lowered, the advantage of DVDV will become adversely sensitive to low (A) values since $P_{leakage}$ becomes more important due to long idle time (1-A). Similar results were obtained for other circuits.

From the simulations reported in Table 5.2, it was found that both the dual V_{dd} and the DVDV schemes incurred reasonable area penalties in terms of the percentage of level shifters used, as illustrated in the example of Fig 5.13. Running times on an Ultra Sparc station with 120MB memory were 23 and 6 minutes for the largest circuit, using the DVDV and the dual V_{dd} approaches, respectively. The DVDV scheme takes longer time because it requires a second phase and replace more nodes in the first phase.

5.8 Summary

In this chapter, we studied the effect of supply voltage scaling on circuit performance. It was shown that without a corresponding decrease in the threshold voltage, the delay increases rapidly. An important relation (EQ. 5.6) that expresses the new low threshold voltage in terms of the new scaled supply, the old high supply and threshold voltages, and the velocity saturation factor $\bar{\alpha}$ is derived. Scaling V_{th} according to that relation for equal delay results in a very high penalty in terms of the leakage current. Furthermore, it was shown that the delay sensitivity to V_{th} variations is exacerbated at low V_{dd}, especially for velocity saturated devices.

To address the conflicting requirements of high-speed and low-power, a new design approach, called DVDV, targeting very high-speed and low-power designs is proposed. DVDV reduces both dynamic and leakage power through the use of two V_{dd} voltages and two V_{th} voltages. The main advantages of this technique are:

- it allows high-performance designs due to the availability of (HV_{dd}, HV_{th}) gates. In other words, high-performance is achieved without the need for excessive V_{th} scaling which comes at the expense of high leakage current.
- it reduces dynamic power dissipation because LV_{dd} can be deeply scaled, owing to having LV_{th} devices.
- it reduces leakage power by using HV_{th} transistors whenever possible.
- it allows faster level shifters which also consume less dynamic power than conventional ones.
- it requires a dual V_{th} CMOS process which adds minor alterations to current design trends and technologies.
- as explained in Section 5.5, the availability of different types of cells in the DVDV approach gives us the flexibility to trade-off between power and delay sensitivity to V_{th} variations which increases at low supply voltages.

A heuristic of two phases to implement the DVDV technique is described. The algorithm is based on depth-first-search and the user has control over supply and threshold voltages as well as latency constraints. Exercising DVDV over a set of ISCAS and MCNC benchmark circuits has indicated that circuits can largely benefit (consume less power) from this scheme, particularly at very high frequencies.

Chapter 6

Power Macro-Modelling for RTL Power Estimation

6.1 Introduction

As stressed in the previous chapters, low-power consumption has become a primary target in contemporary digital VLSI design. Two main factors are behind this trend. First, increased circuit speed accompanied with high integration densities have resulted in significantly large heat dissipations. Unless power consumption is brought down to acceptable levels, reliability is affected and expensive cooling and packaging systems are required. Second, battery-driven mobile electronics put a high demand on low-energy consumption in order to extend the battery life to the maximum possible level before recharging is needed.

The above factors have added to the complexity of digital IC design since low power has become an important design goal with equal footings as high-speed and small area. To minimize the power consumption of a digital system, the design that dissipates the least power has to be chosen provided that it meets some speed performance and/or area constraints. To aid designers to achieve that in a reasonable amount of time, power-sensible CAD tools and methodologies at all levels of the design hierarchy have been developed over the past few years [3][100].

More recently, particular attention has been given to power optimization techniques at the architectural, also called Register-Transfer (RT), level because design experience and research efforts have shown that decisions made at the highest levels could have the largest impacts on power savings At the RT level, different schedules and/or bindings of components can result in significantly dissimilar power savings. To assess the effects of these and other "what-if" decisions on power consumption, one possibility is to map *each* RT design to gate-/transistor-level implementation and then use accurate gate-/transistor-level tools for power estimation. However, this approach is extremely unpractical because: (1) the number of feasible high-level decisions might be substantial, and (2) modern circuits contain a sizable number of components which take a long time to evaluate. This motivates the need for developing a methodology for estimating power at the high-level of abstraction.

To be useful, an architectural-level power macro-model should possess the following features. First, it must be simple to evaluate, otherwise there would be no advantage in using it and one might as well resort to low-level tools. Secondly, it should be accurate enough to be applied over a wide range of *input stimuli* (for example random or correlated) and high-level components (for example, different adders, multipliers, etc.). Note that at the architectural-level, relative accuracy is much more important than the absolute one, since the objective is to find out whether one alternative design is better than the other. Finally, a high-level model should have a well-defined mechanism in which a certain number of parameters can be extracted by an automated *characterization* process, with minimal user intervention.

In this chapter, a new technique to characterize the power dissipation of data-path components (that is, adders, multipliers, etc.) is presented. Given a *hard* data-path module with N primary inputs, and M primary outputs, exactly Nx$(N+M+1)$ effective capacitance coefficients are derived for that module. A transistor-level tool is used for capacitance extraction. The proposed methodology is based on linear regression theory. In addition to the above features, the technique presented here allows the user to trade-off between accuracy and characterization time, and propagates power information directly from transistor-level (real) implementations. Simulation experiments on a set of data-path components with various sizes are performed. The results illustrate that as compared to a previously published approach [101], our scheme significantly improves the accuracy of RTL power estimation and produces results within 15% from a transistor-level tool on the average.

The rest of this chapter is organized as follows. In the next section, we define the concept of effective capacitance based on the power equations given in Chapter 2. In Section 6.3, we quickly review power estimation at the transistor, gate, and architectural levels. Section 6.4 presents the proposed power characterization technique, and simulation results to verify the applicability of the approach are given in Section 6.5. The chapter ends with a summary.

6.2 Effective Capacitance

In a static CMOS circuit, power is mainly dissipated as a result of input signal transitions. This power is called dynamic power and it has two components: switching power and short-circuit power. Switching power is used for charging/discharging capacitive loads during logic transitions. For a circuit with S gates, this power is given by:

$$P_{switch} = \left(\sum_{i=1}^{S} \alpha_i C_i\right) \times V_{dd}^{2} \times f_{clk} \tag{6.1}$$

where C_i is the lumped capacitive load at the output node of gate i, V_{dd} is the supply voltage, f_{clk} is the clock frequency which controls the rate at which the circuit's primary inputs change values and, α_i is the *switching activity* of gate i defined as the expected (average) number of power consuming (or 0-to-1) transitions the output node of gate i will make during a clock cycle ($1/f_{clk}$). The term between the parentheses in equation (6.1) represents the average switched capacitance per clock cycle or C_{switch}.

In a static CMOS gate, a short-circuit transient current flows between V_{dd} and ground whenever both the NMOS and PMOS devices are turned on during logic transitions. Similar to the switching power, the short-circuit power can be approximated through an equivalent short-circuit capacitance (C_{sc}) [102]. In a well-designed circuit, C_{sc} is usually much smaller than C_{switch} (10 to 20% of the total dynamic power). However, in some cases, it represents a significant portion and should not be neglected. Thus, the total *effective capacitance* is given by:

$$C_{eff} = C_{switch} + C_{sc} \tag{6.2}$$

Indeed, estimating the dynamic power in CMOS circuits mainly involves predicting C_{eff} since both V_{dd} and f_{clk} are assumed to be given. Because C_{eff} depends on the input activities, the problem of power estimation is known as highly input pattern dependent. This means that to accurately estimate power consumption, it is essential to account for these activities.

Notice that the leakage power is ignored in the above discussion. As shown in the previous chapter, this power can not be neglected as V_{th} is scaled down. In this scenario, design techniques (like those given in Chapter 5) to virtually eliminate leakage power must be employed.

6.3 CMOS Power Estimation Tools

The problem of power estimation has attracted lots of interest over the past years. As a result, the amount of literature which tackled this problem is immense. In this section, we will give a brief overview of the main state-of-the-art efforts in power estimation. For the sake of thoroughness, we will start with transistor level power estimation tools and then move up to higher level approaches.

6.3.1 Transistor Level Tools

In general, transistor level power estimation tools are *simulative*, meaning that a digital circuit is simulated directly with a set of vectors that represent the "typical" behavior of the input data. At the end of simulation, the total current is divided by the total simulation time (number of cycles times cycle width) and the resultant average is multiplied by V_{dd} to get the average power. A well-known example of transistor-level tools is HSPICE from Meta-Software Inc. [103].

Transistor-level tools are generally very accurate and can estimate power dissipation irrespective of circuit type, design style or logic family employed. This is achieved through time-driven simulation coupled with detailed device models. As a result, these tools are very slow if used to estimate the power of large circuits.

Several efforts have been made in the past to improve the speed of transistor-level tools without compromising their accuracy. An example is PowerMill [104] from EPIC Inc. PowerMill is a transistor-level simulator which is two to three orders of magnitude faster than SPICE, but within 10% of its accuracy. Instead of employing time-driven algorithms as in SPICE, PowerMill uses an event-driven approach along with simplified look-up tables for device models. Upon user's request, PowerMill can report average switching current and average short-circuit current. PowerMill also offers several power diagnosis options such as DC path detection, hot-spot (nodes with high load capacitance and high switching activities) analysis and excessive rise/fall time detection. However, the performance of PowerMill is still far from acceptable for large designs. For example, PowerMill is expected to take about 11 months' CPU time for simulating a design with three million transistors using only 10,000 vectors [105].

More recently, a completely different approach to improve the speed of transistor-level tools (and simulative techniques in general) was proposed in [106]. The basic idea is to generate a compact vector set that is representative of the original input set and can be simulated in a reasonable time. Experimental results shows that a 20x compaction ratio can be achieved with average error less than 2%.

Most transistor-level simulators (other than SPICE) ignore the sub-threshold power. As mentioned before, for deep sub-micron designs with very low threshold voltages, the sub-threshold current can be very significant. A simple analytical model for estimating both switching and sub-threshold power in deep sub-micron digital circuits at the transistor-level was described in [107]. The model shows good accuracy as compared to SPICE results.

6.3.2 Gate Level Tools

At this level, a digital circuit is represented by a a netlist of basic cells or gates. Each cell/gate has a simplified/advanced power and delay models which are created using transistor-level tools and are stored in a library. Since the circuit is described in terms of gates/cells rather than transistors, gate-level power estimation is typically orders of magnitude faster than at the transistor level. Roughly speaking, gate-level techniques can be divided into two categories: *simulative* and *non-simulative.*

Just like transistor-level techniques, gate-level simulators (for example [108] and [109]) simulate the circuit using a number of user-specified (or randomly-generated) input vectors. After each input vector transition, the switching activities of all gates (that is, α_i in equation EQ 6.1) are updated. At the end of simulation, the total switching power is calculated using EQ 6.1. In general, gate-level simulators show good degree of accuracy as compared to SPICE for a large range of circuits. However, their main drawback is long simulation times for large circuits with thousands of gates.

Non-simulative techniques uses *probabilities* to characterize the primary inputs behavior. These probabilities are then propagated to each gate in the circuit to estimate its switching activity, and EQ 6.1 is used as before to calculate the switching power[1]. This process is very quick and so probabilistic approaches are usually much faster (10x-1000x) than simulative techniques. However, basic probabilistic techniques suffer from low accuracy. There are three main reasons for this inaccuracy:

1) since an input data stream is represented as a set of probabilities rather than as real vectors, some error is introduced.

[1] See Section 2.1.1.1 for an example.

2) *correlation* (or dependence) between logic signals is ignored. Among logic signals, three kinds of correlations can be identified: temporal, spatial, and spatiotemporal. *Temporal correlation* exists if the values of the same signal in some clock cycles are not independent. *Spatial* correlations exists if the values of two or more signals in one clock cycle are not independent. Finally, *spatiotemporal* correlation exists if the values of two or more signals in some clock cycles are not independent. Correlations can exist between signals on the primary input lines of the circuit due to the nature of the input data, or can be internally generated due to reconvergant fan-out and feed-back lines. In general, different degree of correlations can result in very large variations in power dissipation. In simulative techniques, signal correlation is not a concern since any correlation can be automatically taken care of during simulation. On the other hand, in probabilistic techniques, simple probabilities do not capture any kind of correlations, and this can result in very large errors in power estimation. One possible solution is to use conditional probabilities to describe correlations. However, the number of all possible correlations is exponential and thus accounting for correlations between signals is computationally very expensive.

3) *zero-delay model* is assumed. In a zero-delay model, only steady state transitions are counted, while spurious transitions (glitches) due to signal races are ignored. In [110], it has been shown that the glitching power is normally 20% of the total switching power and can be as high as 70% in some combinational adders and multipliers.

Some heuristic techniques that tackled the above issues in order to improve the accuracy of gate-level power estimation were proposed over the past years for both combinational and sequential circuits [110-117]. In general, these approaches try to achieve a good compromise between estimation speed and accuracy. More recently, a new a hybrid approach which combines both simulative schemes with probabilistic ones was described in [118]. The idea is to use a simulation approach for the highly correlated user-specified control sequence and a probabilistic technique for the weakly correlated input data.

As a result of the extensive research on gate-level techniques, power estimation tools that operate at the gate level are now available from several CAD companies, examples of which are DesignPower from Synopsys Inc. [119] and WattWatcher-Gate from Senté Inc. [128]. However, due to their relatively low speed, the applicability of gate-level tools is still limited to designs with a few thousands gates. For architectural-level low-power optimizations where a single design option can easily have hundreds of thousands of gates, architectural tools are more appropriate. These tools, which deal with blocks rather than gates, are discussed in the next section.

6.3.3 Architectural Level Tools

At the architectural-level, a digital circuit is represented by an interconnected set of macro-modules like adders, multipliers, memory units, control units and multiplexors. An RTL circuit is generated from a Control/Data Flow Graph (CDFG) representation of the design during a process called *architectural synthesis*. In general, a large number of RTL designs can be created during architectural synthesis. For low-power applications, it is important to choose the design with the lowest power dissipation provided that it meets some speed and/or area constraints. Therefore, a fast, yet relatively accurate, tool for architectural-level power estimation is needed.

In general, existing architectural-level power estimation techniques can be classified into: top-down or bottom-up [120]. In top-down approaches like those based on entropy theory [121-122], a combinational circuit is specified as a boolean function, with no information on the circuit structure or number of gates, etc. Estimates of switching activities are obtained based solely on information about the primary inputs and the primary outputs entropies, which can be approximated from input and output signals probabilities, respectively. However, as in all probabilistic-based power estimation techniques, input data correlations are very difficult to account for and so some simplifying assumptions about the input behavior (like temporal and spatial independence) are necessary. It was noted in [120] that top-down approaches are still in their infancy and much work is to be done to demonstrate their applicability.

In contrast, bottom-up techniques are used when one is designing an architecture using previously designed (hard) macros with known internal details. In this case, rather than relating the power of RTL blocks to fundamental parameters such as gate equivalent count or entropy, the approach here is to develop a *power macromodel* for each block in the given library. This macro-model can be used later during high-level power estimation to quickly predict the power dissipation of the block. A clear advantage here is the strong link to real implementations which will give the designer an approximately correct power figures.

An early contribution to power marco-modelling was described by Powell et. al. [123]. The method, called Power Factor Approximation (PFA), treats all input bits of a macro as Uniform White Noise (UWN) data and derive a single (average) C_{eff} value for a given module. Looking up one single value per module during architectural-level power estimation is extremely fast. However, this approach can result in very large errors since the effect of input activities on power dissipation is ignored.

Several schemes to overcome the above limitation have been developed over the past few years. These approaches can be further classified into: *statistical* (*non-simulative*) and *simulative*. The first statistical technique was the Dual-Bit Type (DBT) model [124] which derives an analytical equation with a number of capacitive coefficients for a given module. These coefficients reflect: (1) the random activity of the least significant bits, and (2) the correlated activity of the most significant (sign) bits in DSP data streams. During architectural level power estimation, functional simulation of the system is used to derive a number of *word-level* statistics that describe the data being fed to each module. These statistics such as mean, variance, and temporal correlation are employed to decide about the regions of inputs for which different capacitive coefficients should be used. Although accurate, the DBT scheme is limited because to characterize a module, it requires some human knowledge of its internal structure. This means that a different equation has to be derived for each module depending upon its structure, which is obviously hard.

The above problem was addressed by the input-output statistical model proposed in [125]. The model is based on a single fixed template which is independent of the macro being analyzed. For each module, a three dimensional look-up table is derived. The axes of this table are: (1) the *average* input signal probability (probability of a signal being logic "1") (2) the *average* input transition density (the expected number of logic transition per unit time), and (3) the *average* output transition density. An automatic procedure is used to characterize a module for different combinations of the above three variables. As in the DBT scheme, functional simulation is used to derive the above numbers which are used at the end to look-up the estimated power. A similar approach was also proposed in [126]. The main difference is that other word-level statistics such as spatial correlation, temporal correlation, and glitching activity are included in the macro-model.

As with gate-level non-simulative techniques, signal correlations are difficult to account for. This problem is further exacerbated in architectural-level approaches since only *word-level* correlations can be partially accounted for. Furthermore, ignoring individual primary bits activities can result in large errors since some circuits power are more sensitive to activities on some primary inputs than other. Finally note that signal statistics of a module are highly dependent on the context and/or the application the module is run on, plus they can be dynamic (that is, non-stationary). If the module is used in such an environment which is different than the one assumed during characterization, large errors are also possible.

Approaches based on direct simulation and profiling based on input transitions rather than input statistics were proposed as an alternative to statistical techniques. An early simulative scheme which does not make any assumption about input data statistics nor it requires any knowledge about the module structure was described in [127]. Since the number of possible input transitions for a circuit with N inputs is 2^{2N}, a clustering algorithm which collapses closely related input vector transitions into clusters with approximately equal power values was proposed. These clusters are stored in a table and retrieved during power estimation. The number of clusters produced by this approach for a given module depends on the module size and the level of accuracy as set by the user. For a good accuracy, this number can be very large especially for big circuits and this increases the characterization time. In addition, if these clusters are not distinct, they can not be hashed. This implies that during architectural-level power estimation, one has to linearly search for the cluster corresponding to each input vector transition, which lengthens the estimation time.

A more efficient, though possibly less accurate, simulative method based on regression theory was described by Benini et. al. [101]. The main idea is to model the power as a linear function of single input and output bit activities. The coefficients of the regression model are derived by first simulating the module at the gate-level using a very large sample of input vectors and then solving an over-defined system of equations using least-square fitting techniques. If a circuit has N primary inputs $(I_1,I_2,...,I_N)$ and M

primary outputs $(O_1, O_2, \ldots, O_M)$, then a single equation is obtained to model the power as given by:

$$P = \beta_0 + \beta_1 I_1 + \beta_2 I_2 + \ldots + \beta_N I_N + \beta_{N+1} O_1 + \beta_{N+2} O_2 + \ldots + \beta_{N+M} O_M \qquad (6.3)$$

During architectural level power estimation, functional simulation is performed and the primary inputs and outputs of a module are monitored. In a given cycle, an I_j (for $j = 1,2,\ldots,N$) or an O_k (for $k = 1,2,\ldots,M$) in EQ 6.3 take the value of one if there is a transition on the corresponding *j*th primary input or *k*th primary output of the module, respectively. Otherwise, the corresponding entry will take the value of zero. At each cycle, the power equation is evaluated and at the end of the simulation, the total power is found and averaged over the simulation period.

The linear regression technique (henceforth called LR) is attractive because of its simplicity. Only (M+N+1) coefficients have to derived and stored for each module. Moreover, these coefficients can be quickly retrieved during power estimation. This can be done, for example, by XORing consecutive input (output) vectors as shown in Fig. 6.1.

As mentioned before, to obtain the coefficients in EQ 6.3, the module is first simulated using accurate gate-level power simulator with a large input sample. For

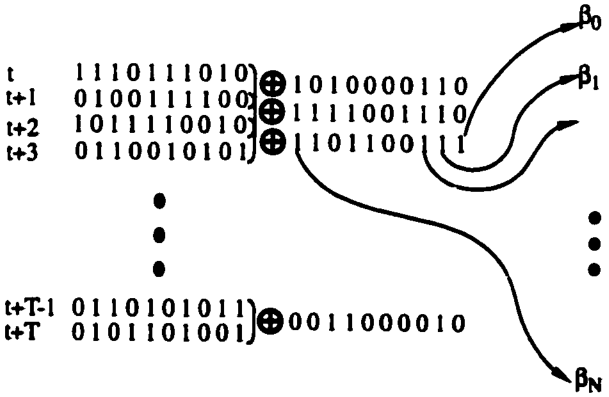


Figure 6.1: Finding transitional bits in the LR method.

every input vector transition, the I_j and O_k values along with the dissipated power (or effective capacitance) per cycle are recorded. Note that I_j and O_k are binary numbers and are stored in a matrix A whose ith row has $1, I_1^i, I_2^i, \ldots, I_N^i, O_1^i, O_2^i, \ldots, O_M^i$. The power values are stored in a vector P. At the end of the simulation, we have the linear system of equations:

$$P = A\beta \tag{6.4}$$

where P is a vector of size [*sample size* x 1], A is a matrix of size [*sample size* x $(M+N+1)$], and β is a vector of size $[(N+M+1) \times 1]$. To overdefine the system, the *sample size* must be much larger than $(M+N+1)$. By using least-square fitting techniques to solve the above system, EQ 6.3 is derived.

It was noted in [101] that the least square fitting always produces an estimate of P with the same average value as the average value of power in the sample employed for fitting. This will guarantee that the LR model will perform at least as good as an average power approximation using the same input sample. In this case, β_0 in EQ 6.3 approaches the average value of the sample with the rest of the coefficients being (additive/subtractive) correction terms. However, If the module is used in a design environment with input patterns having behavior different than the characterization patterns, then adding β_0 to the total power every clock cycle can significantly degrades the estimation accuracy. Note that in [101], a uniformly distributed and independent input patterns was used for model building.

In this work, we propose two important modifications to the LR approach. First, we significantly improves the accuracy of the model by introducing a piece-wise linear regression technique based on the number of transitional input bits. Second, we propose a Monte-Carlo based simulation approach to limit the size of the sample needed to extract the power coefficients. Limiting the sample size is critical in order to reduce the characterization time especially if accurate (and more versatile) transistor-level simulators are to be used to measure the power.

Before concluding this section, it is worth noting that due to the increased demand for architectural-level power estimation, a commercial tool, called Watt-Watcher-Architect [128] from Senté Inc., is now available in the market. WattWatcher is claimed to have an accuracy within 25% of gate-level tools with a speed of 34 minutes for a circuit of 120,000 gates.

6.4 Effective Capacitance Macro-Modelling

In a static MOS circuit, it was observed experimentally that C_{eff}correlates well with input and output bits activities[1]. Several simulations were performed to verify this observation. One result is shown in Fig. 6.2 for a 5x5 array multiplier ($N = M = 10$). The figure gives two curves: the upper curve plots C_{eff} as a function of the total number of transitional input bits, λ_{in}, which in this case can range from 1 to 10, and the lower curve plots C_{eff} as a function of the total number of transitional output bits, λ_{out}, (which also varies between 1 to 10). Each point on every curve represents the total switched capacitance averaged over a very large number of randomly generated input cycles. In each cycle, a particular λ_{in} or λ_{out} is satisfied as shown in the onsets of Fig 6.2. A transistor-level power simulation tool, PowerMill [104], was used for capacitance extraction.

Fig 6.2 clearly illustrates the strong dependency between C_{eff} and λ_{in} or λ_{out}. In addition, it shows that a large difference in C_{eff} (up to 3 times in this case) can result

[1] we will limit our discussion now on modules with no control inputs. Extension of the approach for multi-functional modules is discussed at the end of this section.

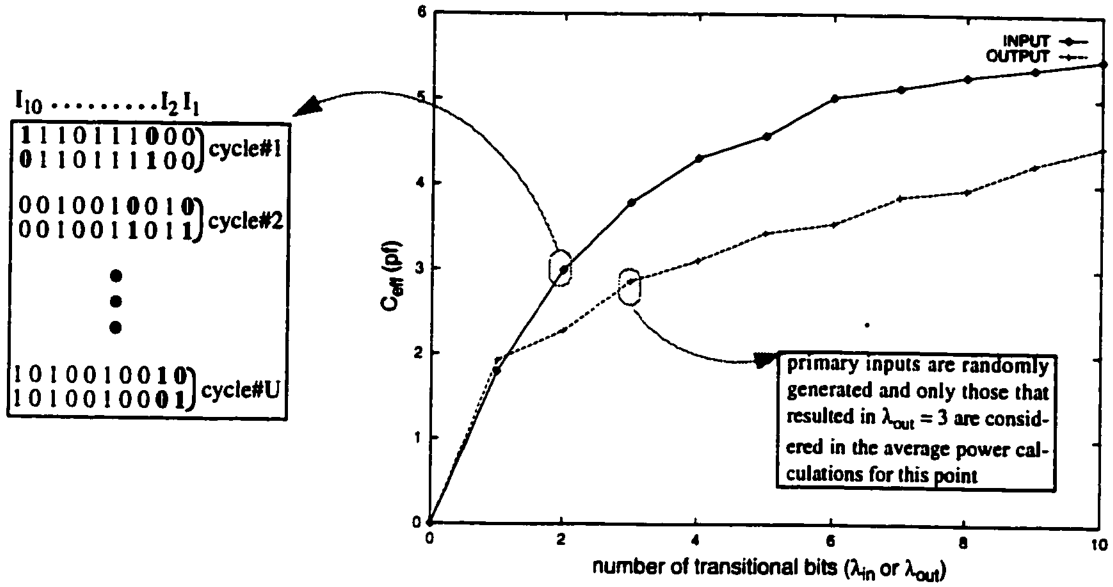


Figure 6.2: Dependence of C_{eff} on the number of transitional input and output bits for a 5x5 (N=10, M= 10) array multiplier.

depending on the number of transitional bits. Similar behaviors were also noticed for other data-path modules with various sizes. Although this correlation is not necessarily linear, it gives a good *validation* to the simple LR approach [101]. Moreover, the LR approach gives more details than Fig 6.2 since it accounts for *individual* input and output bit activities.

As pointed out in the previous section, the LR technique can be largely inaccurate if employed to estimate the power when highly correlated input data sequence is used. The reason for this is that the least-square fitting solution will produce a β_0 term that converges to the average value of the random sample used for fitting. Adding this term for every input vector transition will greatly over estimate the actual power.

In order to overcome this problem, we propose a modification to the LR approach (henceforth called Modified LR, or shortly MLR). The basic idea is to generate a number of LR equations, one for each possible value of λ_{in} (1,2,...,N). For each case, a random sample of input vectors is generated such as exactly λ_{in} bits are allowed to flip between each consecutive input vectors (as in Fig. 6.2). The module is simulated N times using this modified set of samples and for each run, the least-square fitting technique is used as before to produce the corresponding LR model. Thus at the end of the characterization process, the following set of equations is obtained:

$$\begin{aligned}
C_{eff}^{1} &= \beta_0^1 + \beta_1^1 I_1 + \beta_2^1 I_2 + \ldots + \beta_N^1 I_N + \beta_{N+1}^1 O_1 + \beta_{N+2}^1 O_2 + \ldots + \beta_{N+M}^1 O_M \\
C_{eff}^{2} &= \beta_0^2 + \beta_1^2 I_1 + \beta_2^2 I_2 + \ldots + \beta_N^2 I_N + \beta_{N+1}^2 O_1 + \beta_{N+2}^2 O_2 + \ldots + \beta_{N+M}^2 O_M \\
&\vdots \\
C_{eff}^{N} &= \beta_0^N + \beta_1^N I_1 + \beta_2^N I_2 + \ldots + \beta_N^N I_N + \beta_{N+1}^N O_1 + \beta_{N+2}^N O_2 + \ldots + \beta_{N+M}^N O_M
\end{aligned} \tag{6.5}$$

The rationale behind this approach is that C_{eff} has a direct relationship with λ_{in} and therefore it is intuitively more accurate to use this dependance to construct piecewise linear models. In this case, the minimum mean square error obtained from fitting is not global (over all possible values of λ_{in}) but rather local (over a single value of λ_{in}). As a result, the constant $\beta_0^{\lambda_{in}}$ will be restricted to the average capacitance over a single value of λ_{in}. Note that it is easier to generate random vectors that confine to a particular λ_{in} value rather than λ_{out} which is why the former parameter is used for the MLR model construction.

The main limitation of the MLR scheme is that it requires long characterization time. Instead of performing one simulation using a large sample, we need to execute N simulations one for each value of λ_{in}. This can take very long time, especially if transistor-level tools are used for extracting the switched capacitance. To handle this problem, we use a Monte Carlo based scheme to execute each simulation. Monte Carlo simulation was previously used to find the average power of a module using random/user-specified input data [130]. Instead, we employ it here in the characterization process to limit the number of cycles required to derive each of the above equations.

Let $C_T^{\lambda_{in}}$ represents the effective capacitance switched when the module is stimulated with T input vectors satisfying a given λ_{in} value. Assume for now that $C_T^{\lambda_{in}}$ have a normal distribution with unknown mean μ. For each λ_{in}, the size of the characterization sample is determined by the number of input vectors needed to obtain a close estimate of μ. This will allow us to say with some confidence when the sample size is large enough to stop the simulation and so cut down the characterization time.

Each simulation is executed as follows. For a given λ_{in}, the module is simulated for T cycles (using a transistor-level simulator) where in each cycle, a randomly generated pair of consecutive input vectors that satisfy λ_{in} is used. Using random vectors ensures that the characterization procedure is not biased for a specific input data. This process is repeated (independently) R times, and hence a random sample of size R is created. The sample mean can be found by:

$$C_m^{\lambda_{in}} = \frac{\sum_R \left(\sum_{i=1}^{T} C_i^{\lambda_{in}} \right) / T}{R} \tag{6.6}$$

Using the above normality assumption of $C_T^{\lambda_{in}}$, it can be shown that $\frac{\sqrt{R}\left(C_m^{\lambda_{in}} - \mu \right)}{s}$ has a t-distribution with (R-1) degrees of freedom, where (s) is the sample standard deviation [129]. This allows us to write the following probability inequality:

$$P\left\{\left|\frac{\sqrt{R}\left(C_m^{\lambda_{in}} - \mu\right)}{s}\right| < t_{\xi/2}\right\} = 1 - \xi \tag{6.7}$$

where $t_{\xi/2}$ is obtained from a t-distribution with (R-1) degrees of freedom. From EQ 6.7, we can say that we have (1-ξ)x100% confidence that $\left|\frac{\sqrt{R}\left(C_m^{\lambda_{in}} - \mu\right)}{s}\right| < t_{\xi/2}$, or by rewriting:

$$\frac{\left|C_m^{\lambda_{in}} - \mu\right|}{C_m^{\lambda_{in}}} < \frac{t_{\xi/2}s}{C_m^{\lambda_{in}}\sqrt{R}} \tag{6.8}$$

if we let $\varepsilon = \frac{\left|C_m^{\lambda_{in}} - \mu\right|}{C_m^{\lambda_{in}}}$, then for a desired percentage error, ε, in the effective capacitance estimate and for a given confidence level, (1-ξ), the circuit must be simulated until:

$$\frac{t_{\xi/2}s}{C_m^{\lambda_{in}}\sqrt{R}} > \varepsilon \tag{6.9}$$

which sets a lower limit on R given by:

$$R \geq \left(\frac{t_{\xi/2}s}{\varepsilon C_m^{\lambda_{in}}}\right)^2 \tag{6.10}$$

Note that both ε and ξ can be set by the user to trade-off between characterization time and accuracy. Further notice that exactly 2xTxR input vectors are used for a given λ_{in}. During characterization, the I and O values of each vector pair are stored along with their switched capacitance and at the end of the simulation are used to derive the corresponding LR equation[1].

The above procedure assumes that a large fixed value of T is used for all runs. On the other hand, R is not constant and changes depending on (other than the user-specified ε and ξ parameters): (1) the sample standard deviation (s), and (2) the magnitude of $C_m^{\lambda_{in}}$. In general, it can be observed that for small λ_{in} (highly correlated input patterns), s tends to be high since the effective capacitance is greatly affected not only by λ_{in} but also by the state of the non-switching primary input bits. In addition, the magnitude of the effective capacitance itself is the very small since not too many bits are switching. Both these factors force R to be large as given by EQ 6.10. As λ_{in} gets larger, s becomes smaller and $C_m^{\lambda_{in}}$ rises, decreasing R as a result. Thus, for each value of λ_{in}, this technique automatically adjusts the sample size to enable convergence and cuts the characterization time. As an example, Fig 6.3 shows the number of cycles required to characterize a 10x10 array multiplier. Notice how the number needed decreases as λ_{in} gets larger.

The use of a Monte Carlo based technique in the characterization process has the implicit assumption that the effective capacitance has a normal (Gaussian) distribution. We checked the normality hypothesis by plotting the distribution of $C_T^{\lambda_{in}}$ obtained from several modules. A typical example is shown in Fig. 6.3 for a 5x5 array multiplier. The bell-like curve for each value is similar to that of a normal distribution. Notice that for

[1] Another possible way which cuts the characterization time in half is to use the destination vector of each pair as the starting vector of the next pair. Thus, the total number of vectors simulated will be $(T+1)\times R$. The above approach allows more randomness, however, since in the later case the subsequently generated vectors will depend on the initial one.

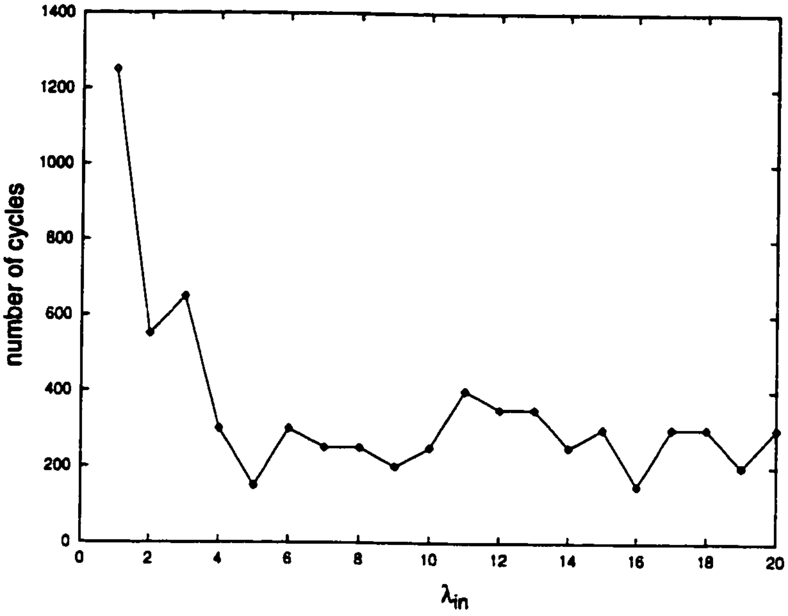


Figure 6.3: Number of cycles required to characterize a 10x10 array multiplier (λ_{in} = 1,2,...,20).

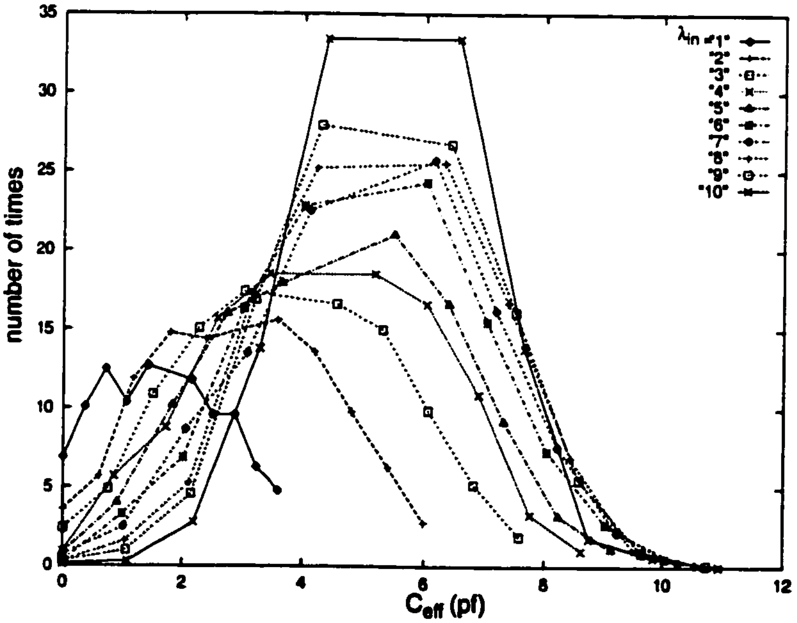


Figure 6.4: Bell-like distribution of C_{eff} for a 5x5 array multiplier (λ_{in}= 1,2,...,10).

small values of λ_{in} the curves are more distorted and wider than those for larger ones, which is one of the reasons why the Monte Carlo technique requires more iterations for the former cases[1].

A summary of our characterization approach is shown in Fig 6.5. This procedure is only valid for uni-functional modules. For multi-functional units (an ALU for example), the steps have to be repeated for every possible setting at the control inputs. Thus, a unique set of linear regression equations must be computed for each distinct function that the module may be called on to execute. During power estimation, the control inputs are used to select among the different regression functions.

During architectural-level power estimation, the MLR macro-model can be used with a typical/user-specified input data. For each pair of consecutive input vectors, the I and O values are found as well as λ_{in} to select the corresponding LR equation for power calculation. At the end of the simulation, all cycle capacitances are added up and averaged over the simulation period to obtain C_{eff} which if multiplied by $V_{dd}^2 f$ gives the total power.

6.5 Results

We tested the proposed methodology on a set of data-path circuits: a 16-bit Ripple Carry Adder (RCA), 15x15 Array Multiplier (AM), a 16-bit Conditional Sum Adder (CSA), a 6x6 modified Booth Multiplier (BM), a 16-bit Magnitude Comparator (COMP), and an 8-by-4 Divider (DIV). Some of these modules were implemented using static CMOS circuit styles while others were designed using Complementary Pass-transistor Logic

[1] The normality observation is not assumed to be general but we found it to valid across several data-path modules.

(CPL)[1]. Each circuit was laid out in a 3-metal 0.8μm CMOS technology. Layout extraction was performed to include parasitic capacitance from interconnect metals, and the netlist was translated into PowerMill format.

The capacitance characterization procedure given in Fig. 6.5 was implemented in the *C* programming language and was applied on each benchmark circuit. The least-square fitter in MatLab was used for model generation. The following parameters were

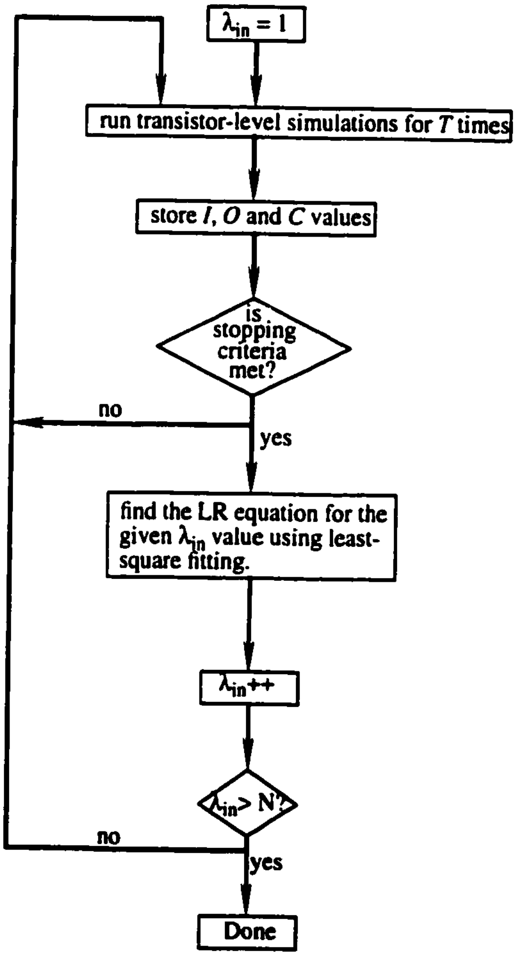

λ_{in} = 1
run transistor-level simulations for *T* times
store *I*, *O* and *C* values
is stopping criteria met?
no
yes
find the LR equation for the given λ_{in} value using least-square fitting.
λ_{in}++
λ_{in}> N?
no
yes
Done

Figure 6.5: Summary of the MLR characterization technique.

[1] see Section 2.2.3 for a CPL example.

utilized for the Monte Carlo simulations: $\varepsilon = 0.10$ (that is, maximum percentage error of 10%), $\xi = 0.01$ (that is, 99% confidence level), and $T = 50$ cycles. For comparisons, we also simulated each circuit using a large sample with random vectors and derived a *single* LR equation (as in [101]).

To test the performance of the proposed approach, two long data streams were employed: (1) **Random** with average switching activity per primary input line of 0.5, and (2) **Correlated** where the average switching activity per bit was set to 0.15 and random correlations between different bits were generated using the state lines of an up/down counter. Using the outputs of a digital counter to generate highly correlated input data is a common practice in the literature [131]. Notice that the way the test streams were created is different from the method used for characterization. Table 6.1 gives a summary of the main features of the test circuits and also presents the accurate effective capacitance values obtained from PowerMill for each data stream. It is worth noting here the large difference in C_{eff} obtained using the random sequence and the correlated one.

Table 6.1: Characteristics of modules and their C_{eff} values from PowerMill.

Module Name	**# of input bits (N)**	**# of output bits (M)**	**Implementation Style**	**Random (pf)**	**Correlated (pf)**
RCA_{16}	33	17	CMOS	12.8	3.3
AM_{15x15}	30	30	CPL	119.4	33.5
CSA_{16}	33	17	CPL	3.9	0.96
BM_{6x6}	12	12	CPL	26.1	18.3
$COMP_{16}$	32	2	CMOS	1.2	0.30
$DIV_{8\text{-}by\text{-}4}$	12	5	CPL	15.8	12.2

Comparisons between the MLR approach and the LR technique are shown in Table 6.2. The measure used is the relative error on the average defined as:

$$\%error = \left|\frac{PowerMill - Model}{PowerMill}\right| \times 100 \qquad (6.11)$$

Both LR and MLR shows good performance when the input data is random (LR is slightly better). For correlated data, MLR is superior to LR and has an average error within 15% from PowerMill. Both approaches produced large errors per cycle and so neither is suitable for cycle-based RTL power estimation. Cycle-based accuracy might not be crucial at this level of the design since to compare two design alternatives the cumulative power is more important than the instantaneous one.

Table 6.2: Summary of power estimation results.

Module Name	**LR (% error)**		**MLR (% error)**	
	Random	**Correlated**	**Random**	**Correlated**
RCA_{16}	1.2%	80%	1.6%	11.2%
AM_{15x15}	0.0%	146%	0.9%	7%
CSA_{16}	1.7%	61%	1.2%	2.3%
BM_{6x6}	0.0%	35%	1.6%	14%
$COMP_{16}$	0.8%	130%	2.6%	34%
$DIV_{8\text{-}by\text{-}4}$	0.1%	13.2%	0.4%	13%
Average	**0.63%**	**77.3%**	**1.4%**	**13.6%**

Both techniques performed *power estimation* very quickly (less than 3.0 CPU seconds for the largest circuit on a Sparc 20 station with 240M memory). LR is slightly faster since the coefficients of only one equation has to be looked up during power estimation (as compared to selecting one out of N equations and then looking up its coefficients for MLR). In [101], the amount of time needed to *characterize* a module is not given. Our approach requires about 8000 PowerMill cycles for the largest circuit. Of course, much longer time would have been required if the Monte-Carlo simulation technique was not used. If accurate gate-level power simulation tools are used for characterization, this time can be made much smaller. However, not all circuits are gate-level representable, and so in such cases transistor-level simulation is the only alternative.

6.6 Summary

In this chapter, a new procedure to characterize the effective capacitance in data-path components was described. The technique is based on piece-wise linear regression modelling using the number of transitional input bits as the split criteria. We also employ a Monte-Carlo simulation scheme to cut-down the characterization time. Our approach significantly improves the estimation accuracy as compared to simple LR approach presented in [101] with similar estimation time.

Chapter 7

Conclusions and Recommendations

For many years, chip area and speed have been the conventional measures of quality in a VLSI design. Only over the last decade or so has power becomes a crucial design criterion. On one hand, high-performance systems with increased packing density have led to power-related reliability problems and expensive cooling/packaging requirements. Another motivation for low-power is the extraordinary growth in portable electronics such as cellular phones and portable computers. These devices are battery-operated and therefore must be extremely economical with their energy consumption. The above factors have set strict limitations on power dissipation and forced design engineers to look for solutions on all fronts: process, circuits, gates, architectures and systems.

This dissertation presented novel circuit techniques and methodologies pertaining to low-power CMOS VLSI design. The proposed schemes achieve the main objective of low-power with similar or improved performance as compared to traditional design approaches. The effect on area is also addressed and the most feasible ways to reduce power with similar or smaller area overhead are investigated.

The work presented in this thesis can be broadly divided into two main parts:

1) new low-swing circuit techniques based on charge sharing and current injection. These techniques were used to reduce the power consumption and improve the speed of: (a) dynamic CMOS circuits with high capacitive nodes such as internal bus lines, bit-lines in ROMs, and match lines in CAMs, and (b) the write and read operations in multi-port SRAMs. For this part, four chips were designed, fabricated and tested to verify the functionality and performance of the proposed schemes.

2) new low-power design methodologies at the gate and architecture levels. These methodologies target: (a) high-speed and low-power/low voltage logic design, and (b) power marco-modelling for efficient high-level power estimation. For this part, two pieces of software were written to implement the proposed methodologies and the later were tested using some benchmarks circuits.

The main contributions of this thesis were described in Chapters 3-6 and are summarized below:

- Chapter 3 reported a new circuit technique for low-power and high-performance design of a class of dynamic digital circuits with high capacitive nodes. The basic idea is to create a low swing on these nodes using the principle of charge sharing. A new model (called HiCapCS) to create this swing very quickly, without any DC power dissipation or the need for pulsed operation or reference voltage generator was described. An added advantage of HiCapCS is reducing the supply line noise. Three applications of this technique; namely match lines in CAMs, bit-lines in ROMs and internal bus interconnect lines, were described. HSPICE simulations of the extracted circuits show significant improvements in speed and energy as compared to conventional approaches. The functionality and performance of the proposed method were verified through three chips: a 2-bit internal bus, a 4Kx16b ROM, and a 32x32 TLB. Measurement results of these chips down to 1.2V were presented.

- in Chapter 4, a new method for fast and low-power memory write operation was proposed. The basic idea is to initially bias the back-to-back CMOS inverters of the memory cell in the transient region to increase their voltage gain. Then by creating an extremely small swing on the bit-lines, a differential current is injected into the cell. This current difference will in turn be converted to full CMOS voltage levels by the memory cell which exhibits a high gain. The proposed method, called Current-Injection Write (CIW), comes at a small increase in the cell area. However, we showed that by using this technique in multi-port SRAMs, the area penalty is actually reduced. Furthermore, CIW allows no degradation in the cell noise immunity. In addition to the write circuit, a sense amplifier for low-power and high-speed memory reads was also described. This amplifier works in a similar fashion as the CIW circuitry. These techniques were used in the design of a 3-port, single-phase clocked, 32x64b register file fabricated in a standard 3-metal 0.35μm CMOS process. Measurements of the file chip have confirmed the feasibility of the approach.

- Chapter 5 presented a comprehensive summary of low-voltage/low-threshold design techniques. The pros and cons of several dynamic and static approaches were discussed. Also, a study on the effect of supply voltage scaling on delay, leakage power, and delay sensitivity was presented. Based on the above discussions, a new approach, called DVDV, was proposed as a means for low-power/low-voltage high-performance design of submicron CMOS logic circuits. The DVDV scheme adds minor alterations to current design trends and technologies. The main idea is to utilize a library of gates having dual supply voltages and dual threshold voltages, hence the name DVDV, to achieve high-speed at the lowest possible dynamic and leakage power dissipation. A simple algorithm to implement the DVDV approach was developed and implemented under the Berkeley's SIS-1.2 environment. Application to benchmark circuits (using a 0.25μm process) show larger power savings as compared to using two supply voltages (and a single V_{th}), and faster speeds than could be achieved utilizing two threshold voltages (and a single V_{dd}).

- in Chapter 6, a simple, yet accurate, method to characterize the effective capacitance in data-path macros was proposed. The resultant macro-models can be used to quickly estimate the power at the architectural-level. The capacitance models are based on linear regression and a transistor-level tool was employed for capacitance extraction. The characterization methodology assumes no specific word-level statistics of the input data, requires little knowledge about the internal structure of the modules, allows the user to trade-off accuracy and characterization time, and propagates effective capacitance directly from transistor-level (real) implementations. Simulation experiments on a set of data-path components with various sizes were performed. Compared to a previously published approach, the proposed scheme significantly improved the accuracy of RTL power estimation and produced results within 15% from a transistor-level tool on the average.

As with most research efforts, this dissertation possibly raised as many questions as it solved. The limited time allocated for this research prohibited further investigations of these problems. However, the ideas presented here should provide a strong foundation upon which to build. The following discussion will point out the limitations of the techniques presented in the previous chapters and suggest improvements that could be made to them.

For the HiCapCS scheme, new applications that can benefit from its low-power and high-performance features should be investigated. An example would be Programmable Logic Arrays (PLAs), which similar to ROMs, have highly capacitive bit-lines.

Also common to all HiCapCS applications described in Chapter 3, a Sense Amplifier Enable (*SAE*) signal has to be generated in order to turn on the single-/double-ended sense amplifiers. In the current implementation, this signal is generated by simply delaying the clock signal using a set of cascaded inverters. More reliable circuit techniques to produce this signal while tracking worst/best process variations should be developed. Finally, the dynamic and low-swing nature of HiCapCS puts a lower limit on how small the threshold voltage can be scaled in circuits utilizing this approach. More analysis is needed to check the necessary adjustments to HiCapCS under the low V_{dd}/low V_{th} regime with increased leakage current.

Extensions to the CIW scheme include employing it in a general (column-decoded) SRAM. As mentioned in Chapter 4, this could be done by using multiple write word-lines. The effect on area and layout under this scenario need to be investigated. Also in contemporary register files, a desirable feature is to allow read-after-write of the same word in the same cycle. In our register file, this can be accomplished for example by delaying the *SAE* signal. Other more versatile circuit techniques to allow this operation should be explored.

It is our belief that the DVDV design methodology, in which all legal dual V_{dd} and dual V_{th} are considered in the optimization search space, addresses a new design paradigm which no other research effort has previously examined. However, the current solution using DFS is rather simple and greedy (looking at one gate at a time). Perhaps there is someway to take advantage of the problem structure in order to generate better results. Other stronger solutions using for example mathematical/dynamic programming or iterative improvement techniques with dynamic evaluation of the cost (power) function should be studied. The last point will allow the inclusion of the switching activity (α) and the activity factor (A) in the mapping decisions taken by the algorithm. Other enhancements to the technique include:

- improving the initial implementation of the algorithm through better data structures and selective timing information updating. In the current implementation, the later is performed globally after each permissible move, which accounts for a significant portion of the total running time.

- better modelling of the delay and power. Currently this is done using the simple models in SIS which have limited accuracy. For example, power estimation is performed assuming uncorrelated primary inputs with each having 0.5 signal probability. More accurate results can be obtained by using switch-level tools such as PathMill (for delay) and PowerMill (for power).

- including the delay sensitivity to V_{dd} scaling in the optimization process. As explained in Chapter 6, this is becoming more important and the availability of different types of cells in the DVDV approach gives us the flexibility to sacrifice more power for less sensitivity.

- simultaneous gate sizing and DVDV optimizations. The current implementation utilizes a small library with a few cells. Using a larger library with various cells and different sizes per cell can possibly allow lower power. For example, wider (and so faster) gates can be operated at LV_{dd} to reduce dynamic power. A recent work has addressed this problem using a single supply voltage [132].

- also related to the above bullet, the used library has no detailed layout/interconnect information, which can affect the delay and power estimates.

- it would be interesting to stretch the technique using multiple V_{dd}'s and V_{th}'s and see if the resultant gain (if any) warrants this extension, and

- finally, applying the DVDV methodology at the architectural (block) level might be advantageous too. Previous efforts utilizing multiple V_{dd}'s [32] or dynamically variable V_{th} [41] at the RT level have shown significant (dynamic or leakage) power savings than using conventional single V_{dd} or single V_{th} schemes, respectively.

Future work of the MLR power characterization scheme should focus on improving its accuracy as well as the characterization time. The former can be achieved for example by partially accounting for the type of transitions at the primary input bits as well as the state of non-transitional bits. The distorted distributions of C_{eff} for small λ_{in} values suggest that these modifications should be particularly applied to these cases. Further improvements are also possible by increasing the number of characterization vectors utilized for those C_{eff}'s with large standard deviation (by using smaller ε, for example). Reducing the characterization time can be done, for example, by creating equivalent sets of patterns with "nearby" λ_{in} values to reduce the number of simulations and linear regressions that have to be performed.

Publications

[1] Muhammad M. Khellah and M. I. Elmasry, "Low-Power High-Performance Design of Deep Sub-micron CMOS Circuits Using a Dual-V_{dd} Dual-V_{th} Approach", *accepted for publication in the 1999 IEEE Symposium on Low-Power Electronics and Design*, San Diego, CA, Aug. 16-17, 1999.

[2] Muhammad M. Khellah and M. I. Elmasry, "Low-Power Design of High-Capacitive CMOS Circuits Using a New Charge Sharing Scheme," *Proc. of the IEEE International Solid-State Circuits Conference*, pp. 284-285, San Francisco, CA, Feb. 15-17, 1999.

[3] Muhammad M. Khellah and M. I. Elmasry, "Circuit Techniques for High-Performance and Low-Power Multi-port SRAMs", *Proc. of the IEEE International ASIC Conference*, pp. 157-161, Rochester, NY, Sept. 13-16, 1998.

[4] Muhammad M. Khellah and M. I. Elmasry, "Use of Charge Sharing to Reduce Energy Consumption in Wide Fan-in Gates," *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. II9-II12, Monterey, CA, May 31 - June 3, 1998.

[5] Muhammad M. Khellah and M. I. Elmasry "Effective Capacitance Macro-Modelling for Architectural-Level Power Estimation," *Proc. of the IEEE Great Lakes Symposium on VLSI*, pp. 414-419, Lafayette, Louisiana, Feb. 19-21, 1998.

[6] Muhammad M. Khellah and M. I. Elmasry, "HiCapCs: A Novel Charge Sharing Approach for Low-Power Design of High-Capacitive CMOS Circuits," *submitted for publication in the IEEE Journal of Solid-State Circuits.*

[7] Muhammad M. Khellah and M. I. Elmasry, "Current-Mode Circuit Techniques for Low-Power High-Performance Multi-port SRAM Design," *submitted for publication in the IEEE Transactions on VLSI Systems (special issue on low-power).*

Bibliography

[1] J. Rabaey and M. Pedram, "Low-Power Design Methodologies," Kluwer Academic Publisher, 1996.

[2] J. Meindl, "Low-Power Micro-electronics: Retrospect and Prospect," *Proc. of the IEEE*, Vol. 83, No. 4, pp. 619-635, April 1995.

[3] D. Singh, et. al., "Power Conscious CAD Tools and Methodologies: A Perspective," *Proc. of the IEEE*, pp. 570-594, April 1995.

[4] P. K. Charttejee and G. Larrabee, "Gigabit age microelectronics and their manufacture," *IEEE Trans. on VLSI Systems*, vol. 1, pp. 7-21, 1993.

[5] R. Swanson and J. Meindl, "Ion-implanted Complementary MOS transistors in low-voltage circuits," *IEEE Journal of Solid-State Circuits*, Vol. SC-7, pp. 146-152, April 1972.

[6] A. Chandrakasan, and R. Brodersen, "Low-Power Digital CMOS Design,", Kluwer Academic Publishers, 1995.

[7] H. Veendrick, "Short-circuit Power Dissipation of Static CMOS Circuitry and Its Impact on the Design Buffer Circuits," *IEEE Journal on Solid State Circuits*, v. 19, no. 8, pp. 468-473, 1994.

[8] A. Bellaouar and M. I. Elmasry, "Low-Power Digital VLSI Design: Circuits and Systems," Kluwer Academic Publishers, 1995.

[9] B. Davari. et. al., "CMOS Scaling for High-Performance and Low-Power - The Next 10 Years," *Proceedings of the IEEE*, pp. 595-606, April 1995.

[10] M. Nandakumar, et. al. "A Device Design Study of a 0.25μm Gate Length CMOS for 1V Low-Power Applications," *Proc. of the 1995 Symp. on VLSI Technology*, pp. 80-81.

[11] G. Shahidi, et. al. "Device and Circuit Design Issues in SOI Technology," *Proc. of the IEEE Custom Integrated Circuits Conference*, pp. 339-346, San Diego, CA, May 1999.

[12] K. Chao, and F. Wong," Low-Power Consideration in Floorplan Design," *Proc. of the 1994 International Workshop on Low-Power Design*, pp. 45-50, 1994.

[13] H. Vaishnav, and M. Pedram, "Delay Optimal Partitioning Targeting Low-Power VLSI Circuits," *Proc. of the IEEE International Conference on Computer Aided Design*, pp. 638-643, 1995.

[14] H. Vaishnav, "Optimization of Post-Layout Area, Delay, and Power Dissipation," *Ph.D Thesis, Computer Engineering Dept., Univ. of Southern California*, 1995.

[15] J. Cong, C-K Koh, K-S Leung, "Simultaneous Driver and Wire Sizing For Performance and Wire Optimization," *IEEE Trans. on VLSI Systems*, v. 2, no. 4, pp. 408-425, December 1994.

[16] J. Xi, and W-M Dai, "Buffer Insertion and Sizing Under Process Variation for Low-Power," *Proc. of the 32nd Design Automation Conference*, pp. 491-496, 1995.

[17] S. Thompson, et. al. "Dual Threshold Voltages and Substrate Bias: Keys to High-Performance, Low-Power, 0.1mm Logic Designs," *Proc. of the 1997 Symp. on VLSI Technology*, pp. 69-70, 1997.

[18] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, "A 3.8-ns CMOS 16x16-b Multiplier Using Complementary Pass-Transistor Logic," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 2, April 1990.

[19] R. Zimmermann, and W. Fichtner, "Low-Power Logic Styles: CMOS Versus Pass-Transistor Logic," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 7, April 1990.

[20] B-S. Kong, et. al. "Asynchronous Sense Differential Logic," *Proc. of the IEEE Solid-State Circuits Conference*, pp. 284-285, San Francisco, CA, Feb. 1999.

[21] C. Svensson, and A. Alvandpour "Low-Power and Low-Voltage CMOS Digital Circuit Techniques," *Proc. of the International Symp. on Low-Power Electronics and Design*, pp. 7-10, August, 1998.

[22] P. Ng, et. al., "Performance of CMOS Differential Circuits," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 6, June 1996.

[23] S. Iman and M. Pedram, "Multi-Level Network Optimization for Low-Power," *Proc. of the International Conference on Computer Aided Design*, pp. 371-377, November 1994.

[24] K. Roy and S. Parasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations," *IEEE Trans. on VLSI Systems,* Vol. 1, No. 4, Dec. 1993.

[25] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization," *IEEE Transactions on VLSI Systems,* Vol. 5, No. 4, December 1997.

[26] C. Tsui, M. Pedram, C. Chen, and A. Despain, "Low Power State Assignment Targeting Two- and Multi-level Logic Implementations," *Proc. of the International Conference on Computer Aided Design*, pp. 82-87, November 1994.

[27] J. Monterio, S. Devadas, and A. Ghosh, "Retiming Sequential Circuits for Low Power," *Proc. of the International Conference on Computer Aided Design*, pp. 398-402, November 1993.

[28] L. Benini, P. Sigel, and G. Micheli, "Saving Power by Synthesizing Gated Clocks for Sequential Circuits," *IEEE Design and Test of Computers Magazine*, pp. 32-41, Winter 1994.

[29] M. Alidina, J. Monterio, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-Based Sequential Logic Optimization for Low-Power," *IEEE Transactions on VLSI Systems*, Vol. 2, No. 4, December 1994.

[30] A. Chandrakasan, R. Allmon. A. Stratakos, and R. Brodersen, "HYPER-LP: A system for Power Minimization Using Architectural Transformations," *Proc. of the International Conference on Computer Aided Design*, pp. 300-303, 1992.

[31] A. Raghunathan, S. Dey, and N. K. Jha, "Glitch Analysis and Reduction in Register Transfer Level Power Optimization," *33rd Design Automation Conference*, pp. 331-336, June 1996.

[32] J-M. Cheng, and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *Proc. of the International Symposium on Low-Power Electronics and Design*, pp. 157-162, Aug. 1996.

[33] M. Lee and V. Tiwari, "A Memory Allocation Technique for Low-Energy Embedded DSP Software," *Proc. of the IEEE Symposium on Low Power Electronics*, pp. 24-25, 1995.

[34] C. Gebotys, "Low-Energy Memory Component Design for Cost-Sensitive High-Performance Embedded Systems," *Proc. of the 1999 Custom Integrated Circuits Conference*, pp. 397-400, 1996.

[35] C. Papachristou, M. Spining, and M. Nourani, "A Multiple Clocking Scheme for Low Power RTL Design," *Proc. of the International Symposium on Low Power Design*, pp. 27-32, April 1995.

[36] C. Gebotys, "An ILP model for Simultaneous Scheduling and Partitioning for Low Power System Mapping," *to appear in special issue on low power in VLSI Design Journal*, 1996.

[37] A. Raghunathan and N. Jha, "An Iterative Improvement Algorithm for Low Power Data Path Synthesis," *Proc. of the International Conference on Computer Design*, pp. 597-602, Nov. 1995.

[38] J-M. Chang and M. Pedram, "Low Power Register Allocation and Binding," *Proc. of the 32nd Design Automation Conference*, pp. 29-35, June 1995.

[39] C-L. Su, et. al., "Low-Power Design and Compilation Techniques for High-Performance Processors," *IEEE Trans. on Signal Processing*, Vol. 41, no. 2, pp. 901-905, Feb. 1994.

[40] G. Tellez, et. al., "Activity-Driven Clock Design for Low-Power Circuits," *Proc. of the IEEE International Conference on Computer-Aided Design*, pp. 62-65, 1995.

[41] A. Chandrakasan, et. al. "Design Considerations and Tools for Low-Voltage Digital System Design," *Proc. of the 33rd Design Automation Conference*, Las Vegas, NV, June 1996.

[42] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data-Driven Signal Processing: An Approach for Energy Efficient Computing," *Proc. of the International Symposium on Low Power Electronics and Design*, pp. 347-352, August 1996.

[43] M. Hiraki, H. Kojima, and H. Misawa," Data-Dependent Logic Swing Internal Bus Architecture for Ultra-Low-Power ULSIs," *Proc. of the 1994 Symposium on VLSI Circuits*, pp. 29-30.

[44] Mireca Stan, and W. Burleson "Low-Power Encodings for Global Communication in CMOS VLSI," *IEEE Trans. on VLSI Systems*, Vol. 5, No. 4, Dec. 1997.

[45] T. Kobayashi, K. Nogami, T. Shirotori, and Y. Fujimoto, "A Current-Controlled Latch Sense Amplifier and a Static Power-Saving Input Buffer for Low-Power Architecture," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 4, April 1993.

[46] J. L. Hennessy, and D. A. Patterson, "Computer Architecture: A Quantitative Approach", Morgan Kaufmann Publishers, 1990.

[47] D. A. Hodges, and H. G. Jackson, "Analysis and Design of Digital Integrated Circuits," 2nd Edition, McGraw-Hill Publishing Company, 1988.

[48] H. Takahashi, S. Muramatsu, and M. Itoigawa, "A New Contact Programming ROM Architecture for Digital Signal Processor" *Proc. of the 1998 Symposium on VLSI Circuits*, pp. 158-161, June 1998.

[49] E. D. Angel and E. E. Swartzlander, "Survey of Low-Power Techniques for ROMs," *Proc. of the 1997 International Symposium on Low-Power Electronics and Design*, pp. 7-11, August, 1997.

[50] K. J. Schuktz, and P. G. Gulak, "Fully Parallel Integrated CAM/RAM Using Preclassification to Enable Large Capacities," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 5, pp. 689-699, May 1996.

[51] M. Motomura, et. al., "A 1.2-Million Transistor, 33-MHz, 20-b Dictionary Search Processor (DISP) ULSI with a 160-kb CAM," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 5, pp. 1158-1165, October 1990.

[52] T. Ogura, et. al., "A 336-kbit Content Addrerssable Memory for Highly Parallel Image Processing," *Proc. of the IEEE Custom Integrated Circuit Conference*, pp. 273-276, May 1996.

[53] S. Jalaleddine, and L. Johnson, "Associative IC Memories with Relational Search and Nearest-Match Capabilities," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 6, pp. 892-900, June 1992.

[54] S. Santhanam, "StrongARM SA110, a 160 MHz, 32 B, 0.5W CMOS ARM Processor," In Hot Chips 8, August 1996.

[55] C. Zukowski, and S. Wang, "Use of Selective Precharge for Low-Power on the Match Lines of CAMs," *International Symp. on Circuits and Systems*, 1997.

[56] R. Heald, and J. Holst, "A 6-ns Cycle 256-kb Cache Memory and Memory Management Unit," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 11, pp. 1078-1083, Nov. 1993.

[57] H. Higuchi, et. al. "A 2-ns, 5-mW, Synchronous-Powered Static-Circuit Fully Associative TLB," *Proc. of the 1996 Symp. on VLSI Circuits*, pp. 21-22.

[58] S. Sun and P. Tsui, "Limitation of CMOS Supply Voltage Scaling by MOSFET Threshold-Voltage Variation," *IEEE Journal of Solid-State Circuits*, Vol. 30, No. 8, pp. 947-949, Nov. 1993.

[59] K. Itoh, et. al., "Trends in Low-Power RAM Circuit Technologies," *Proc. of the IEEE*, Vol. 83, No. 4, pp. 524-543, April 1995.

[60] J. Rabaey, "Digital Integrated Circuits: A Design Perspective", Prentice Hall, 1996.

[61] S. Shinagawa, "A Multi-Speed Digital Cross-Connect Switching VLSI Using New Circuit Techniques in Dual-Port RAMs," *Proc. of the IEEE Custom Integrated Circuits Conference*, pp. 3.4.1-3.4.4, 1991.

[62] S. Chao et. al., "A 1.3 ns 32x32 three port BiCMOS register file," *in Proc. of the 1994 Bipolar/BiCMOS Circuits and Technology Meeting*, pp. 91-94, 1994.

[63] M. Nomura et. al., "A 500-MHz, 0.4-μm CMOS, 32x32 3-Port Register File," *Proc. of the IEEE Custom Integrated Circuits Conference*, pp. 151-154, 1995.

[64] R. L. Franch, J. Ji, and C. L. Chen, "A 640-ps, 0.25-μm CMOS, 16x64-b Three-Port Register File," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 8, pp. 1288-1292, August 1997.

[65] J. Alowersson, "A CMOS Circuit Technique for High-Speed RAMs," *IEEE International ASIC Conference*, pp. 243-246, 1993.

[66] H. Nambu, "A 1.8ns access, 550MHz 4.5Mb CMOS SRAM" *in Proc. of the International Solid-State Circuits Conference*, pp. 360-361, Feb. 1998.

[67] K. Sung and Y. Leblebici, "CMOS Digital Integrated Circuits: Analysis and Design", McGraw Hill, 1996.

[68] T. Hirose, et. al., "A 20-ns 4-Mb CMOS SRAM with Hierarchical Word Decoding Architecture," *IEEE Journal of Solid-State Circuits*, vol. 25, No. 5, pp. 1068-1074, October 1990.

[69] K. Osada, et. al., "A 2ns Access 285MHz, Two-Port Cache Macro using Double Global Bit-Line Pairs," *Proc. of the IEEE International Solid-State Circuit Conference*, pp. 402-403, 1997.

[70] S. Yamamoto et. al., "A 256K CMOS RAM with Variable Impedance Loads", *Proc. of the IEEE International Solid-State Circuit Conf.*, pp. 58-59, Feb 1985.

[71] H. Mizuno, et. al., "Driving Source-Line Cell Architecture for Sub-1-V High-Speed Low-Power Applications," *IEEE Journal of Solid-State Circuits*, vol. 31, No. 4, pp. 552-557, April 1996.

[72] H. Yamauchi, et. al. "A 0.5V single Power Supply Operated High-Speed Boosted and Offset-Grounded (BOGS) SRAM Cell Architecture," *IEEE Transactions on VLSI Systems*, vol. 5, No. 4, pp. 377-387, Dec. 1997.

[73] A. Fahim, M. Khellah, and M.I.Elmasry, "A Low-Power High-Performance Embedded SRAM Macrocell" *Proc. of the 8th Great Lakes Symp. on VLSI*, pp. 13-18, Lafayette, LA, Feb. 1998

[74] J. Alowersson and P. Andersson, "SRAM Cells for Low-Power Write Buffer Memories," *in Proc. of the IEEE Low-Power Electronics*, pp. 61-62, 1995.

[75] H. Shinohara, "A Flexible Multi-Port RAM Compiler for Data Path," *IEEE Journal of Solid-State Circuits*, vol. 26, No. 3, pp. 343-349, March 1991.

[76] T. Blalock et. al., "A High-Speed Clamped Bit-Line Current-Mode Sense Amplifier," *IEEE Journal of Solid-State Circuits*, vol. 26, No. 4, pp. 542-548, April 1991.

[77] K. Sasaki et. al., "A 7ns 140mW 1Mb CMOS SRAM with Current Sense Amplifier," *Proc. of the International Solid-State Circuits Conference*, pp. 208-209, Feb. 1992.

[78] T. Sakuri et. al., "Alpha-Power Law MOSFET Model and its Application to CMOS Inverter Delay and Other Formulas," *IEEE J. of Solid-State Circuits*, Vol. 25, No. 2, pp. 584-593, Apr. 1990.

[79] K. Chen, and C. Hu, "Performance and V_{dd} Scaling in Deep Submicrometer CMOS," *IEEE J. of Solid-State Circuits*, Vol. 25, No. 2, pp. 1586-1589, Oct. 1998.

[80] R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and Threshold Voltage Scaling for Low Power CMOS," *IEEE J. of Solid-State Circuits*, Vol. 32, No. 8, pp. 1210-1216, Aug. 1997.

[81] B. J. Sheu et. al., "BSIM Berkeley Short-Channel IGFET Model for MOS transistors," *IEEE J. of Solid-State Circuits*, Vol. SC-22, pp. 558-566, April. 1990.

[82] T. Kuroda, et. al., "A 0.9V 150MHz 10 mW 4 mm^2 2-D Discrete Cosine Transform Core Processor with Variable Threshold Voltage Scheme," *IEEE J. of Solid-State Circuits*, Vol. 31, pp. 1770-1779, Nov. 1996.

[83] S. Mutoh et. al., "1-V Power Supply High-Speed Digital Circuit Technology with Multi-Threshold Voltage CMOS," *IEEE J. of Solid-State Circuits*, Vol. 30, No. 8, pp. 847-853, August 1995.

[84] S. Shigematsu, et. al. "A 1-V High-Speed MTCMOS Circuits Scheme for Power-Down Application Circuits," *IEEE J. of Solid-State Circuits*, Vol. 32, No. 6, pp. 861-869, June 1997.

[85] J. Kao, et. al. "MTCMOS Hierarchical Sizing Based on Mutual Exclusive Discharge Patterns," *Proc. of the 35th Design Automation Conference*, San Francisco, CA, June 1998.

[86] M. Stan, "Low-Threshold CMOS Circuits with Low Standby Current", *Proc. of the International Symposium on Low-Power Electronics and Design*, pp. 97-99, Monterey, CA, Aug. 1998.

[87] K. Kumagai, "A Novel Powering-Down Scheme for Low V_t CMOS Circuits," *Proc. of the 1998 Symposium on VLSI Circuits*, pp. 44-45, 1998.

[88] J. Hatler and F. Najm, "A Gate-Level Leakage Power Reduction Method for Ultra-Low-Power CMOS Circuits," *Proc. of IEEE Custom Integrated Circuits Conference*, pp. 475-478, Santa Clara, CA, May, 1997.

[89] Z. Chen, L. Wei, and K. Roy, "Estimation of Stand-by Leakage Power in CMOS Circuits Considering Accurate Modelling of Transistor Stacks," *Proc. of the Internationl Symposium on Low-Power Electronics and Design*, Monterey, CA, Aug. 1998.

[90] Y. Ye, S. Borkar, and V. De, "A New Technique for Standby Leakage Reduction in High-Performance Circuits" *Proc. of the 1998 Symposium on VLSI Circuits*, pp. 40-41, 1998.

[91] K. Usami, et. al. "Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor," *IEEE Custom Integrated Circuits Conference*, pp. 131-134, May 1997.

[92] L. Wei, et. al., "Design and Optimization of Low-Voltage High Performance Dual Threshold CMOS Circuits," *ACM/IEEE 35th Design Automation Conference*, June 1998.

[93] M. Miyazaki, H. Mizuno, and K. Ishibashi, "A Delay Distributing Squeezing Scheme with Speed-Adaptive Threshold-Voltage CMOS (SA-Vt CMOS) for Low-Voltage LSIs," *Proc. of the Internationl Symposium on Low-Power Electronics and Design*, Monterey, CA, Aug. 1998.

[94] Z. Chen, et. al., "0.18µm Dual V_t MOSFET Process and Energy-Delay Measurement," *International Electron Devices Meeting Digest*, pp851, 1996.

[95] J. Wang, et. al., "Design of Standard Cells Used in Low-Power ASIC's Exploiting the Multiple-Supply-Voltage Scheme," *Proc. of the 1998 ASIC Conference*, pp. 119-123, Rochester, NY, Sept. 1998.

[96] M. Johnson, and K. Roy, "Datapath Scheduling with Multiple Supply Voltages and Level Converters," *ACM Transactions on Design Automation*, July 1997.

[97] A. Stratakos, A., "High-efficiency Low-voltage DC-DC Conversion for Portable Applications," *Proc. of the International Workshop on Low-Power Design*, 1994.

[98] D. Chen et. al., "An Exact Algorithm for Low-Power Library-Specific Gate Resizing," *ACM/IEEE 33rd Design Automation Conference*, pp. 763-788, June 1998.

[99] "SIS: A System for Sequential Circuit Synthesis," Report M92/41, UC Berkeley.

[100] M. Pedram, "Power Minimization in IC Design: Principles and Applications," *ACM Trans. on Design Automation of Electronic Systems*, Vol. 1, No. 1, pp. 3-56, Jan. 1996.

[101] L. Benini, A. Bogliolo, M. Favalli, and G. De Micheli, "Regression Models for Behavioral Power Estimation," *Sixth International Workshop on Power and Timing Modelling, Optimization, and Simulation*, Sept. 1996.

[102] S. Turgis, N. Azemard, and D. Auvergne, "Explicit Evaluation of Short Circuit Power Dissipation for CMOS Logic Structures," *1995 Int'l Symp. on Low Power Design*, pp. 129-134.

[103] HSPICE User's Manual, Meta-Software Inc., 1995.

[104] A. Deng, "Power Analysis for CMOS/BiCMOS Circuits," *Proc. of the International Workshop on Low Power Design*, pp. 3-8, 1994.

[105] "A D&T Round Table on Low Power Design," *IEEE Design and Test of Computers Magazine*, pp. 84-90, Winter 1995.

[106] C. Tsui, R. Marculescu, D. Marculescu, and M. Pedram, "Improving the Efficiency of Power Simulators by Input Vector Compaction," *Proc. of the 33rd Design Automation Conference*, pp. 165-168, June 1996.

[107] R. X. Gu, and M. I. Elmasry, "Power Dissipation Analysis and Optimization of Deep Sub-micron CMOS Digital Circuits," *IEEE Journal of Solid-State Circuits*, v. 31, no. 5, pp. 707-713, May 1996.

[108] M. Xakellis, and F. Najm, "Statistical Estimation of the Switching Activity in Digital Circuits," *Proc. of the 31st Design Automation Conference*, pp. 728-733, 1994.

[109] A. Bogliolo, B. Ricco, L. Benini, and G. De. Micheli, "Accurate Logic Level Power Estimation," *Proc. of the IEEE Symposium on Low Power Electronics*, pp. 40-41, 1995.

[110] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks," *Proc. of the IEEE Int'l Conf. on Computer Aided Design*, pp. 402-407, November. 1992.

[111] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits", *Proc. of the 29th ACM/IEEE Design Automation Conference*, 1992.

[112] C. Tsui, M. Pedram, and A. Despain, "Efficient Estimation of Dynamic Power Consumption Under a Real Delay Model," *Proc. of the IEEE International Conference on Computer Aided Design*, pp. 224-228, 1993.

[113] C. Tsui, M. Pedram, and A. Despian, "Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs," *Proc. of the 31st ACM/IEEE Design Automation Conference*, pp. 18-23, 1994.

[114] T. Chou, K. Roy, and S. Parasad, "Estimation of Circuit Activity Considering signal Correlations and Simultaneous Switching," *Proc. of the IEEE International Conference on Computer Aided Design*, pp. 300-303, November 1994.

[115] R. Marculescu, et. al., "Switching Activity Analysis Considering Spatiotemporal Correlations," *Proc. of the IEEE International Conference on Computer Aided Design*, pp. 294-299, 1994.

[116] F. Najm, and M. Zhang, "Extreme Delay Sensitivity and Worst-Case Switching Activity in VLSI Circuits," *Proc. of the 32nd ACM/IEEEDesign Automation Conf.* 1995.

[117] R. Marculescu, et. al., "Efficient Power Estimation for Highly Correlated Input Streams," *Proc. of the 32nd ACM/IEEE Design Automation Conference*, 1995.

[118] D. Cheng, et.al., "A New Hybrid Methodology for Power Estimation," *Proc. of the 33rd Design Automation Conference*, pp. 439-444, June 1996.

[119] Synopsys' DesignPower, *http://www.synopsys.com/products/power/power.html.*

[120] P. Landman, "High-Level Power Estimation," *Proc. of the International Symp. on Low-Power Electronics and Design*, pp. 29-35, 1996.

[121] D. Marculescu, et. al., "Information Theoretic Measures for Energy Consumption at Register Transfer Level", *Proc. of the International Symposium on Low Power Design*, pp. 81-86, April 1995.

[122] F. Najm, "Towards a High-Level Power Estimation Capability," *Proc. of the International Symposium on Low Power Design*, pp. 87-92, April 1995.

[123] S. Powell and P. Chau, "Estimation Power Dissipation of VLSI Signal Processing Chips: The PFA Technique," *VLSI Signal Processing IV*, pp. 250-259, 1990.

[124] P. Landman and J. Rabaey, "Black-Box Capacitance Models for Architectural Power Analysis," *Proc. of the International Workshop on Low Power Design*, pp. 165-170, April 1994.

[125] S. Gupta and F. Najm, "Power Macro-Modeling for High Level Power Estimation," *Proc. of 34th ACM/IEEE Design Automation Conf.*, pp. 365-370, June 1997.

[126] A. Raghunathan, S. Dey and N. Jha, "Register-Transfer Level Estimation Techniques for Switching Activity and Power Consumption," *ICCAD*, Nov. 1996.

[127] H. Mehta, R. Owens, and M. Irwin, "Energy Characterization Based on Clustering," *Proc. of the 33rd Design Automation Conference*, pp. 702-707, 1996.

[128] Senté's Watt Watcher, *http://www.powereda.com/wwinfo.htm.*

[129] S. Ross, "Introduction to Probability and Statistics for Engineers and Scientists," J Wiley, 1987.

[130] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo Approach for Power Estimation," *IEEE Trans. on VLSI Systems*, Vol. 1, No. 1, pp. 63-71, March. 1993.

[131] P. Schneider, and S. Krishnamoorthy, "Effects of Correlation in Accuracy of Power Analysis - An Experimental Study," *Proc. of the International Symposium on Low Power Electronics and Design*, pp. 113-116, 1996.

[132] P. Pant, V. De, and A. Chatterjee, "Simultaneous Power Supply, Threshold Voltages, and Transistor Size Optimization for Low-Power Operation of CMOS Circuits," *IEEE Transactions on VLIS Systems*, Vol. 6, No. 4, pp. 538-545, Dec. 1998.