A Sub-threshold Cell Library and Methodology

by

Joyce Y. S. Kwong

B.A.Sc. in Computer Engineering
University of Waterloo, 2004

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author .....
Department of Electrical Engineering and Computer Science
May 25, 2006

Certified by .....
Anantha P. Chandrakasan
Joseph F. and Nancy P. Keithley Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Chairman, Department Committee on Graduate Students

A Sub-threshold Cell Library and Methodology

by

Joyce Y. S. Kwong

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Sub-threshold operation is a compelling approach for energy-constrained applications where speed is of secondary concern, but increased sensitivity to process variation must be mitigated in this regime. With scaling of process technologies, random within-die variation has recently introduced another degree of complexity in circuit design. This thesis proposes approaches to mitigate process variation in sub-threshold circuits through device sizing, topology selection and fault-tolerant architecture.

This thesis makes several contributions to a sub-threshold circuit design methodology. A formal analysis of device sizing trade-offs between delay, energy, and variability reveals that while minimum size devices provide lowest energy and delay in sub-threshold, their increased sensitivity to random dopant fluctuation may cause functional errors. A proposed variation-driven design approach enables consistent sizing of logic gates and registers for constant functional yield. A yield constraint imposes energy overhead at low power supply voltages and changes the minimum energy operating point of a circuit. The optimal supply and device sizing depend on the topology of the circuit and its energy versus V_{DD} characteristic. The analysis resulted in a 56-cell library in 65nm CMOS, which is incorporated in a computer-aided design flow. A test chip synthesized from this library implements a fault-tolerant FIR filter. Algorithmic error detection enables correction of transient timing errors due to delay variability in sub-threshold, and also allows the system frequency to be set more aggressively for the average case instead of the worst case.

Thesis Supervisor: Anantha P. Chandrakasan
Title: Joseph F. and Nancy P. Keithley Professor of Electrical Engineering

Acknowledgments

This thesis concludes the first chapter of my experiences at MIT, a time that has been productive and enjoyable, thanks largely to many people I have come to known here.

I am grateful to my thesis advisor, Professor Chandrakasan, for including me in his group and for his invaluable guidance throughout my studies. He motivates by setting high standards, but he is always sensitive to the interests and needs of his students. Professor Chandrakasan's enthusiasm and love of research have inspired me to pursue a Ph.D. I trust that this is but the beginning of a rewarding and fruitful relationship.

This work would not be possible without chip fabrication support from Dennis Buss, David Scott, Alice Wang, Terence Breedijk, Richard White, and Texas Instruments.

I am grateful to my family for their continuing support and constant reminders to eat more. They provided all the right opportunities but more importantly, allowed me to find my own way. Thanks to my Waterloo friends for always being on my side, and to Milton Lei for putting up with me these past few years. I am also much indebted to Zhengya Zheng; I literally would not be here today without his personal and technical advice.

My research group is a boundless source of inspiration. Thanks especially to Brian Ginsburg (the honourary Canadian and guru of all things), who bailed me out more than once during my first tape-out. I am very fortunate to have Benton Calhoun as my mentor. His help was invaluable when I was starting this research, and he is the model to which I aspire. I am grateful to Naveen Verma for his very constructive comments on parts of this thesis. I have also thoroughly enjoyed many entertaining discussions with all members of Ananthagroup.

Vivienne, Yogesh, and Chun-Ming: I am glad we made it through first year together in one piece. Viv, Daniel, Denis, Naveen, and Payam: I feel much more at home with the strong Canadian presence in our group. Taeg Sang: thanks for listen-

ing to my daily/hourly rants. Viv (again), Maryam, Rumi, and Karen: our SATC outings keep me sane even when all else seem to go wrong. Finally, no acknowledgement would be complete without thanking Margaret Flaherty for making life in Ananthagroup run much more smoothly.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The advent of portable electronics and emerging technologies such as sensor networks have led to interest in energy-aware circuit design techniques. Sub-threshold operation, in which the power supply voltage is lowered to below the transistor threshold voltage, enables drastic savings when energy rather than speed is the primary constraint. Sub-threshold and above-threshold behavior of digital circuits share some common aspects, but differ in many others. In particular, circuits in weak inversion display order of magnitude higher variability from process variation. This thesis addresses these differences and presents a methodology for designing robust, low-energy digital circuits for sub-threshold applications.

1.2 Previous Work

Operating digital circuits in the weak inversion region was first considered in [4], which examined the theoretical limits to supply voltage scaling. A minimum power supply (V_{DD}) of three to four times the thermal voltage V_{th} was found to be necessary for an inverter to have sufficient gain. Work in [5] showed that minimum energy operation occurs in the sub-threshold region. This was revisited in [6], which plotted constant energy contours for a ring oscillator as V_{DD} and transistor threshold voltage (V_T) were

varied. The minimum energy point was shown to lie in sub-threshold and varied with activity factor of the circuit. Analytical expressions for the optimum V_{DD} and V_T for minimum energy [7] showed that V_{DDopt} is independent of frequency and depends on the relative contributions of active and leakage energy.

Other work in [8] and [9] investigated the theoretical use of pseudo-NMOS and domino styles in sub-threshold. Authors of [10] presented simulations of a sub-threshold circuit in pseudo-NMOS for an adaptive filter designed for hearing aids. A 0.35μm test chip implementing an 8x8 array multiplier was described in [11]. The first major digital system operating in weak inversion was demonstrated in the 0.18μm FFT processor of [12]. Circuit techniques in device sizing and choice of topology enabled operation down to 180mV, although minimum energy operation occurred at 350mV. A 0.18μm test chip in [13] compared two synthesized FIR filters, one with an unmodified commercial standard cell library, and one sized for operation at the minimum energy point. It was concluded that sizing to operate at the minimum energy V_{DD} enabled small energy savings at the worst case process corner, but imposed energy overhead under typical operating conditions. Commercial standard cell libraries in static CMOS provided a good solution for sub-threshold logic in this process generation.

Since sub-threshold currents are exponentially dependent on V_T, increased variation in recent process nodes gave rise to a new research focus. In [14], the authors analyzed static noise margin (SNM) in traditional six-transistor SRAM with changes in supply voltage, temperature, transistor sizes, and V_T mismatch, and provided a statistical model for the tail of the SNM distribution. Work in [15] addressed the problem of SNM variation in practical terms and presented a 256-kbit SRAM in 65nm CMOS that functions at 400mV, based on a 10-transistor bitcell with buffered read and a floating V_{DD} during write.

A statistical analysis of the energy and delay variation of a sub-threshold inverter chain was presented in [16], along with an algorithm to compute the greatest of several lognormal delay distributions. However, the effect of V_T variation on noise margins in logic gates was not addressed until [17], which considered a sub-threshold

inverter whose output levels were degraded by leaking devices, such as in a register file. Authors of [18] derived a unified model for gate delay in strong- and weak inversion and used it to examine the sensitivity of delay variability to deviations in V_T and channel length, as well as the effect of spatial correlations. These recent works present encouraging results, but mitigating process variation in sub-threshold remains a challenging task, with many opportunities yet to be explored.

1.3 Sub-threshold Operation

1.3.1 Device Operation in Sub-threshold

Sub-threshold circuits employ leakage currents to charge and discharge load capacitances. In this regime, the source-to-drain weak inversion current is the main leakage contributor, while other leakage currents, such as gate tunneling current and gate-induced drain leakage current, are typically considered to be negligible. Equation 1.1 gives a simple model for the sub-threshold drain current [19]

$$I_{Dsub-threshold} = I_o e^{\left(\frac{V_{GS}-V_T+\eta V_{DS}}{nV_{th}}\right)} \left(1 - e^{\left(\frac{-V_{DS}}{V_{th}}\right)}\right), \tag{1.1}$$

where I_o is the drain current when $V_{GS} = V_T$ and is given in Equation 1.2 [19][20].

$$I_o = \mu_o C_{ox} \frac{W}{L}(n-1)V_{th}^2 \tag{1.2}$$

In Equation 1.1, I_D varies exponentially with V_{GS} and V_T, the device threshold voltage. V_{th} denotes the thermal voltage and $n = (1 + C_d/C_{ox})$ is the sub-threshold slope factor. η represents the drain-induced barrier lowering (DIBL) coefficient. The I_D versus V_{DS} characteristic of Figure 1-1 resembles that of strong inversion, with a quasi-linear region at low V_{DS} and a quasi-saturation region when V_{DS} is close to V_{DD}. The roll-off current at small V_{DS} is modeled by the term in the rightmost parentheses in Equation 1.1. The quasi-saturation slope results from DIBL and is modeled by η.

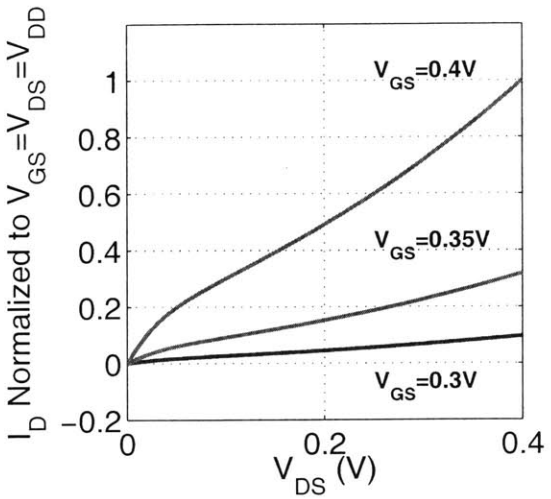


Figure 1-1: I_D versus V_{DS} characteristic in a 65nm technology. V_{DD} = 0.4V and V_{GS} = 0.3V, 0.35V, and 0.4V.

1.3.2 Minimum Energy Operating Point

The concept of an optimal supply voltage to minimize energy has been examined using different approaches, for example in [5], [7], and [21]. From [7], the total energy per operation consumed by an arbitrary circuit is modeled as

$$E_{DYN} = C_{eff} V_{DD}^2 \tag{1.3}$$

$$E_{LEAK} = W_{eff} I_{leak} V_{DD} t_d L_{DP} \tag{1.4}$$

$$E_T = E_{DYN} + E_{LEAK} = C_{eff} V_{DD}^2 + W_{eff} I_{leak} V_{DD} t_d L_{DP}. \tag{1.5}$$

E_{DYN} and E_{LEAK} model the dynamic switching and leakage energy per cycle respectively. C_{eff} and W_{eff} denote the average total switched capacitance and normalized width contributing to leakage current. t_d and I_{leak} represent the delay and leakage current of a characteristic inverter, while L_{DP} is the logic depth in terms of the inverter delay. Figure 1-2 shows the different energy components as V_{DD} is reduced. E_{DYN} decreases quadratically with the supply voltage. The leakage current reduces due to DIBL, while t_d goes up exponentially at sub-threshold voltages. The

net effect is an exponential increase in the leakage energy per cycle as the supply voltage scales down. The opposing trends in E_{DYN} and E_{LEAK} give rise to an optimal supply voltage V_{DDopt} at which total energy per operation is minimized.

V_{DDopt} depends on the relative contributions of dynamic and leakage energy components [7]. A system dominated by dynamic energy, such as a ring oscillator, has a lower V_{DDopt} than a system with a larger proportion of leakage energy. For this reason, V_{DDopt} also changes with system conditions such as workload or duty cycle.

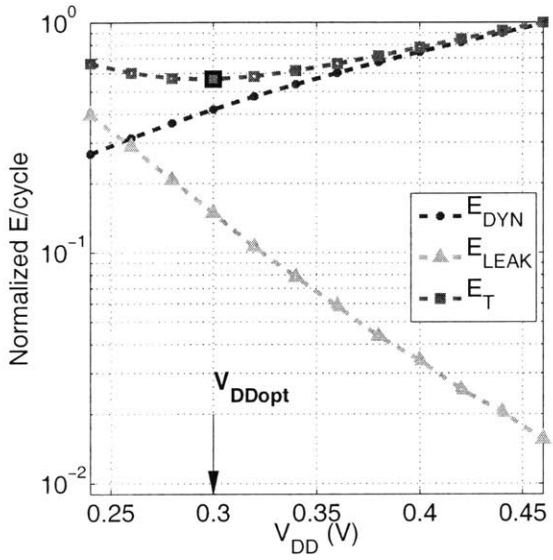


Figure 1-2: Dynamic (E_{DYN}), leakage (E_{LEAK}), and total energy (E_T) per operation in a 32-bit adder.

1.4 Process Variation

1.4.1 Classification and Sources of Variation

As with all manufacturing processes, semiconductor fabrication is subject to many sources of variation. A survey of semiconductor process variation can be found in [22], while [23] performs a study correlating MOSFET model parameter variation to the underlying process settings. In the context of circuit design, these sources of variation are typically classified into global (inter-die) and local (intra-die) variation [24].

Global variation affects all devices on a die equally and causes device characteristics to vary from one die to the next. For example, global variation results from wafer-to-wafer discrepancies in alignment or processing temperatures. In sub-threshold logic, the main effect of global variation is seen at skewed P/N corners with a strong PMOS and weak NMOS, or vice versa. Logic errors may occur if the weaker device cannot drive the gate output to a full '0' or '1' level. Previous work [25] has addressed global variation in sub-threshold design by sizing the PMOS/NMOS width ratio to satisfy opposing constraints at the two skewed corners.

Local variation affects devices on the same die differently and can consist of both systematic and random components. For example, an aberration in the processing equipment gives rise to systematic variation, while placement and number of dopant atoms in device channels contribute to random variation. Device models in advanced process technologies typically account for both random dopant fluctuation (RDF) and differences in the effective channel length (L_{eff}). As noted by [16], L_{eff} variation affects V_T through the DIBL coefficient [26], and becomes less significant at low supply voltages. On the other hand, V_T variation from RDF is independent of the supply voltage and is typically modeled as a Gaussian distribution, with a standard deviation inversely proportional to the square root of the channel area [27]. Therefore, at sub-threshold supply voltages, RDF is the dominating factor in local V_T mismatch.

1.4.2 Lognormally Distributed Sub-threshold Characteristics

The lognormal distribution occurs frequently in analysis of sub-threshold circuit variability. This is due to the exponential dependence of current on V_T, and the assumption that V_T is normally distributed from local process variation.

A random variable X is lognormally distributed if $Y = ln(X)$ has a normal distribution. The probability density function (PDF) of a lognormal distribution is characterized by two parameters M and S as follows

$$P(X) = \frac{1}{S\sqrt{2\pi}x} e^{-(lnx-M)^2/2S^2}. \tag{1.6}$$

M and S correspond to the mean and standard deviation of the normal variable $Y = ln(X)$. The mean and variance of the lognormal variable X are given by

$$E(X) = \mu = e^{S^2/2+M} \tag{1.7}$$

$$Var(X) = \sigma^2 = e^{S^2+2M}(e^{S^2} - 1). \tag{1.8}$$

Sub-threshold currents under a Gaussian model of V_T mismatch have a lognormal distribution. Therefore, performance metrics with a first order dependence on current, such as delay and leakage energy, also follow this distribution. An example delay distribution of an inverter in sub-threshold is shown in Figure 1-3, with simulation results plotted in markers and fitted to an ideal lognormal distribution. It is worth noting that the distribution is asymmetric with a long tail on the right. This implies that below average circuit delays deviate only slightly from the mean, while above average delays can reach several times the nominal value.

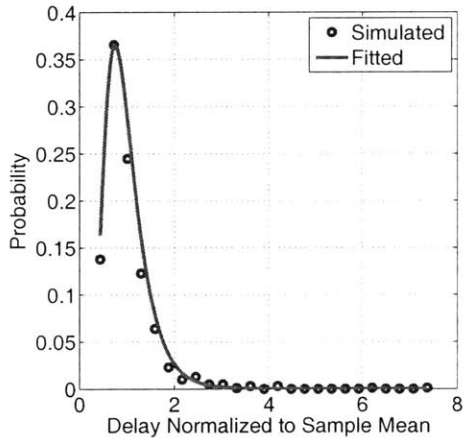
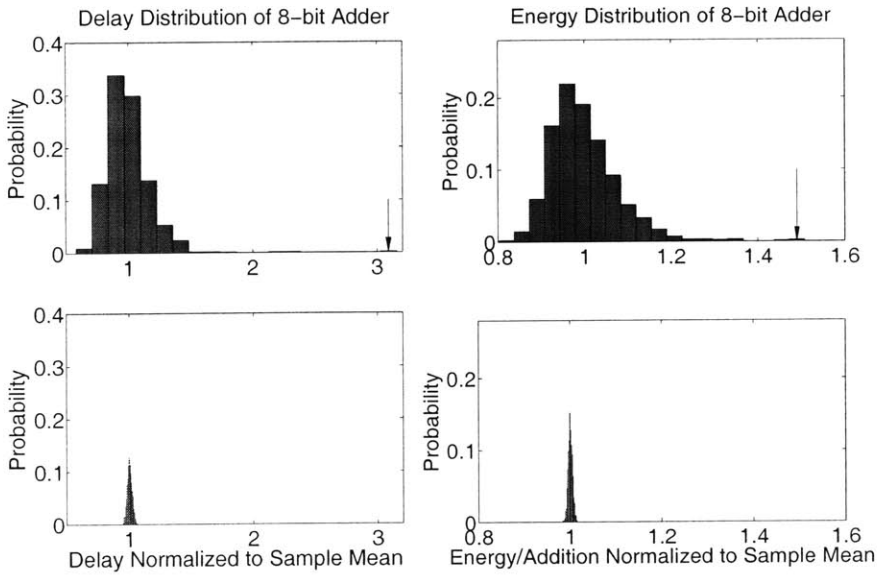


Figure 1-3: Lognormal inverter delay distribution at V_{DD}=0.25V

One statistic of interest in comparing circuit variability is the coefficient of variation, which for a lognormal variable X is defined as

$$c_v = \sigma/\mu = \sqrt{e^{S^2} - 1}. \tag{1.9}$$

c_v allows comparison of the variation in two populations with significantly different mean values. For instance, Figure 1-4(a) and Figure 1-4(b) respectively plot the delay and energy distribution of two 8-bit adders in sub-threshold and above-threshold. Since circuit delay is significantly higher in sub-threshold, the standard deviation is correspondingly larger in magnitude than at nominal voltage. We thus compare the spread of the two distributions after normalizing each to their respective sample mean. The figures show that the normalized spread at low V_{DD} is an order of magnitude higher than at nominal voltage. Mitigating increased variability is important in sub-threshold design and forms the focus of this thesis.



(a) (b)

Figure 1-4: (a) Delay and (b) energy distribution of an 8-bit adder. Top panels are simulated in sub-threshold (0.3V), while bottom show above-threshold data (1.2V). The x-axis of each plot is normalized to the sample mean.

1.5 Thesis Contribution and Organization

Previous work has demonstrated the feasibility of operating circuits in the sub-threshold region and identified variation as the primary challenge. This thesis presents

a design methodology for sub-threshold circuits with emphasis on within-die variation, which remains a relatively unexplored area. The thesis contributes in the following areas.

Chapter 2: Sub-threshold Device Sizing

- A general analysis of device sizing given opposing objectives of reducing variability and energy consumption.
- A guideline for characterizing functional failure in logic gates due to insufficient output swing.

Chapter 3: Sub-threshold Standard Cell Library

- Design decisions relating to a 65nm CMOS sub-threshold library, including selection of logic functions and drive strengths.
- Topology selection and transistor sizing of sub-threshold registers, considering the impact of local variation.
- Discussion of issues specific to using a computer-aided design flow for sub-threshold circuits.

Chapter 4: Minimum Energy Operation With Process Variation

- Analysis of how a yield constraint imposes an energy overhead and affects the minimum energy operating point of a circuit.
- Given a yield constraint, upsizing to operate at reduced supply voltages provides energy savings in certain scenarios.

Chapter 5, 6: Algorithm-Based Fault Tolerance Implementation and Results

- Design of an FIR filter using algorithm-based fault tolerance techniques to correct transient timing errors.
- Simulation results of the FIR filter test chip synthesized from the 65nm sub-threshold library.

- Evaluation of the effectiveness and overhead costs of algorithm-based fault tolerance.

Chapter 2

Sub-threshold Device Sizing

Previous work in [12] and [13] have successfully demonstrated 0.18μm test chips in sub-threshold using commercial standard cell libraries with slight modifications. However, increased variability with technology scaling has significant impact on weak inversion operation and motivates a closer examination of standard cell design. This chapter discusses device sizing trade-offs and choice of topology in standard cells. It is shown that minimum size devices allow minimum energy consumption but may exhibit unacceptable output swing and performance variability.

2.1 Traditional Sizing Approach for Minimum Energy

This section describes the basis for transistor sizing in the sub-threshold standard cell library. It presents energy and delay trade-offs relating to global device widths and the PMOS/NMOS width ratio (β). Without considering variation, it is shown that minimum size devices enable minimum energy and delay in sub-threshold.

2.1.1 Ratio of PMOS and NMOS Width

As a starting point, energy and delay are characterized for an inverter chain while varying the ratio of PMOS and NMOS widths. We will refer to the β ratio, typically

defined for an inverter as

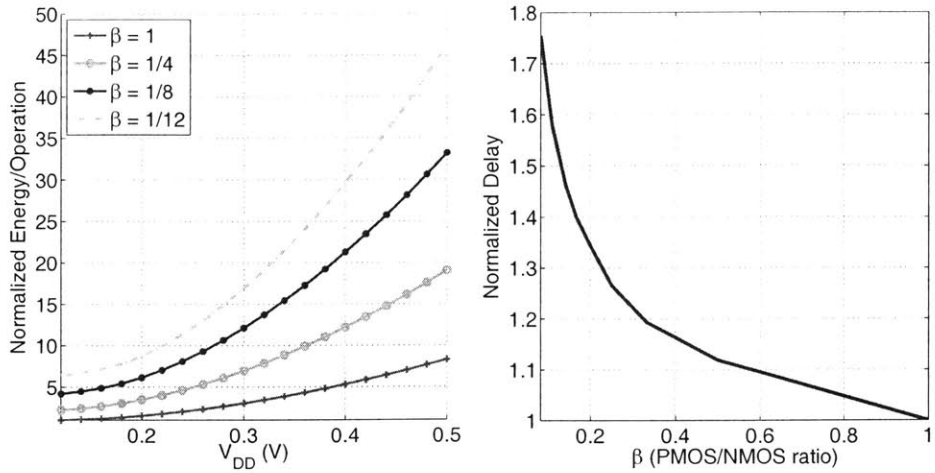
$$\beta = \frac{\text{PMOS width}}{\text{NMOS width}}. \tag{2.1}$$

In this particular technology, the sub-threshold NMOS current is weaker than PMOS current. To balance the current strengths, the NMOS needs to be upsized relative to PMOS. Therefore in contrast to above-threshold design, we perform analysis for values of $\beta \leq 1$.

In sub-threshold digital design, β not only affects relative rise/fall propagation delays, but also the output-high (V_{OH}) and output-low (V_{OL}) levels of logic gates. Active currents in sub-threshold are comparable in magnitude to idle leakage currents, therefore the pull-up and pull-down networks in a digital gate act as a resistive divider. Varying β changes the relative strengths of the pull-up and pull-down networks and thus the gate output voltage. From [28], a logic gate can achieve minimum V_{DD} operation when PMOS and NMOS devices are sized to carry the same current. However, upsizing NMOS to match the PMOS strength increases leakage and switched capacitance, and does not necessarily decrease energy.

Figure 2-1(a) plots the total energy consumed per cycle in an 11-stage FO4 inverter chain. One cycle is defined as the time taken for an input edge to propagate through the chain. Each curve represents one value of β. Decreasing β by keeping PMOS constant and increasing the size of NMOS causes the circuit to move to a higher energy curve. Since the curves do not intersect, $\beta = 1$ always offers minimum energy even if matching NMOS and PMOS strengths allows minimum V_{DD} operation.

Figure 2-1(b) plots the propagation delay through the chain versus β. Each stage in the chain is loaded by three copies of itself and β is applied to all inverters including the loads. Decreasing β, or increasing NMOS strength, speeds up the pull-down delay (t_{phl}) but slows the pull-up (t_{plh}) edge. This results in a net increase in average delay, defined as $(t_{plh} + t_{phl})/2$. Thus in the nominal case, $\beta = 1$ is optimal for both energy and delay.



(a) (b)

Figure 2-1: (a) Total energy consumed for an edge to propagate through an 11-stage inverter chain and (b) average rising and falling delay through the chain, plotted at various PMOS/NMOS width ratios.

2.1.2 Global Device Width

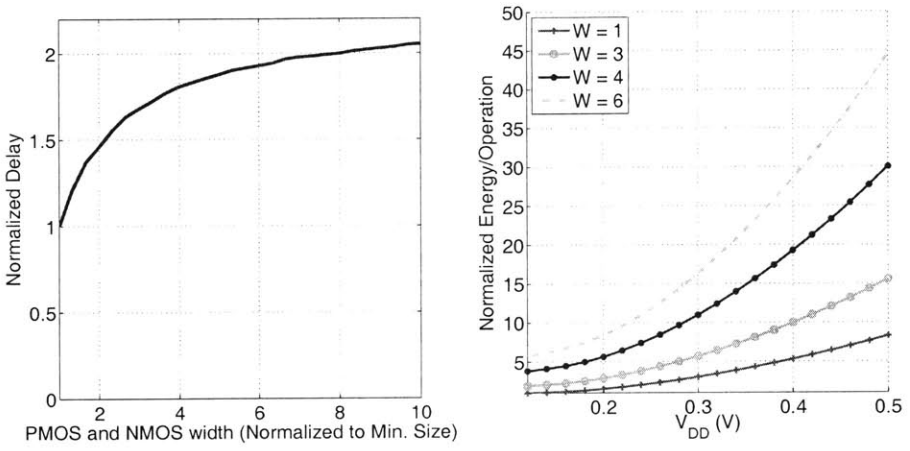
This section examines the impact of increasing both PMOS and NMOS widths on sub-threshold circuit performance. In the context of standard cell design, if one cell is upsized, the preceding cell will also likely be upsized to drive the increased load capacitance, and so on. To emulate this situation, the delay and energy of an FO4 inverter chain are simulated as the PMOS and NMOS widths of all devices are increased.

The trend in delay versus global device width is plotted in Figure 2-2(a). To analyze the results, it is instructive to consider a delay model for a sub-threshold logic gate [7], [18]

$$t_d = \frac{KC_gV_{DD}}{I_oe^{\frac{V_{GS}-V_T}{nV_{th}}}}, \tag{2.2}$$

where K is a delay fitting parameter, C_g is the load capacitance, and the denominator denotes sub-threshold active current. Both C_g and I_o are proportional to the device width (Equation 1.2). From Equation 2.2, sub-threshold delay should nominally stay constant with an increase in width (W). However, it is seen in Figure 2-2(a) that delay

initially increases with W before leveling off to the expected constant trend. This can be attributed to narrow-channel effects. In narrow-width transistors, the classical channel depletion region formed from vertical fields is augmented significantly by fringing fields. This increase in depletion region is modeled in BSIM [26] as a width-dependent adjustment to the threshold voltage. Substituting this width-dependent term in Equation 2.2 and finding $\frac{dt_p}{dW}$, we find that increasing width also increases delay when narrow-channel effects are significant. Since a low-power standard cell library typically employs device sizes close to minimum, it is reasonable to conclude that smaller widths are preferable for reducing circuit delay in sub-threshold.



(a) (b)

Figure 2-2: (a) Average rising and falling delay through an 11-stage FO4 inverter chain and (b) total energy consumed for an edge to propagate through the chain, plotted at various device widths ($W_p = W_n$).

Upsizing leads to a rise in both switched capacitance and leakage current of a circuit. Since delay increases with global upsizing in the presence of narrow channel effects, total energy also rises monotonically. This is illustrated in the total energy versus V_{DD} curves of Figure 2-2(b). Increasing the width of all transistors moves the circuit to a higher energy curve.

We can conclude that a β ratio of 1 and minimum size devices are optimal for minimum energy operation in sub-threshold. However, it is well-known that minimum

size transistors are the most susceptible to V_T variation. This trade-off between energy and variability is considered in the next section.

2.2 Variation-Driven Device Sizing

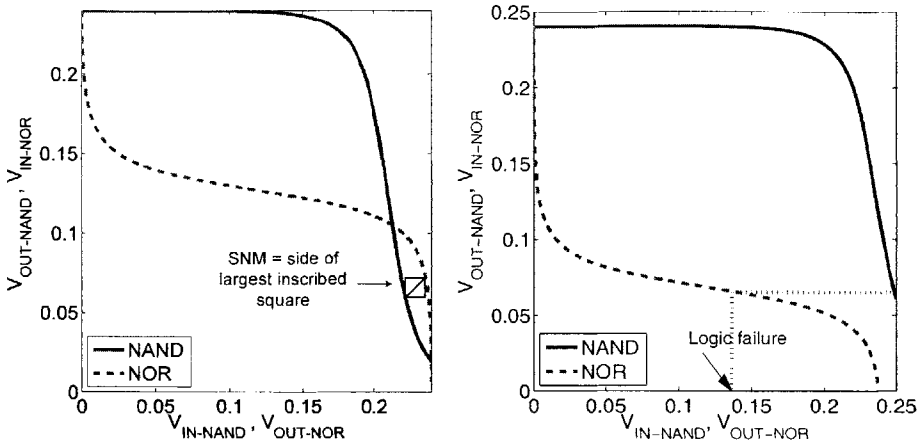
Process variation affects both functionality and performance of sub-threshold digital logic. We define a consistent metric to determine whether logic gates fail functionally because of insufficient output voltage swing. Functional failure rates are found to vary inversely with width and V_{DD}. We then examine current and delay variability for the inverter and other digital circuit primitives. The analysis in this section forms the basis for cell design in the sub-threshold library.

2.2.1 Variability Metrics

Logic Gate Output Swing

In the sub-threshold regime, the ratio of active to idle currents (I_{ON}/I_{OFF}) in a logic gate is much lower than in strong inversion. If, for example, process variation strengthens NMOS relative to PMOS, a pull-up network will not be able to drive the logic gate output fully to V_{DD} because of idle leakage in the pull-down network. This degradation in gate output swing is illustrated in Figure 2-4(a). The solid line shows the voltage transfer characteristic (VTC) of a minimum size inverter in a 65nm technology at skewed global process corner. Dashed lines plot the VTCs when random local V_T mismatch is applied to the inverter. One case shows a severely degraded V_{OL}, which can cause functional error if it is above the input low threshold (V_{IL}) of the succeeding gate. Therefore, V_T variation significantly impacts circuit functionality in deeply scaled technologies.

Among common static CMOS gates, NOR has the worst-case V_{OH} and V_{IL} characteristics because of stacked devices in the pull-up network and parallel devices in the pull-down. NAND similarly exhibits the worst-case V_{OL} and V_{IH}. Therefore, in the context of standard cell library design, V_{OL} of each cell should be checked against



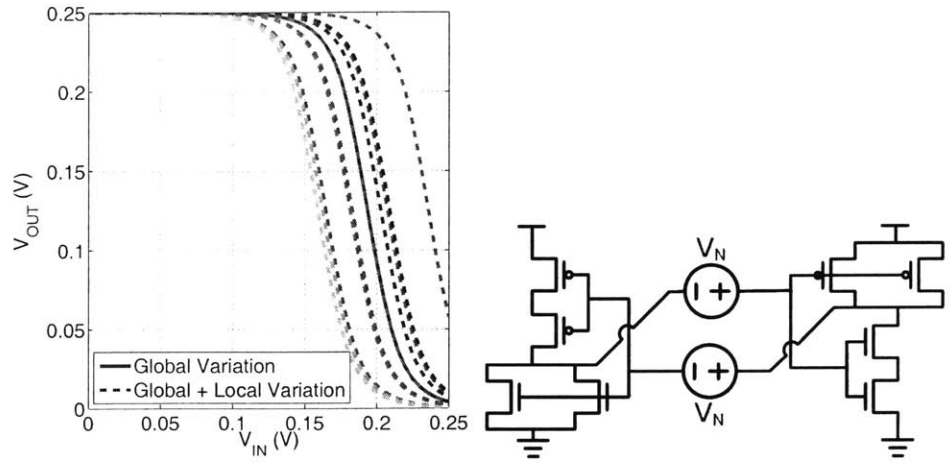
(a) (b)

Figure 2-3: (a) Butterfly plot of NAND/NOR gates with functional output levels. (b) Butterfly plot of NAND with failing V_{OL}.

V_{IL} of NOR and V_{IH} of NAND to capture the worst-case scenario. The butterfly plot, formed by superimposing the VTC of one gate with the mirrored VTC of another, is one way to verify the output levels of a logic gate. The use of butterfly plots is illustrated as follows.

Figure 2-3(a) shows a NAND gate having sufficient output swing such that $V_{OL-NAND}$ produces a logic high output in a succeeding NOR gate. In contrast, the NAND gate in Figure 2-3(b) exhibits $V_{OL-NAND}$=65mV and produces a NOR output of 136mV, close to mid-rail and thus causing logic failure. Note that the absolute value of $V_{OL-NAND}$ required for proper functionality changes with global process conditions. That is, if all NMOS on a die are weakened relative to PMOS, $V_{OL-NAND}$ rises but the VTC of NOR also shifts upward to partially compensate, an effect which is captured by the butterfly plot. Therefore, this method of verifying output levels is preferable to setting an absolute requirement on V_{OL} and V_{OH}.

A gate with failing output levels is analogous to a six-transistor SRAM cell displaying negative static noise margin (SNM), in that the butterfly plots for both cases do not contain an inscribed square. A common method in [1] to measure the SNM of an SRAM cell can also be used to verify the output levels of two back-to-back



(a) (b)

Figure 2-4: (a) Inverter VTCs at skewed process corner with random V_T mismatch. (b) Example circuit [1] for verifying logic gate output levels.

logic gates, as shown in Figure 2-4(b). Both inputs of a gate are varied simultaneously to obtain the worst case skewed VTC. To consider logic gates with up to three stacked devices, we verify the INV, NAND2, and NOR2 gates against NAND3 and NOR3, which give the most stringent V_{IH} and V_{IL} requirements respectively. Sizing of NAND3 and NOR3 are fixed to provide a starting point for designing the remaining gates.

Limitations of the Output Swing Metric

It should be noted that this metric does not reflect the exact mismatch conditions in a circuit. However, it does provide a guideline for sizing standard cells consistently.

The formal definition of noise margin of two back-to-back gates G1 and G2 is given in [29] and is subsequently used to characterize SRAM cell stability in [1]. SNM is equivalent to the maximum noise that can be applied to an infinitely long chain of alternating G1 and G2, before two consecutive gates at the end of the chain have the same logic polarity (functional failure). In this formulation, noise is applied to all gates in the chain in a way that causes the maximum upset in logic levels. Thus when using SNM to size an inverter, we essentially assume that all logic paths in a

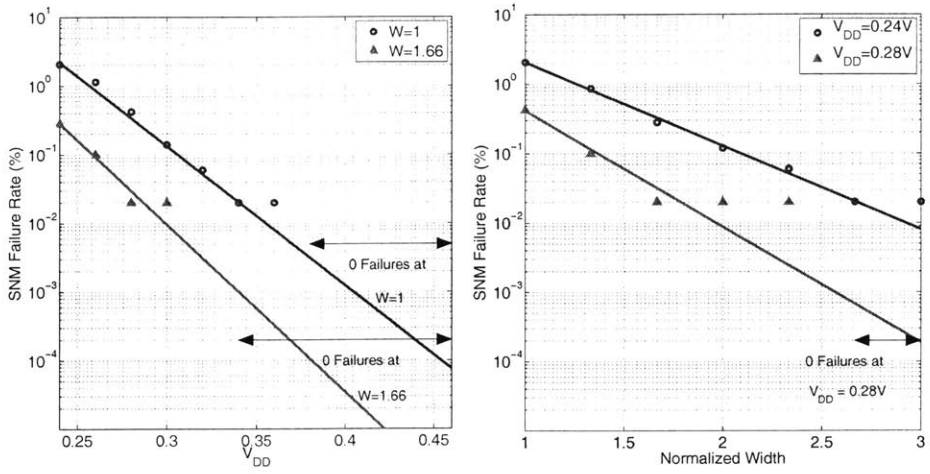
synthesized circuit are composed of alternating inverters and NAND3 gates. This is likely a conservative assumption and can be verified by comparing the failure rate of the inverter in an SNM simulation to that in a logic path from an actual circuit.

To accurately model the failure rate of a custom-designed logic path, we would perform Monte Carlo simulation while plotting VTCs of all gates and tracing the signal propagation through the path. Exact modeling is not possible for standard cell design where the target circuit is unknown. Instead, we can approximate a representative logic path by analyzing the sequence of gates and the logic depth across many synthesized designs. However this requires significantly more design effort than the SNM-based approach and may not lead to a more accurate model.

Definition of Logic Failure

We now define logic failure as negative SNM in the butterfly plot and measure how the failure rate varies with V_{DD} and device sizing. The failure rate is estimated by counting samples with negative SNM in a 5k-point Monte Carlo simulation. This is performed at worst-case temperature and with local V_T mismatch applied to all transistors in the logic gate under test. The global (inter-die) process conditions are also randomized such that the Monte Carlo runs are analogous to sampling logic gates across multiple die. Figure 2-5(a) shows the failure rate versus V_{DD} of an inverter at various widths normalized to minimum size. Simulated values in markers fit closely to an exponential function ae^{bx}, drawn as a solid line. Note that the failure rate decays at a higher rate when W=1.66 compared to W=1. Furthermore, zero samples failed in the 5-k point run at higher voltages, as indicated by arrows on the graph. Figure 2-5(b) illustrates similar trends for failure rate versus NMOS and PMOS widths.

Figure 2-6 plots the failure rate versus normalized device width of INV, NAND2, and NOR2. In NAND2 and NOR2, widths in the critical two-transistor stack are varied while the two parallel devices are kept constant. The failure rates also decay exponentially with widths. By increasing the device width or V_{DD}, the failure rate can be made to approach 0.



(a) (b)

Figure 2-5: SNM failure rate versus (a) V_{DD} and (b) NMOS and PMOS width of inverter.

Current and Delay Variability

In addition to output swing, active current and delay variability are also of interest for the purpose of timing verification. From Section 1.4, the normalized spread of active current distribution is given by

$$\sigma_{I_{sub}}/\mu_{I_{sub}} = \sqrt{e^{(\frac{\sigma_{V_T}}{nV_{th}})^2} - 1}. \tag{2.3}$$

σ_{V_T} increases with smaller channel area, while the sub-threshold swing factor n [26] decreases with lower V_{DD}. Equation 2.3 therefore shows that uncertainty in sub-threshold current through a single device is higher in small devices operating at low voltages.

To examine the impact of different topologies, Figure 2-7 plots the simulated $\sigma_{I_{sub}}/\mu_{I_{sub}}$ versus device width for static CMOS primitives consisting of one to three devices in series. Variability decreases with larger widths as expected. Stacked device topologies clearly display lower spread in active currents.

Figure 2-8 and Figure 2-9 plot the σ/μ variability of pull-up and pull-down delay for the NOR2 and NAND2 gates respectively, as well as for the inverter. Generally

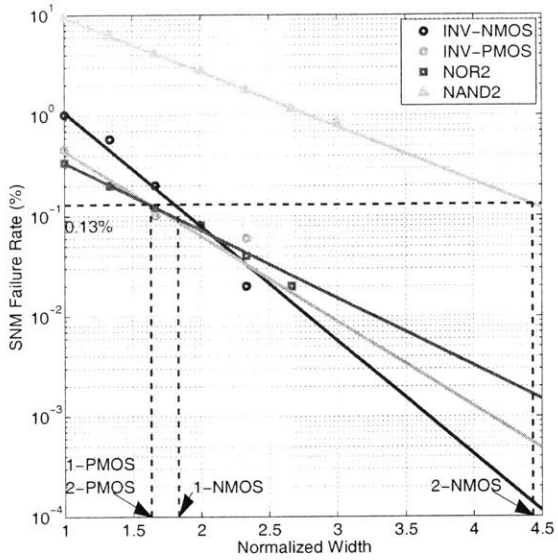


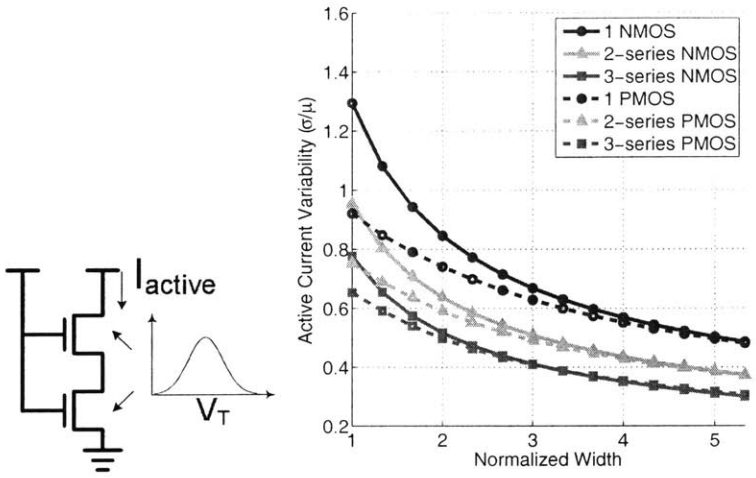
Figure 2-6: Failure rate due to negative SNM in the inverter, NAND2, and NOR2, plotted against device width (normalized to minimum size). V_{DD}=240mV.

delay spread follows the same trend as active current variation and reduces with larger widths or higher V_{DD}. However, inverter delay variability deviates from ideal monotonic behavior and peaks at V_{DD}=300mV. This shows that the simple model $t_d = \frac{KC_gV_{DD}}{I_o e^{\frac{V_{GS}-V_T}{nV_{th}}}}$ provides intuition but cannot accurately predict delay spread.

2.2.2 Constant Yield Device Sizing

We now address the issue of device sizing for single and stacked topologies. In conventional above-threshold design, it is possible to find a fixed width ratio between series devices and the inverter to obtain equivalent *on* resistance. However, in sub-threshold design when the objective is to minimize energy, device sizes should be kept as small as possible while satisfying variability constraints. As seen previously, the width ratio between single and stacked devices to obtain constant yield or delay variability varies with V_{DD}, so the sizing relationship is less clear.

It was observed that compared to a single device, stacked devices display lower current spread but higher uncertainty in output levels, which may lead to functional errors. Reducing the likelihood of functional errors clearly takes precedence, so static



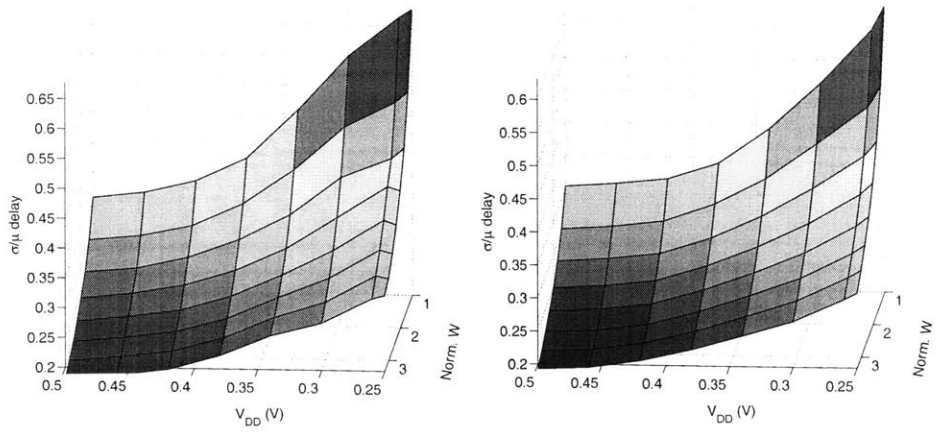
(a) (b)

Figure 2-7: (a) Monte Carlo setup for current variability measurement. (b) Active current variability of different CMOS primitives versus device width (normalized to minimum size) at V_{DD}=300mV.

noise margin rather than current variability should be considered first in device sizing decisions.

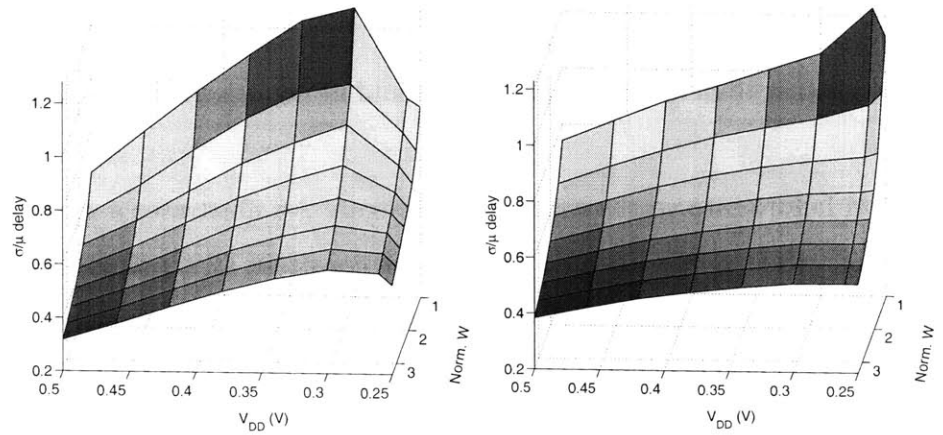
The SNM failure rate versus width plot of Figure 2-6 illustrates a sizing methodology for single and stacked devices. Suppose we constrain all topologies to have the same failure rate, or interchangeably, a constant yield. We obtain the required device sizes by drawing a horizontal line at the desired failure rate, then finding where this line intersects the failure curve and the corresponding x-axis value. In the example of Figure 2-6, a target failure rate of 0.13% requires a single and 2-stack NMOS to be sized at 2 and 4.43 times minimum width respectively. The 2-stack sizing here can be used for any static CMOS gate with two series NMOS, since it was derived from NAND2 which exhibits the worst-case V_{OL} due to two leaking parallel devices in the pull-up network.

Because the failure rate reduces at higher V_{DD}, the required size for a given yield constraint also decreases. This is seen in Table 2.1, which lists device widths for a constant failure rate of 0.13% at discrete values of V_{DD}. It is interesting to note that in transmission gates, minimum size devices are sufficient even at $V_{DD} = 0.24V$. This



(a) (b)

Figure 2-8: Pull-up delay variability versus V_{DD} and width (normalized to minimum size) of (a) inverter (single PMOS) and (b) NOR2 (two series PMOS).



(a) (b)

Figure 2-9: Pull-down delay variability versus V_{DD} and width (normalized to minimum size) of (a) inverter (single NMOS) and (b) NAND2 (two series NMOS).

suggests that for the same functional yield, transmission gate-based logic requires smaller device widths than stacked static CMOS topologies, which in turn lead to energy savings.

0.13% represents the 3σ tail of a normal distribution and is chosen for demonstration. It should be noted that a target of 0.13% provides a consistent guideline

for sizing various logic gates, but does not relate in a straightforward way to the failure rate of a circuit built from these gates. As mentioned previously, this value is a pessimistic estimate because it assumes that every second gate in the circuit is NAND3 or NOR3. Furthermore, failing logic gates will tend to cluster on the die at process corners, thus overall die yield will likely be higher. Modeling failure distribution across global process conditions will help in setting a more accurate target yield for standard cell design.

Table 2.1: Required widths (normalized to minimum size) versus V_{DD} for constant failure rate=0.13%.

$V_{DD}(V)$	0.24	0.26	0.28	0.30	0.32	0.34
1-NMOS	1.83	1.27	1	1	1	1
2-NMOS	4.43	2.93	2.3	2.27	1.3	1
1-PMOS	1.63	1.03	1	1	1	1
2-PMOS	1.63	1	1	1	1	1
TG	1	1	1	1	1	1

Chapter 3

Sub-threshold Standard Cell Library

General sizing and topology considerations have been discussed in Chapter 2. Here we present implementation details of the sub-threshold standard cell library designed in a 65nm CMOS process. We address operating specifications of the library, selection of available logic functions, sub-threshold register design issues, and sizing for logic gates of various drive strengths. Analysis in Chapter 2 and Chapter 3 culminated in a sub-threshold standard cell library of 56 cells implementing 24 distinct logic functions.

3.1 Library Specifications

The cell library is intended as a general-purpose library for use in a CAD flow for building sub-threshold digital circuits. Since the supply voltage in the eventual application is unknown, a target minimum operating voltage must be specified *a priori*. The library must also be designed for a range of temperature and process conditions.

The temperature range is set at 0 to 70°C, a standard for commercial products. Standard cells are verified at global process corners using the methodology described in Section 3.5.1.

The target V_{DD} should accommodate the optimum supply voltage V_{DDopt} for a variety of systems. A reasonable lower bound V_{DDopt} in this particular process is

obtained from characterization of a ring oscillator. The oscillator consists of a chain of 2-input NAND and NOR gates rather than inverters to better emulate the stack effect and leakage currents in an actual circuit. The movement of V_{DDopt} with relative significance of switching energy can be captured by varying the activity factor of this characterization circuit [6]. In simulation, this can be achieved by separately characterizing the switching and leakage energy components of the ring oscillator, then multiplying the switching energy by the activity factor. This is effectively the same as reducing the number of nodes being switched per cycle while including leakage through the entire circuit.

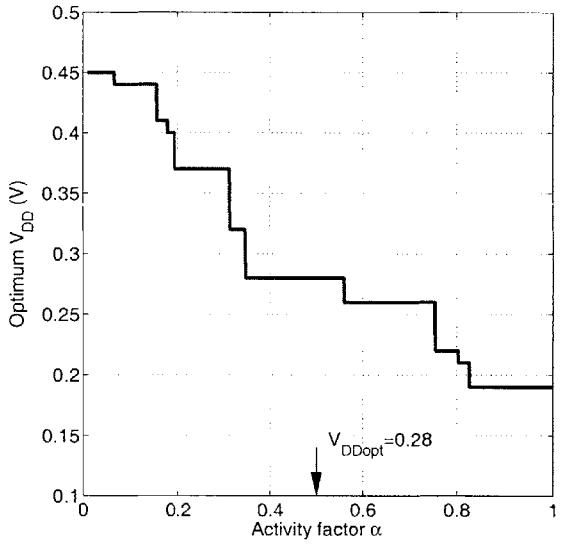


Figure 3-1: Optimum V_{DD} for minimum energy in a 65nm ring oscillator characterization circuit, plotted against activity factor.

Figure 3-1 plots V_{DDopt} versus the activity factor α. A reasonable upper bound on α remains to be selected. Different systems have widely varying activity factors, and definitive values are difficult to obtain from literature. Table 3.1 lists results for a 90nm microprocessor [3], where the activity factor of each node in the logic core is recorded from extensive simulation. A large majority of nodes have activity factors ≤ 0.5. From Figure 3-1, the optimum V_{DD} corresponding to $\alpha = 0.5$ is 280mV in this technology. We select a minimum V_{DD} specification of 250mV for a 30mV design margin.

Table 3.1: Node activity factors in commercial microprocessor core [3].

Activity Factor Bin	1>=AF≥0.5	0.5>AF>0.001	0.001>AF
Percentage of Nodes (%)	1.9	20.0	78.1

3.2 Logic Function Selection

The number of available logic functions in a standard cell library affects the efficiency of the synthesis tool in performing logic mapping and optimization. A reduced library [30] with 22 logic functions in 92 cells reportedly allows the synthesis tool to achieve either lower delay, area, or power compared to a full library. The cells are chosen from a large number of experiments where circuits are synthesized with and without a specific cell. Work in [13] compared a 0.18μm commercial cell library and one modified for reduced V_{DD} operation. The modified library contains resized flip-flops and omits complex gates with large stacks to enable 300mV operation at process corners. It was found that restricting the cell set without corresponding optimization in the synthesis tool led to a 50% energy overhead at the typical corner and only 10% energy savings at the worst case corner.

Drawing from these previous findings, we adopt the strategy of providing all two- and three-input logic functions normally available in commercial libraries. We observed that the addition of AOI (and-or-invert), OAI (or-and-invert), NAND3, NOR3, and full adders to the library enabled the synthesis tool to reduce circuit area and thus switching energy from lower interconnect capacitance. One drawback is that NAND3 and NOR3 impose more stringent requirements on V_{OH} and V_{OL} of all cells as addressed in Section 2.2.1, and in some cases require slight upsizing.

In terms of sequential cells, the library provides a standard D-type register and variations for negative edge-triggering, reset, and preset. Active-high and active-low D-type latches are also included.

3.3 Sub-threshold Register Design

Sub-threshold register design merits special attention because idle leakage currents can significantly degrade data retention capabilities. Process variation in V_T affects leakage currents exponentially, causing further uncertainty in reliability of sub-threshold registers. We first examine several designs commonly used in the strong inversion regime, including dynamic and static styles. It is shown that dynamic registers cannot retain state consistently in the presence of process variation. We then compare two static register designs in terms of nominal performance and failure mechanisms under V_T mismatch. Lastly, we characterize distributions of timing parameters such as setup, clock-to-output, and hold time for insight into how registers affect circuit delay variability.

3.3.1 Register Logic Styles

Many register designs have been reported in literature, targeting various constraints such as high speed, low power, or immunity to race conditions. Most registers can be loosely classified into static and dynamic [31], which refer to whether state is retained by a statically driven or dynamic node. Other types include pulse-based registers, which latch data during a short pulse generated around a clock edge, and designs with conditional clock gating when input and output data are equal. They are typically used in custom-designed systems and are not examined in this section.

Dynamic Registers

Dynamic registers are targeted for high-speed applications in above-threshold design. Trading noise margin for speed, the dynamic storage node is susceptible to charge leakage. The problem is exacerbated in sub-threshold because of the small amount of stored charge, exponentially varying leakage currents, and long clock periods over which state must be retained. To verify these observations, we simulate a simple dynamic register, the clocked CMOS [32] (C^2MOS) design shown in Figure 3-2.

Figure 3-3(a) shows the register operating at the weak-NMOS, strong-PMOS

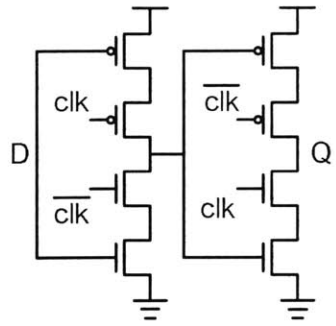
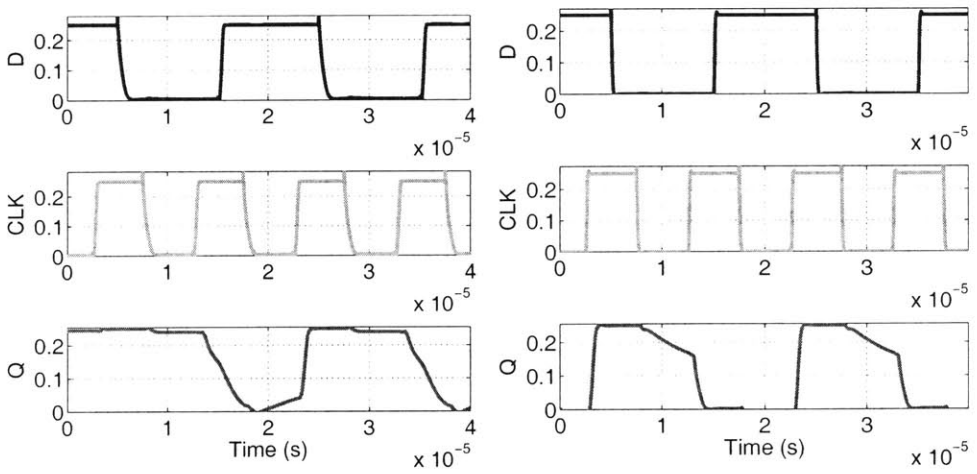


Figure 3-2: C^2MOS register.

global process corner with no local mismatch. NMOS device widths are sized to be two times PMOS widths to strengthen the pull-down network and offset the stronger PMOS currents. At 20μs, the register output Q rises in voltage gradually during the low phase of CLK due to leakage through PMOS to the dynamic node. In this simulation, half the clock period is occupied by the clock-to-output delay. If the period is extended to accommodate other gate delays in the critical path, the voltage level of Q when storing a 0 would be further degraded.



(a) (b)

Figure 3-3: (a) C^2MOS at weak-NMOS, strong-PMOS corner. (b) C^2MOS at typical process corner with V_T mismatch.

The effect of local mismatch is illustrated in Figure 3-3(b). The same register is

simulated at the typical process corner, but with random V_T mismatch applied to all transistors. In this case, mismatch strengthens the NMOS transistors such that the output voltage Q droops during the low phase of the clock, when the slave latch is in hold mode. These two examples show that state retention of dynamic registers can be significantly compromised by process variation. It is possible to make the register pseudo-static by attaching cross-coupled inverters to the storage node. However, the clocked inverter must then rely on ratioed device sizing to overpower the cross-coupled inverters.

Static Registers

Common static registers [31] include the multiplexer-based transmission gate (TG-MUX) register and its variants. The first variant is a ratioed version with reduced clock load (RCL), and the second is a non-ratioed version used in the PowerPC 603 processor (PPC) [2], with a transmission gate and inverter being converted into a clocked inverter. It was noted in [13] that the RCL register, and ratioed logic in general, are not robust at process corners because sub-threshold currents have linear dependence on transistor size and exponential dependence on V_T. Therefore, we select only the TG-MUX and PPC registers for comparison. They are shown in Figure 3-4 and Figure 3-5 respectively.

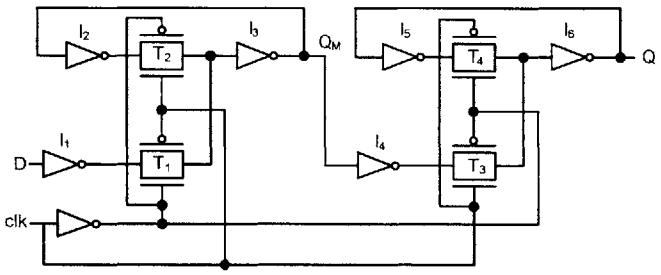


Figure 3-4: Multiplexer-based transmission gate register.

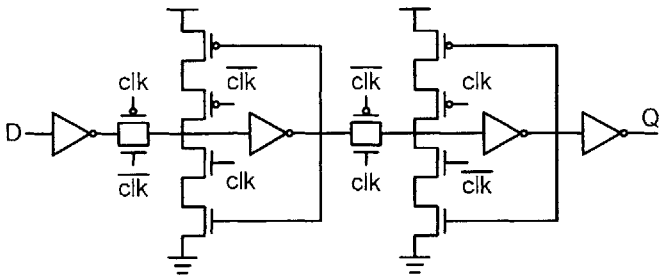


Figure 3-5: PowerPC 603 static register [2].

3.3.2 Register Comparison

Delay and Energy

Since dynamic registers are shown to be unreliable in sub-threshold, we consider static registers and compare the nominal delay and energy performance of TG-MUX and PPC. Both registers are sized to have equal energy. Table 3.2 lists the delay and energy parameters of both registers under nominal process conditions at 0.25V. Setup and hold times are simulated according to [33] by moving the data edge with respect to clock edge until the clock-to-output delay t_{cq} reaches 1.05 times its nominal value. Clock and data buffers are used to condition slopes of input signals to the register and are included in energy measurements to account for the effect of clock and data loading. For equal energy, TG-MUX exhibits a clear delay advantage over PPC.

Static Noise Margin

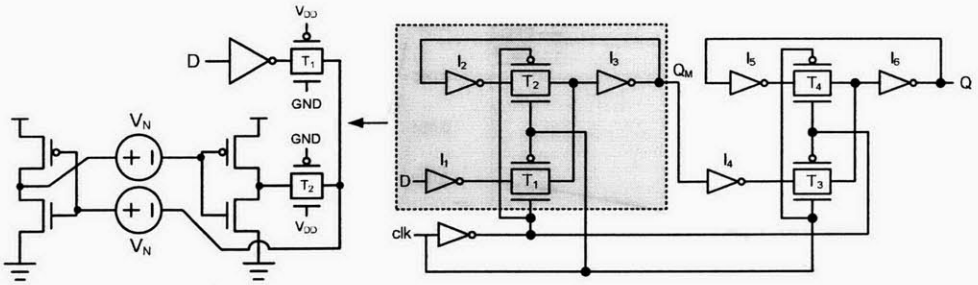
The concept of noise margin is also relevant in sub-threshold register design, where data retention is a particular challenge. Similar to SRAM cells, data retention capability of the register is reflected in the hold static noise margin of its cross-coupled inverters. Figure 3-6(a) shows the equivalent circuit for measuring the register SNM in TG-MUX, accounting for the voltage drop across T_2 and the worst case leakage across T_1. Figure 3-6(b) shows a sample butterfly plot, with the regular and mirrored VTCs of the two half-circuits separated by V_N. SNM is equal to the length of the smaller inscribed square.

Table 3.2: Performance comparison of two static registers. t_{su}, t_{cq}, and t_h denote setup, clock-to-output, and hold time respectively.

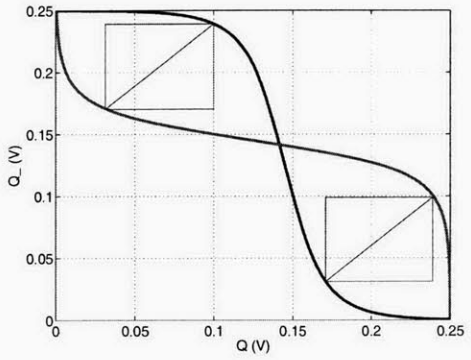
		TG-MUX	PPC
$t_{su}[\mu s]$	rise	0.5195	0.9860
	fall	0.2583	0.3695
$t_{cq}[\mu s]$	rise	0.6431	0.7950
	fall	0.5015	1.178
$t_h[\mu s]$	rise	0.09094	0.1748
	fall	-0.1943	0.1235
$t_d = t_{su} + t_{cq}[\mu s]$	average	0.9612	1.664
$E_{DYN}[fF]$		0.6731	0.6629
$E_{LEAK}[fF]$		0.02479	0.03615
$E_T[fF]$		0.6979	0.6990

Figure 3-7(a) and Figure 3-7(b) compare the nominal static noise margin of the two registers, varying width and length of both PMOS and NMOS in the critical cross-coupled inverters respectively. SNM does not change significantly with width, but does vary approximately 10mV with length. This is due to short channel effects affecting PMOS and NMOS differently as length increases, shifting VTCs in the butterfly plot to cause a change in SNM. The nominal SNM of the two registers differ only by several millivolts.

Because nominal SNM for data retention does not change significantly with device sizing, we compare the effect of V_T mismatch on PPC and TG-MUX latches with uniformly sized transistors in a 1k-point Monte Carlo simulation. As seen in Table 3.3, 1.1% of the PPC latch samples have negative SNM, while none of the TG-MUX latches failed. To reduce the number of failing samples, the stack of two transistors in the PPC latch would need to be upsized, further increasing the energy consumption of PPC. Table 3.2 and Table 3.3 together indicate that the transmission gate design is more energy-efficient under the constraint of equal SNM failure rate.



(a)



(b)

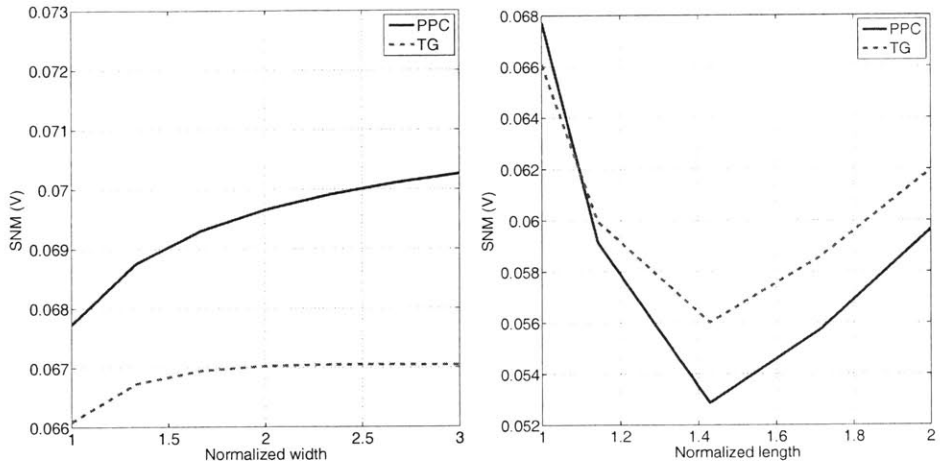
Figure 3-6: (a) TG-MUX schematic and equivalent circuit for measuring SNM. (b) Butterfly plot of master latch in TG-MUX. Length of inscribed square is equal to the static noise margin.

3.3.3 Detailed Design Considerations

Based on Section 3.3.1 and Section 3.3.2, we select the multiplexer-based transmission gate design for the sub-threshold library. The following presents detailed design considerations of this register. The general design strategy is to start with minimum size transistors and upsize when necessary based on local mismatch simulation results.

Latch Transistor Sizing

The cross-coupled inverters in the master and slave latches are sized for data retention robustness under V_T mismatch. The required transistor sizing for a given target failure rate is again estimated by counting the number of samples with negative SNM in a 10-



(a) (b)

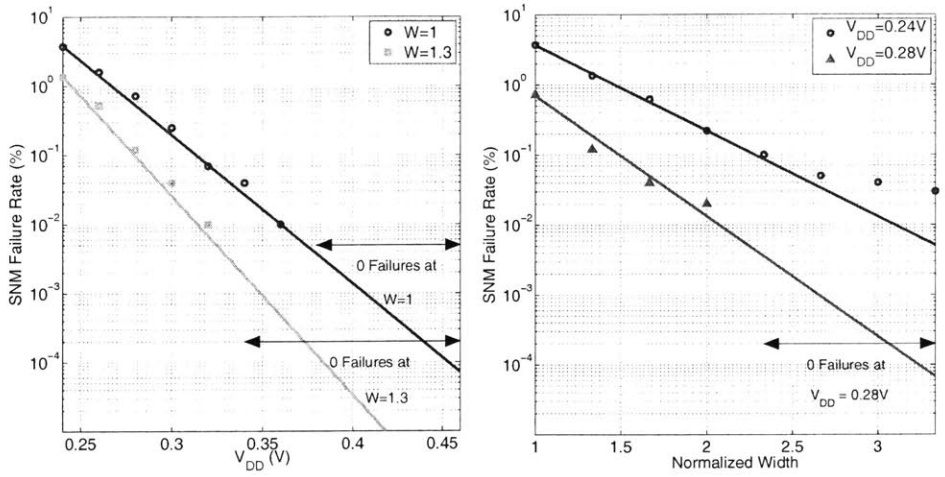
Figure 3-7: Nominal register SNM of PPC and TG-MUX versus (a) width and (b) length of PMOS and NMOS.

Table 3.3: Number of latches with negative SNM from 1000 Monte Carlo runs, performed at the typical process corner.

	TG-MUX	PPC
Number of samples with SNM < 0	0	11
Estimated SNM failure rate	0%	1.1%

k point Monte Carlo simulation, using the setup of Figure 3-6(a). Figure 3-8(a) and Figure 3-8(b) plot the failure rate versus V_{DD} and device width in the cross-coupled inverters. Solid lines in Figure 3-8(a) are fitted to measured data points in markers, and show that failure rate decreases exponentially with increasing V_{DD}. Furthermore, we observe a steeper slope when the normalized device width is 1.3 compared to 1. Similar trends are observed in Figure 3-8(b).

Figure 3-9 plots the transient simulation corresponding to a register with negative SNM. The register schematic with labeled nodes is shown in Figure 3-10 for convenience. The master latch is transparent after the first falling edge of clock. Nodes T1, NT1, and T2 are driven according to the value of D. However, the logic 0 voltage of node T2 is at 90mV, which corresponds to a VTC shifted upwards in a butterfly plot



(a) (b)

Figure 3-8: SNM failure rate versus (a) V_{DD} and (b) device width of cross-coupled inverters.

measuring static noise margin, and in this case leads to a negative SNM. After the second rising edge of CLK, node T1 should stay near ground when the master latch enters hold mode. However, because of negative SNM in the master latch, nodes T1, NT1, and T2 flip to the monostable state.

Sensitivity to V_T Mismatch

.

Monte Carlo simulations of the entire TG-MUX register reveal that V_T variation in transistors outside of the latches can also cause the register to be non-functional. To quantify the relative sensitivities of register functionality to mismatch in different components, we perform transient simulations under typical process conditions and only vary V_T of one component at a time. For example, V_T of the input buffer I_1, or the feedback transmission gate TG_3, is gradually shifted away from the nominal value towards the worst case condition until the register ceases to function. For simplicity, all transistors are uniformly sized and the V_T of both PMOS and NMOS are varied together in each component. The worst case V_T mismatch condition for an inverter is when PMOS is strengthened and NMOS is weakened, or vice versa. The worst

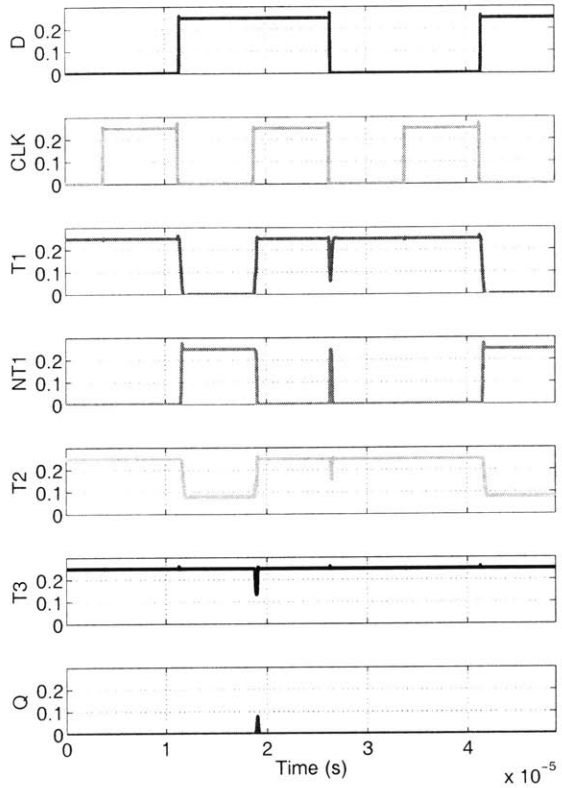


Figure 3-9: Transient waveform for register with negative SNM.

condition for a transmission gate occurs when both PMOS and NMOS are weakened.

The deviation of V_T which first leads to register failure is listed in Table 3.4. A register fails when it outputs incorrect data, either because its input or output buffers have insufficient output swing, or because its latches cannot retain state. For example, if the V_T of PMOS in the input buffer is over 2.4 standard deviations stronger than the nominal value and the NMOS is more than 2.4σ weaker, then the register fails to store correct data. A large value in Table 3.4 indicates that the component is relatively insensitive to V_T mismatch.

The relative sensitivities in Table 3.4 are consistent with register failure modes observed in Monte Carlo transient simulation with V_T variation applied to all transistors. The primary failure mode is the inability of master or slave latch to retain data due to negative SNM, as shown previously in Figure 3-9. Another failure mode occurs when local clock buffers do not produce a clock signal of sufficient swing. This

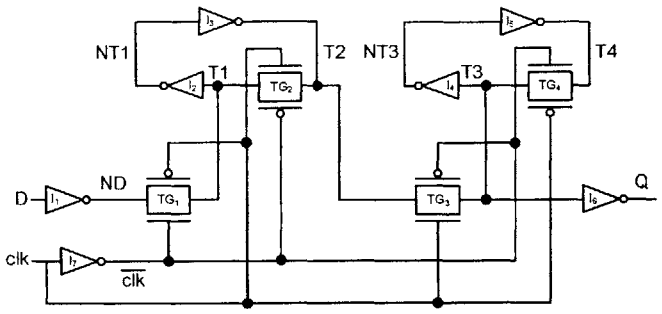


Figure 3-10: Multiplexer-based transmission gate register with labeled nodes.

Table 3.4: Sensitivity of register components to V_T variation. V_T of each component is varied from the nominal value until register outputs incorrect data.

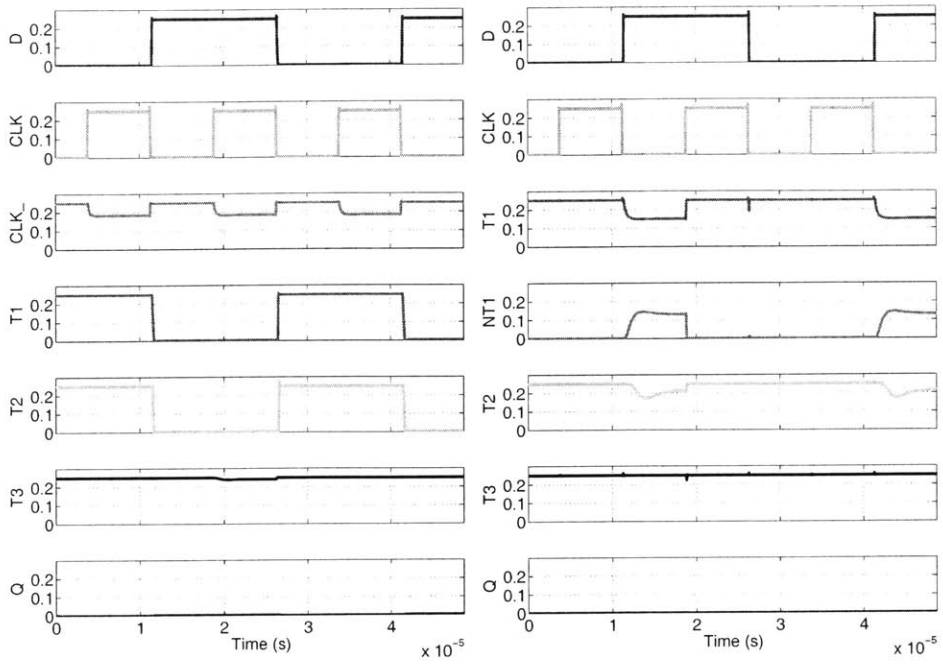
Register Component	Maximum V_T deviation from nominal before register outputs incorrect data (normalized to one standard deviation)
Input buffer (I_1)	2.4
Isolation TG (TG_1)	4.9
Latch inverter (I_2)	0.8
Feedback TG (TG_2)	6.0
Local clock buffer (I_7)	1.4

is illustrated in Figure 3-11(a), where $\overline{CLK}$ exhibits degraded logic 0 level. In this particular simulation, the NMOS in TG_1 is strengthened by mismatch so TG_1 is able to pass the appropriate voltage to node T1, even though it is driven by a degraded clock signal. However, TG_3 is weakened by V_T mismatch, and when driven by a poor $\overline{CLK}$ signal, is unable to drive node T3.

Figure 3-11(b) demonstrates the third dominant failure mode, where V_T mismatch in I_1 limits the voltage swing at node T1 during the low phase of CLK. The master latch does not settle to the correct state on the second rising edge of CLK.

3.3.4 Timing Parameter Distribution

This section presents distributions of TG-MUX timing parameters in sub-threshold under V_T mismatch, which can be used to determine design margins during timing analysis of a synthesized circuit. Setup time, hold time, and clock-to-output delay

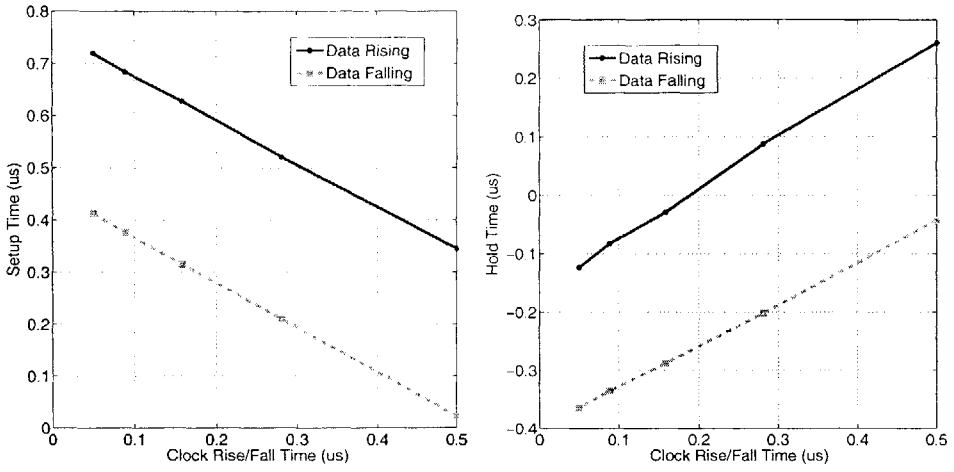


(a) (b)

Figure 3-11: Transient waveform when V_T mismatch in (a) local clock buffer I_7 and (b) input buffer I_1 causes non-functionality.

are the three parameters of interest. Setup time is defined according to [34] as the smallest time between data and clock edges ($t_d - t_{clk}$) such that the clock-to-output delay t_{cq} does not exceed 105% of its nominal value. Similarly, hold time is defined as the smallest amount of time data has to be kept constant after the rising edge of clock. Both parameters are measured using the optimization function in SPICE to move the data edge relative to clock, until t_{cq} reaches 105% of its nominal value.

Figure 3-12(a) and Figure 3-12(b) plot the nominal setup and hold time versus rise/fall time of the clock. Both setup and hold times display a linear relationship with clock slew rate. Setup time has a negative dependence on clock rise/fall times, since a slower clock edge allows more time for the data signal to be propagated through the master latch. Hold time has a positive dependence, since a slow clock edge causes the master latch to be transparent for a longer period of time. Therefore, data must be held constant longer to avoid a race condition.



(a) (b)

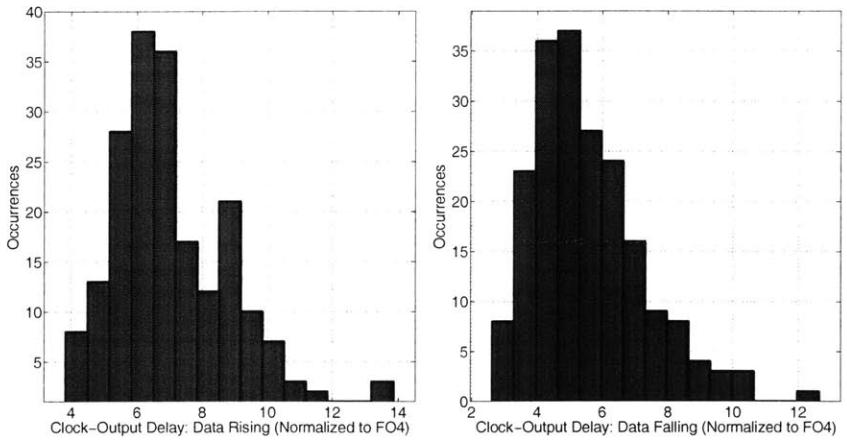
Figure 3-12: (a) Setup and (b) hold time of TG-MUX register versus rise/fall time of clock. $V_{DD} = 0.25$V.

A histogram of clock-to-output delay from a 200-point Monte Carlo simulation is shown in Figure 3-13. In the TG-MUX register, t_{cq} is approximately equal to the delay through TG_3 and I_6 of Figure 3-4. The t_{cq} distribution is thus similar to the delay distribution through a logic gate, following a lognormal trend.

Figure 3-14 plots the setup and hold time distributions of TG-MUX in sub-threshold. The setup time is approximately given by the delay through I_1, T_1, I_2, and I_3 from Figure 3-4. Again, the distribution is roughly lognormal. The hold time of TG-MUX is theoretically 0, since changes in D cannot affect storage nodes in the master latch after the rising edge of CLK. However, because T_1 may not be turned off instantaneously by a clock with finite slew rate, we observe positive hold time in the majority of registers. The distribution of hold time has negative skew (towards the right) and does not appear lognormal.

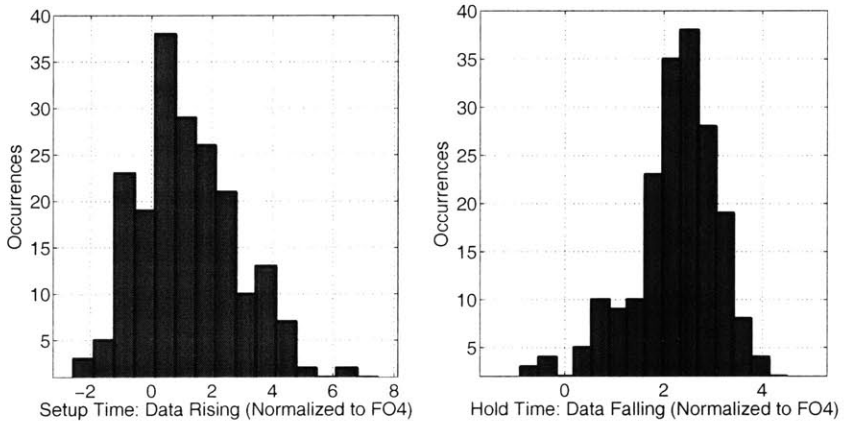
3.4 Drive Strength Design

The sub-threshold library provides several drive strengths for each logic function, to be used in driving differently sized loads. Commonly used cells have more drive



(a) (b)

Figure 3-13: Clock-to-output delay distribution for (a) data rising and (b) falling. $V_{DD} = 0.25$V.



(a) (b)

Figure 3-14: Distributions of (a) setup time (data rising) and (b) hold time (data falling). $V_{DD} = 0.25$V.

strength options available, while less common cells may need to be buffered in order to drive large loads.

Transistor sizing for different drive strengths has not been well-discussed in literature. This library adopts several approaches depending on the cell topology. Cells of different drive strengths are compared with respect to their ability to drive integer

multiples of a standard load (SL). As per [34] which suggests using a small standard load for low-energy design, we define one SL to equal twice the input capacitance of the smallest inverter in the library (INVX1). The following discussion refers to the weakest cell as $X1$ and stronger cells as Xn, where n is an integer denoting the drive strength.

3.4.1 Single-Stage Gates

From the base inverter (INVX1), we obtain inverters of higher drive strengths by increasing both NMOS and PMOS sizes by the same scalar factor. For example, INVX2 has devices twice as wide as those of INVX1.

NAND and NOR gates are sized such that propagation delays of the stronger gate driving a large load are comparable to those of the $X1$ gate driving a smaller load. The ratio of the two loads is given by n, the numeric label of the drive strength. For example, a NAND2X4 driving 16SL would be sized to give the same propagation delay as a NAND2X1 driving 4SL. Pull-up and pull-down networks are sized separately.

3.4.2 Multiple-Stage Gates

AND and OR are the two main double-stage static gates in the cell library, while cells based on transmission gate topologies (e.g. XOR, MUX, AOI) are classified as multiple-stage gates in the library. The general strategy for sizing these gates is to make the output stage identical to the single-stage gate of the same drive strength, and use the concept of logical effort to minimize delay through all stages of the gate. Transmission gates are kept to the minimum width allowable given variability constraints, since it was determined experimentally that increasing their sizes provides no delay or energy benefits.

A thorough treatment of logical effort is given by [35]. In this discussion, we point out some issues specific to applying this method in sub-threshold design. The logical effort of a gate is defined as the ratio of input capacitance of the gate to that of an inverter which delivers the same output current. In above-threshold design, the logical

effort can be estimated for a given circuit topology by linearly increasing widths of series transistors to achieve equal *on* resistance as an inverter. This assumption is not accurate in the velocity saturation or sub-threshold regimes. The actual logical effort can be characterized by measuring the slope of propagation delay versus fanout for an inverter and the logic gate in question. The logical effort of the gate is then the ratio $\frac{\text{slope for logic gate}}{\text{slope for inverter}}$ [35].

Table 3.5 lists logical efforts for NAND2 and NOR2 gates at various supply voltages. Values in each row are normalized to the smallest value in the row to show the relative differences. The logical effort varies up to 60% across sub-threshold voltages and may also vary significantly between 3σ process corners, as seen in Table 3.6. Since the operating voltage of the target application is not known, we average the logical effort across V_{DD} at the typical corner during cell design.

Table 3.5: Logical effort variation across V_{DD}.

$V_{DD}(V)$	0.25	0.30	0.40	0.50	1.20
NAND2	1.330	1.466	1.586	1.503	1.000
NOR2	1.000	1.010	1.123	1.375	1.321

Table 3.6: NAND2 logical effort variation between typical (TT), weak-NMOS strong-PMOS (WS), and strong-NMOS weak-PMOS (SW) process corners.

$V_{DD}(V)$	0.25	0.30	0.40	0.50	1.20
NAND2, TT	1.330	1.466	1.586	1.503	1.000
NAND2, WS	1.514	1.684	2.351	2.137	1.000
NAND2, SW	1.264	1.154	1.092	1.021	1.000

3.5 Design Tool Considerations

Standard cells in the sub-threshold library are designed according the methodology outlined in Chapter 2 and Chapter 3. The standard cells are then incorporated in a commercial computer-aided design (CAD) flow. This enables circuits implemented in a hardware description language, such as Verilog, to be synthesized for sub-threshold

operation. This section describes the preparation process for each cell, the CAD flow, and issues related to automated synthesis of sub-threshold circuits.

3.5.1 Cell Verification Methodology

Following the logic function and drive strength selection criteria described previously, 56 cells have been designed and tested to date. A list of cells is provided in Appendix A. Figure 3-15 illustrates the design and verification process for each cell, performed at 0.25V. The general flow is similar to above-threshold cell design except for steps 2 and 5. The Monte Carlo test for V_{OH}, V_{OL} in step 2 follows the guidelines of Section 2.2.1 by verifying the gate output levels against NAND3 and NOR3. Transistors are upsized until a failure rate of $< 0.1\%$ is achieved at the worst case temperature and process corner. In step 5, we find the 3σ lower and upper bounds on cell propagation delay distribution, which is intended to account for on-chip variation in timing analysis. It should be reiterated that 99.9% yield is a pessimistic estimate. As mentioned in Section 2.2.2, functional errors are much more likely to occur on corner die than on typical die, so the actual yield across process conditions will be higher. Further analysis is necessary to relate yield at process corner to overall yield across wafers.

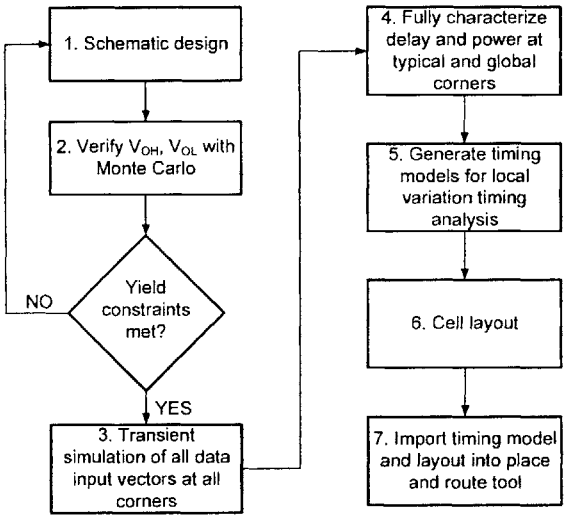


Figure 3-15: Design and verification process for sub-threshold standard cells.

3.5.2 Computer-Aided Design Flow

The design flow shown in Figure 3-16 is used in conjunction with the sub-threshold library to synthesize the test chip of Chapter 6. The general flow is similar to that of above-threshold design. However, commercial tools do not account for large performance variabilities seen in sub-threshold, which introduces additional challenges at all stages of the design flow.

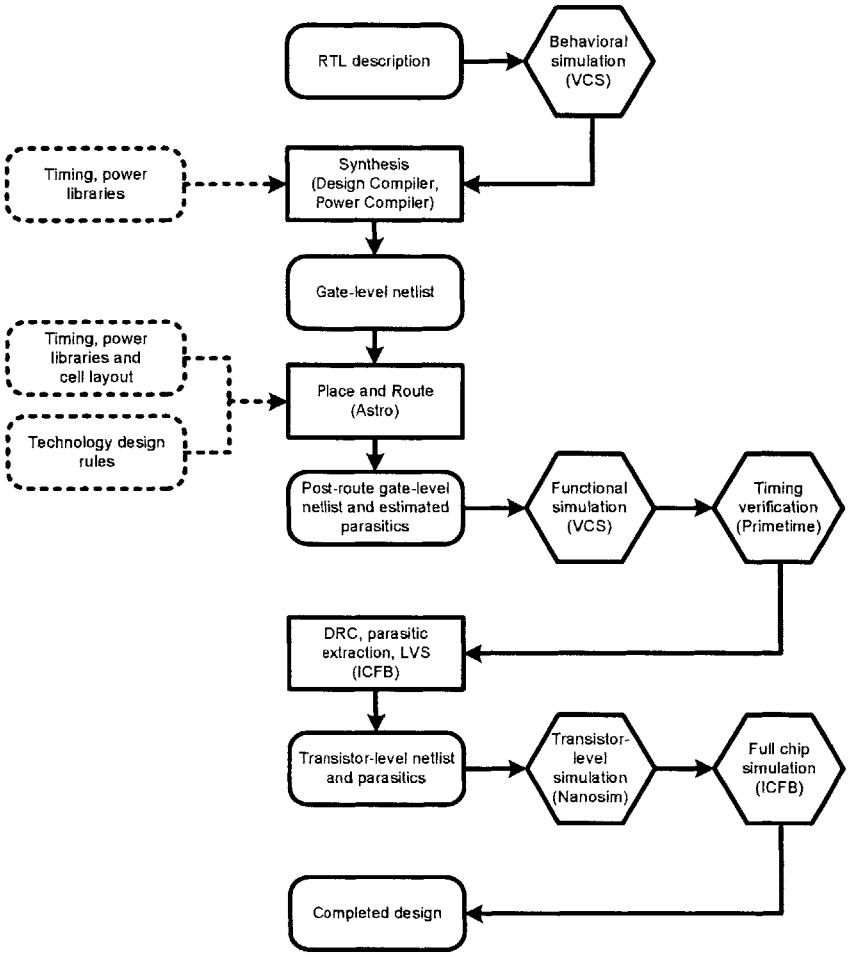


Figure 3-16: Computer-aided design flow for sub-threshold library (name of tool given in parentheses).

Synthesis and Layout

Tools for performing synthesis as well as place and route rely on timing and power information of each cell in order to meet performance constraints set by the designer. These performance libraries are created during the design of each cell by characterizing the delay and power across a range of input and output conditions, at global temperature, voltage, and process corners. The best and worst case delay information is used to select appropriate cell drive strengths to meet timing constraints during the synthesis stage. During place and route, the information is used for clock tree synthesis, cell resizing, and delay insertion to resolve timing violations.

In typical above-threshold design, corner conditions may cause performance deviations on the order of $\pm 20\%$. However, the same range of conditions leads to an order of magnitude higher variation in sub-threshold. With a large performance spread, it is infeasible to synthesize a design for the same set of constraints across corner conditions, as is the case in current tools. Therefore, to improve energy efficiency of designs synthesized from sub-threshold libraries, CAD tools should allow the designer to define multiple sets of constraints for different corner conditions.

Timing Verification

Another important challenge in computer-aided design of sub-threshold circuits is timing verification, or ensuring that setup and hold time constraints are met for all timing paths in a circuit. Unlike above-threshold systems which typically operate at the nominal voltage of the process technology, a sub-threshold circuit is expected to operate at its optimum supply voltage, which varies with external conditions such as temperature and workload. In an ultra-dynamic voltage scaling scenario (UDVS) [36], a circuit synthesized with sub-threshold cells will also be operated in above-threshold. Thus timing verification must be performed across the full range of voltages under which the circuit will operate. Using current tools, this requires timing libraries to be characterized at many supply voltages, a time consuming process. The characterization effort can be reduced if timing verification tools model the scaling of delay with

V_{DD} accurately, particularly in the sub-threshold region.

Random within-die variation causes further difficulties with standard timing verification tools, which treat logic delays as deterministic. Current commercial tools accept timing libraries for best and worst case corners within a die. To check setup time, they use the worst case gate delays along the logic path and best case delays for clock buffers, and vice versa for hold time. This approach is overly conservative, since it is unlikely that random local variation would cause all logic gates in a timing path to have best case delays. The problem is exacerbated at low supply voltages, where the 3σ delay of a single logic gate can be several times the mean value. Using $\pm 3\sigma$ bounds for within-die corner libraries, as is customary in above-threshold, will clearly lead to many false timing violations in sub-threshold.

In this thesis, timing verification is performed by obtaining the longest and shortest paths in a circuit from a commercial tool such as Synopsys Primetime, then manually performing Monte Carlo simulation of the path in SPICE to extract the delay distribution. Clock distribution network delay is obtained similarly. Special attention is given to hold time violations, which cause non-functionality regardless of the system frequency, while setup time violations can be overcome by extending the clock period. Therefore, we characterize the clock skew distribution between two consecutive registers separated by a short logic path, and verify that the relation $t_{skew} < t_{clock-q} + t_{logic,cd} - t_{hold}$ ([31]) is satisfied with high probability. Delay cells are inserted. Delay cells are inserted as necessary. Figure 3-17 shows an example clock skew distribution between two registers at 0.25V.

This approach is cumbersome for designs with many short timing paths or a complex clock distribution network. There is a definite need for automated timing tools that treat gate delays as statistical distributions and verify timing to a specified confidence level. As such, statistical timing analysis (e.g. [37], [38], [39]) has recently become an active area of research.

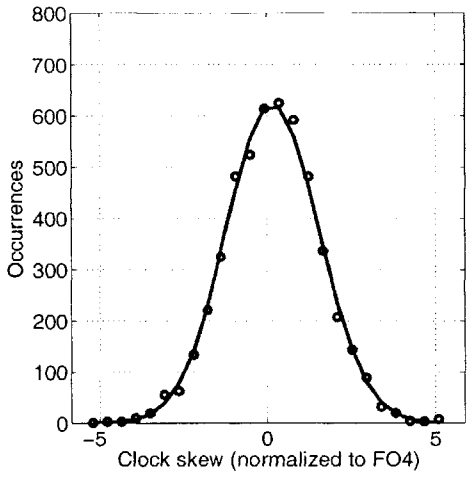


Figure 3-17: Distribution of clock skew between two consecutive registers at 0.25V, normalized to FO4 delay of a minimum size inverter.

Functional Verification

Since sub-threshold circuits rely on small leakage currents, accurate functional simulation in a reasonable time becomes difficult for a large circuit. SPICE simulations with full BSIM models are prohibitively expensive for tests longer than several clock cycles. The Synopsys Nanosim simulator is able to trade-off speed and accuracy by reducing complexity of device models and partitioning the netlist. For the 65nm process technology in this thesis, a medium accuracy setting enabled functional simulation in a reasonable time. However, the total leakage current measurement in Nanosim did not exactly match the more accurate value provided by SPICE, and only gave an estimate to the same order of magnitude. Furthermore, some anomalies were observed when simulating a large test circuit (10,000 transistors) with parasitics, which were resolved by manually setting higher accuracy options on specific nodes. It is uncertain whether Nanosim can provide sufficiently accurate results when device model complexity increases in future process technologies. If not, sub-threshold circuit design must then rely on gate-level functional simulation and accurate timing analysis tools instead of transistor-level simulations. This further motivates development of statistical timing analysis methodologies.

Chapter 4

Minimum Energy Operation With Process Variation

The total energy per operation consumed by an arbitrary circuit is modeled in [7] as

$$E_T = E_{DYN} + E_L = C_{eff} V_{DD}^2 + W_{eff} I_{leak} V_{DD} t_d L_{DP}. \tag{4.1}$$

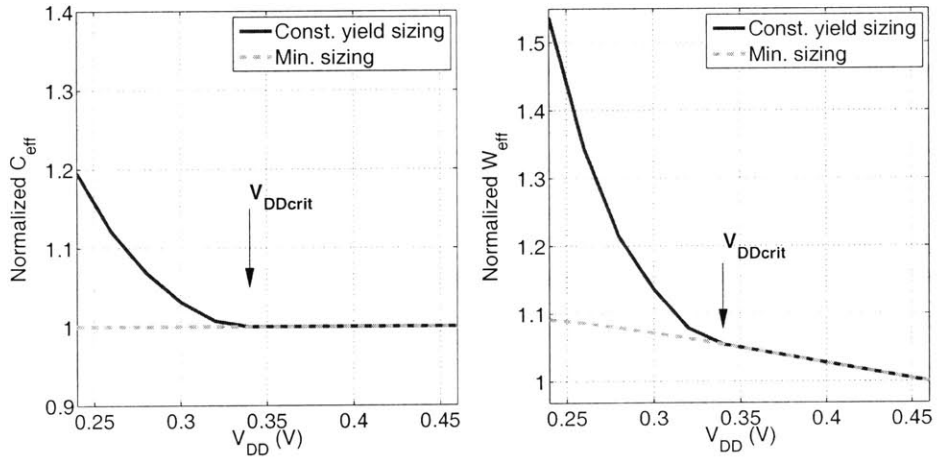
As described in Section 1.3.2, opposing trends in E_{DYN} and E_L with decreasing V_{DD} give rise to an optimal supply voltage V_{DDopt} at which total energy is minimized, assuming the circuit is functional.

Section 2.2 has shown that functionality is no longer guaranteed at low supply voltages when V_T variation is significant, even in typically robust static CMOS circuits. Reducing the probability of logic failure requires either upsizing devices or increasing V_{DD}, which must be considered when finding V_{DDopt}. This can be accounted for within the framework of [7] by treating C_{eff} and W_{eff} as a function of V_{DD}. The resulting energy versus V_{DD} characteristic of an inverter chain and 32-bit Kogge-Stone adder are simulated in a 65nm process and presented as examples.

4.1 Minimum Energy Point with Yield Constraint

Figure 4-1 plots C_{eff} and W_{eff} versus V_{DD} for the Kogge-Stone adder under two sizing schemes. The solid line represents constant yield sizing of Table 2.1 in Section 2.2.2, while the dashed line indicates an adder with only minimum size devices. Note that W_{eff} is obtained by normalizing the adder leakage current to that of a characteristic inverter [7]. DIBL affects leakage through the two circuits differently as V_{DD} decreases, causing a slight increase in W_{eff} in this case. V_{DDcrit} denotes the critical operating voltage at which minimum size devices can be used to satisfy the yield constraint. When $V_{DD} \geq V_{DDcrit}$, the circuit under both schemes are identical.

It should be noted that once the yield constraint is set, V_{DDcrit} can be found immediately from Table 2.1 and the topology of a given circuit. For example, a circuit without stacked devices does not require upsizing when $V_{DD} \geq V_{DDcrit} =$ 300mV. In contrast, a circuit with stacks of two NMOS has $V_{DDcrit} = 340$mV.



(a) (b)

Figure 4-1: (a) C_{eff} and (b) W_{eff} for adder with constant yield and minimum sizing.

The switching, leakage, and total energy of the inverter chain and adder are then calculated according to Equation 4.1. Figure 4-2 plots the energy versus V_{DD} characteristic of the inverter chain at nominal process corner and temperature. Total energy

in both constant yield and minimum sized chains are dominated by the dynamic component. Therefore, the optimum supply voltage of the minimum size chain (dashed line) is the lowest V_{DD} at which yield constraints are met. By definition, this is equal to V_{DDcrit}. In the constant yield sizing scheme (solid line), reducing the supply below V_{DDcrit} necessitates an increase in device widths. The resulting rise in C_{eff} dominates total energy. In this situation, there is no benefit from upsizing in order to operate at lower V_{DD}. The optimum operating point is with minimum sizing at the lowest V_{DD} permitted by the failure rate constraint.

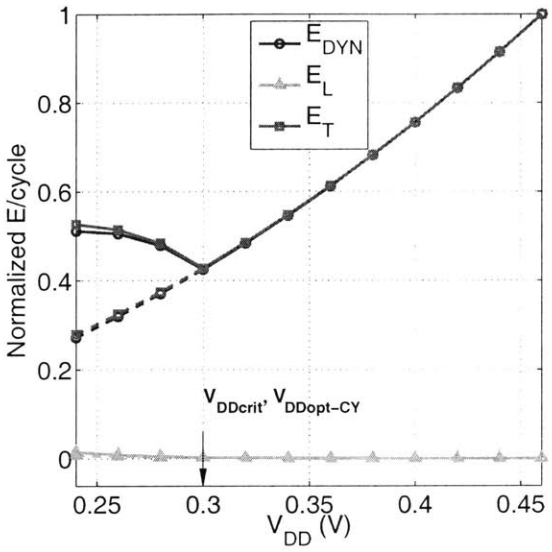


Figure 4-2: Energy versus V_{DD} of 11-stage inverter chain. Solid and dashed lines indicate constant yield and minimum sizing respectively.

When the minimum size circuit does have a local minimum in its energy characteristic, three scenarios exist depending on the relationship between V_{DDcrit} and the optimum V_{DD} of the constant yield ($V_{DDopt-CY}$) and minimum sizing ($V_{DDopt-MS}$) schemes.

Case 1) $V_{DDopt-MS} > V_{DDcrit}$: No upsizing is required to operate at the minimum energy point, therefore a minimum sized circuit at $V_{DDopt-MS}$ yields lowest energy.

Case 2) $V_{DDopt-MS} < V_{DDopt-CY} < V_{DDcrit}$: A minimum size circuit cannot operate at $V_{DDopt-MS}$ without violating failure rate constraints. A circuit suitably upsized

to operate at $V_{DDopt-CY}$ yields lowest energy while satisfying yield requirements.

Case 3) $V_{DDopt-MS} < V_{DDopt-CY} = V_{DDcrit}$: At V_{DDcrit}, the circuit under both sizing schemes are identical. Therefore a minimum size circuit operating at V_{DDcrit} provides minimum energy.

An example of case 2 is seen in Figure 4-3 for a synthesized 32-bit Kogge-Stone adder with interconnect parasitics extracted from layout. If we ignore failure rate constraints, the minimum size adder (dashed line) has an optimum supply voltage of $V_{DDopt-MS}$=280mV. When we account for failure rate constraints, the effect of constant yield sizing (solid line) is to add energy overhead when $V_{DD} < V_{DDcrit}$. This shifts the local minimum to the right, hence $V_{DDopt-CY} > V_{DDopt-MS}$. Here $V_{DDopt-CY}$ is also $< V_{DDcrit}$, therefore the adder with constant yield sizing at $V_{DDopt-CY}$=300mV consumes 10.1% less energy than a minimum size adder at V_{DDcrit}=340mV. In this example, constant yield sizing results in a small reduction in energy due to the shallow minimum of the energy versus V_{DD} curve.

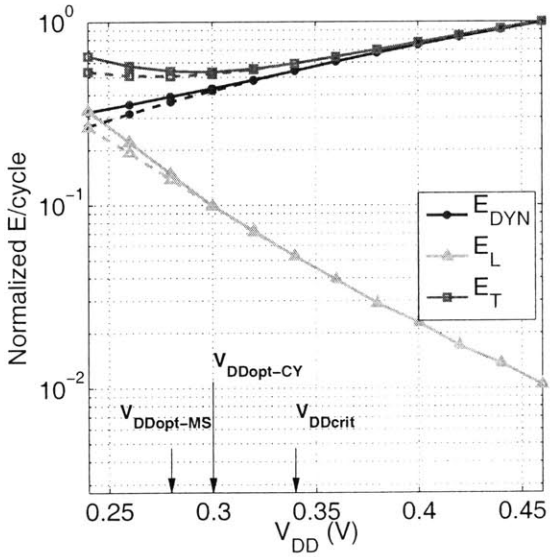
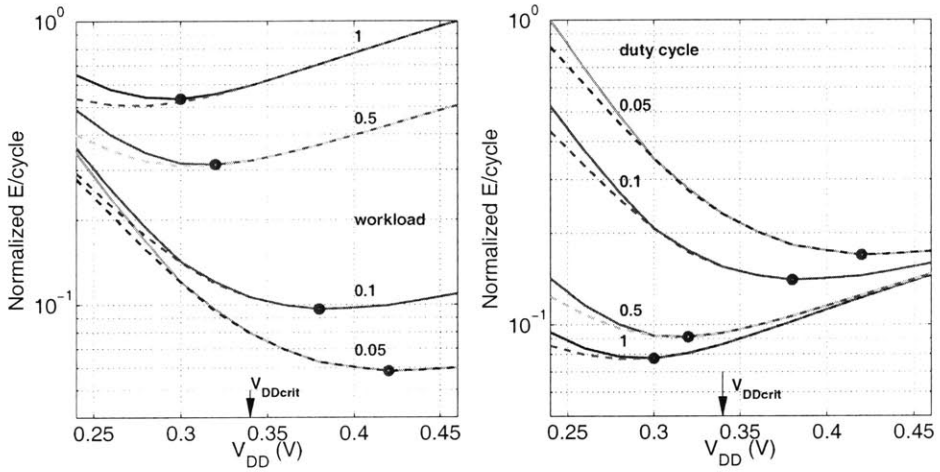


Figure 4-3: Energy versus V_{DD} of 32-bit adder. Solid and dashed lines indicate constant yield and minimum sizing respectively.

4.2 Movement of the Minimum Energy Point

Examples in Section 4.1 show that constant yield sizing allows energy reduction over minimum sizing in some cases, depending on the relationship between $V_{DDopt-MS}$ and V_{DDcrit}. As noted in [7], the local minimum changes with the relative contributions of dynamic and leakage energy, for example with workload and duty cycle of a circuit. At high workload or duty cycle, dynamic energy dominates and $V_{DDopt-MS} < V_{DDcrit}$, hence a circuit benefits from upsizing to operate at a lower V_{DD}. At low workload or duty cycle, $V_{DDopt-MS} > V_{DDcrit}$, and minimum energy operation is achievable with a minimum size circuit. The movement of the minimum energy point with these conditions is illustrated in Figure 4-4 for the 32-bit adder. If the workload and duty cycle of a circuit vary greatly at run-time, then additional analysis is necessary to determine the sizing which gives lowest energy on average across all working conditions.



(a) (b)

Figure 4-4: Total energy per cycle of 32-bit adder as (a) workload and (b) duty cycle are varied. Solid dot indicates $V_{DDopt-CY}$.

Chapter 5

Fault-Tolerant Architecture

In addition to transistor sizing, circuit variability can be mitigated on the architectural level through fault tolerance techniques. Algorithm-Based Fault Tolerance (ABFT) is a class of techniques proposed for error detection with low running time and hardware overhead. This chapter presents conceptual and implementation details of an ABFT scheme to protect an arbitrary digital filter.

5.1 Algorithm-Based Fault Tolerance

Algorithm-based fault tolerance was initially proposed as a low-cost alternative to modular redundancy. Modular redundancy is a general approach that can be used to protect any computational module, where the hardware is replicated multiple times and the system output is determined by a majority vote. In contrast, error detection in ABFT systems is tailored to the computation to be performed. A general ABFT system encodes data to be used by the algorithm into a redundant state, then performs computation on both the original inputs and encoded data. The encoded output data is checked against some criteria to determine if an error has occurred.

The authors of [40] first proposed the concept of algorithm-based fault tolerance to protect matrix operations such as multiplication, LU decomposition, and inversion. Input data in the main matrix is encoded in a redundant row or column, which is appended to the main matrix. The matrix operation is then performed, and the

resulting matrix contains the output data as well as a checksum. ABFT schemes to protect parallel butterfly datapaths in Fast Fourier Transform (FFT) networks have been studied extensively, for example in [41, 42, 43]. Two weighted checksums are computed, one for inputs and one for outputs, and are compared with each other. Design considerations for these algorithms include fault coverage and hardware overhead.

Work in [44] extends the matrix checksum approach of [40] by allowing extra freedom in defining dynamics of the redundant states and the coupling between redundant and non-redundant states. This can reduce the hardware complexity in error detection. An example is provided for a simple fault-tolerant scheme for an arbitrary discrete-time linear time-invariant (DT LTI) filter. The formulation can be further extended to general linear dynamic systems [45] and to design nonconcurrent error detection schemes [46, 47]. In the latter situation, error checking is performed periodically rather than after every time step. The detection algorithm is able to detect state transition errors that occurred in past cycles based on analysis of the current, possibly corrupted state. However, this approach requires much more sophisticated encoding of the system state compared to a simple checksum scheme.

5.2 Fault-Tolerant Digital Filters

In this section, we describe theoretical operation of the ABFT technique in [44] for digital filters. We first discuss a simple case of the approach, the matrix checksum. We then present the generalization proposed by Hadjicostis which reduces the error detection overhead.

5.2.1 ABFT for Matrix Operations

The row and column checksum provides a simple fault tolerance scheme for matrix operations. We discuss an example to illustrate this approach. The formal proof is given in [40].

Given an $n \times m$ matrix A, a row checksum of A is defined as the $1 \times m$ row

vector where the i^{th} element is the sum of elements in the i^{th} column of A. A column checksum of A is similarly defined as an $n \times 1$ column vector, where the i^{th} element is the sum of elements in the i^{th} row in A. Equation 5.1 to Equation 5.3 provide an example of the row checksum, where A is the original matrix and A_{sum} is the row checksum. A_c is the row checksum matrix that is formed by appending A_{sum} to the last row of A.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{5.1}$$

$$A_{sum} = \begin{bmatrix} 12 & 15 & 18 \end{bmatrix} \tag{5.2}$$

$$A_c = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 12 & 15 & 18 \end{bmatrix} \tag{5.3}$$

From [40], the checksum property is preserved under matrix multiplication and addition. A row checksum matrix A_c multiplied by a matrix B gives a matrix C_c which also satisfies the row checksum condition. The addition of two checksum matrices yields another checksum matrix. For example, the resultant matrix of Equation 5.7 satisfies the checksum condition.

$$A_c = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 12 & 15 & 18 \end{bmatrix} \tag{5.4}$$

$$B = \begin{bmatrix} 3 & 9 & 3 \\ 10 & -49 & 1 \\ 4 & 19 & 5 \end{bmatrix} \tag{5.5}$$

$$X_c = \begin{bmatrix} 4 & 28 & -1 \\ -65 & 3 & -4 \\ 12 & 7 & 3 \\ -49 & 38 & -2 \end{bmatrix} \tag{5.6}$$

$$A_c * B + X_c = \begin{bmatrix} 39 & -4 & 19 \\ 21 & -92 & 43 \\ 149 & -151 & 77 \\ 209 & -247 & 139 \end{bmatrix} \tag{5.7}$$

5.2.2 ABFT for Discrete-Time LTI Filters

The fault-tolerance approach for matrix operations can be extended to DT LTI filters [44]. Standard forms for DT LTI filters can be specified in the state variable description given by

$$q_s[t+1] = Aq_s[t] + bx[t], \tag{5.8}$$

where t is the discrete-time index, x is the scalar input, q_s is a vector describing the filter state, and A is a matrix derived from the filter structure that describes the state transition. For example, an 8-tap FIR filter in the direct form II transposed structure of Figure 5-1 has a state variable description given by Equation 5.10. The vector $q_s[t]$ corresponds to contents of the filter registers at time t.

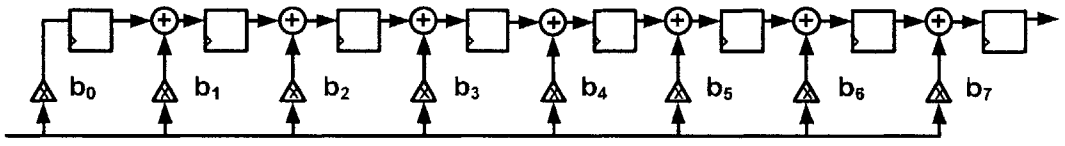


Figure 5-1: 8-tap FIR filter in direct form II transposed structure.

$$q_s[t+1] = Aq_s[t] + bx[t] \tag{5.9}$$

$$q_s[t+1] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} q_s[t] + \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} x[t] \tag{5.10}$$

As per [44], we define a standard redundant system whose state evolution is given by

$$\begin{aligned} q_\sigma[t+1] &= A_\sigma q_\sigma[t] + b_\sigma x[t] \\ &= \left[\begin{array}{c|c} A & A_{12} \\ \hline 0 & A_{22} \end{array}\right] q_\sigma[t] + \left[\begin{array}{c} b \\ \hline 0 \end{array}\right] x[t], \end{aligned} \tag{5.11}$$

where A and b are as given in Equation 5.10. A_{12} describes the coupling from the redundant to the non-redundant states, while A_{22} describes the dynamics of the redundant states.

It was shown in [44] that a redundant implementation of a system in the form of Equation 5.9 can only be obtained through a similarity transformation [48] of the standard redundant system in Equation 5.11. Therefore to add redundancy to the FIR filter of Equation 5.10, we define a system of higher dimension obtained by a similarity transformation as follows

$$q_h[t+1] = TA_\sigma T^{-1} q_h[t] + Tb_\sigma x[t], \tag{5.12}$$

where T is a matrix given by

$$T = \left[\begin{array}{c|c} I_d & \mathbf{0} \\ \hline c^T & I_s \end{array}\right]. \tag{5.13}$$

I_d is the identity matrix and c^T specifies how the system states are encoded. In this example, we use $c^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ which implements a simple row checksum. I_s is then a 1x1 identity matrix. More complex codes [46] can be used to extend error detection capability, for instance nonconcurrent error detection and identification of multiple errors in multiple time intervals. The cost is additional hardware overhead for encoding and decoding several redundant states.

Substituting Equation 5.13 into Equation 5.12, we obtain a general redundant system for a DT LTI system of Equation 5.9

$$q_h[t+1] = \left[\begin{array}{c|c} A - A_{12}c^T & A_{12} \\ \hline c^T A - c^T A_{12} c^T - A_{22} c^T & c^T A_{12} + A_{22} \end{array}\right] q_h[t] + \left[\begin{array}{c} b \\ \hline c^T b \end{array}\right] x[t]. \tag{5.14}$$

The matrix checksum scheme discussed previously in 5.2.1 is a special case of this formulation with A_{12} and A_{22} both being zero matrices. When applied to the filter of Figure 5-1, this approach results in the system described by Figure 5-2 and Equation 5.15.

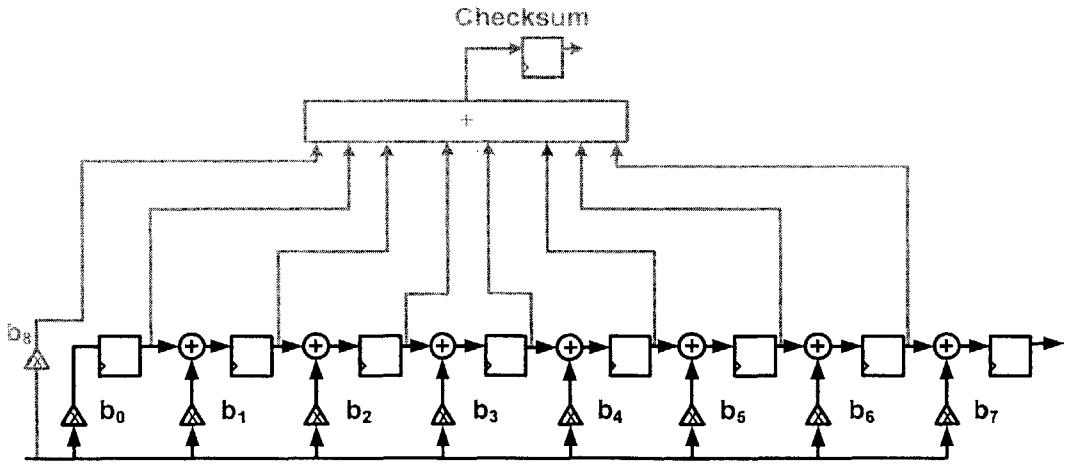


Figure 5-2: 8-tap FIR filter with row checksum redundancy (shaded in gray), $A_{12} = A_{22} = [0]$.

$$q_h[t+1] = \left[\begin{array}{cccccccc|c} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}\right] q_h[t] + \left[\begin{array}{c} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ \hline b_8 \end{array}\right] x[t] \qquad (5.15)$$

$$\text{where } b_8 = c^T b = \sum_{i=0}^{7} b_i$$

A_{12} and A_{22} provide additional freedom in designing the redundant system, which can be used to simplify the error detection hardware. For example, Figure 5-3 and Equation 5.16 result from setting $A_{12} = [0]$ and $A_{22} = [1]$. We see from comparing Figure 5-2 and Figure 5-3 that the latter requires less hardware to generate the redundant state, even though both systems have the same error detection capability. The latter design is implemented in Verilog and synthesized using the sub-threshold library, as will be discussed in Section 5.3.

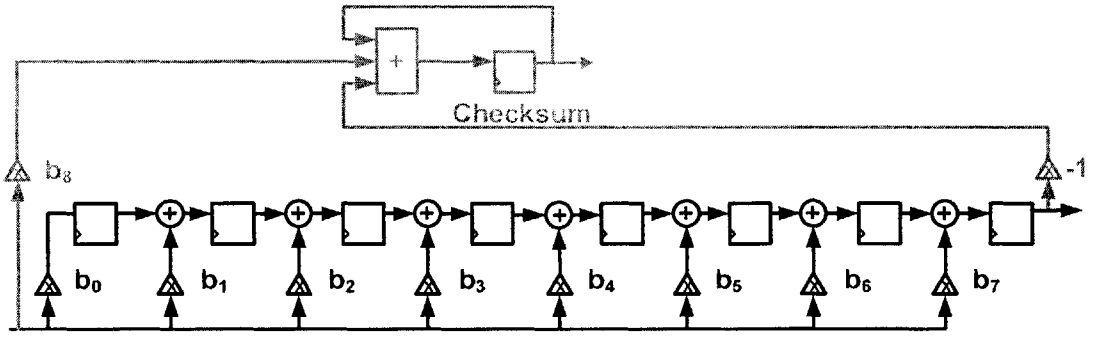


Figure 5-3: 8-tap FIR filter with row checksum redundancy (shaded in gray), $A_{12} = [0]$ and $A_{22} = [1]$.

$$q_h[t+1] = \left[\begin{array}{cccccccc|c} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array}\right] q_h[t] + \left[\begin{array}{c} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ \sum_{i=0}^{7} b_i \end{array}\right] x[t] \quad (5.16)$$

5.3 Fault-Tolerant FIR Filter Implementation

Having described the theoretical basis for a fault-tolerant DT LTI filter, we discuss details of implementing the ABFT scheme for an FIR filter in the Verilog hardware description language.

The FIR filter has a length of eight taps, operating on 16-bit wide fixed point input data. Internally, the filter maintains full precision. The filter output with sign extension is 35 bits wide. Figure 5-4 shows a block diagram of the FIR filter implemented in Verilog. The central portion drawn in black lines is the main filter in direct form II transposed structure. This structure has a shorter critical path of one adder and one multiplier delay, compared to a direct form structure which has one multiplier and seven adder delays. Furthermore the direct form II structure has seven critical paths of nominally equal length. In sub-threshold, these paths exhibit delay variation which can lead to timing violations, thus providing an opportunity to test the ABFT scheme.

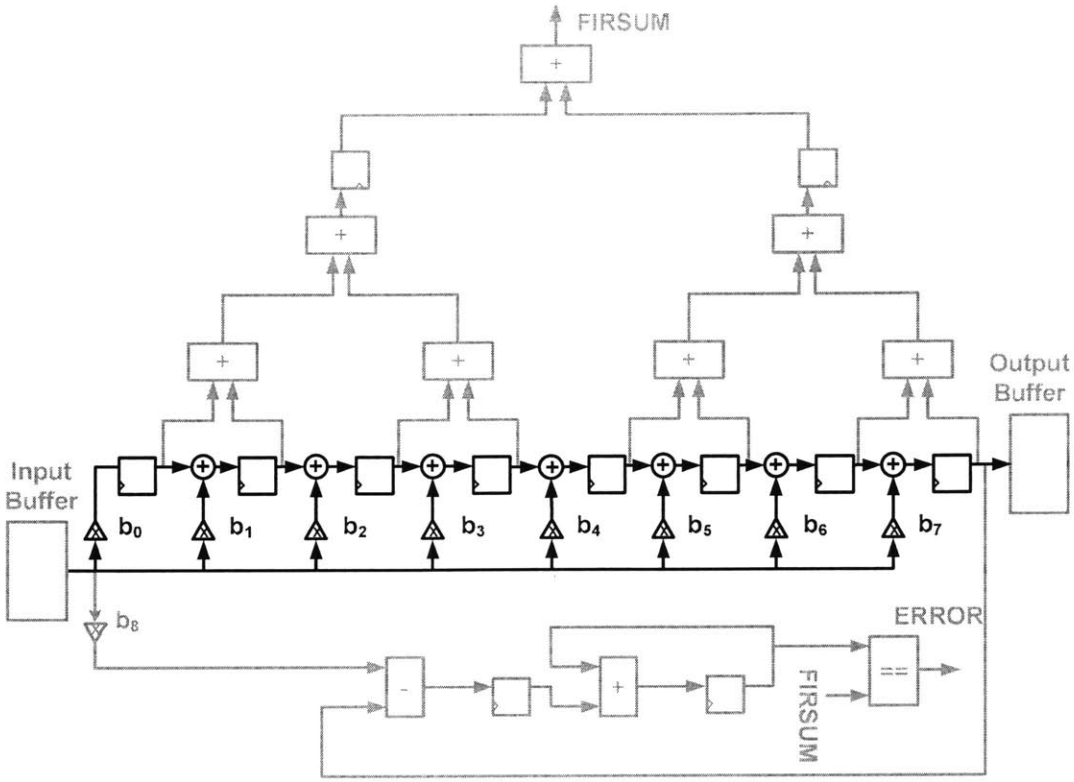


Figure 5-4: Fault-Tolerant FIR filter block diagram.

Error Detection

The redundant state is generated according to Equation 5.16. The addition of three operands is performed over two cycles to relieve timing requirements, as shown in Figure 5-4. Since the delay of redundant state generation is much shorter than the filter critical path, it is reasonable to assume that the redundant state is error-free. Moreover, errors in redundant state generation simply cause false alarms and a delay penalty in error correction, but do not cause catastrophic failure of the system.

The error detection circuitry checks the expected condition that the redundant state is equal to the sum of the main filter states. This requires independently summing the main FIR registers and comparing it with the redundant state. Again, the operation is performed over two cycles to reduce the likelihood of timing violations in error detection. This increases the delay penalty of error correction by one cycle, but does not affect filter throughput during normal operation.

Error Correction

The checksum ABFT enables error detection. Error correction is achieved by lowering the clock frequency and re-computing. As such, data storage is necessary to revert the filter back to a known correct state. This can be accomplished through two methods. First, we can buffer the input data for eight cycles so that the entire pipeline can be flushed of erroneous data, plus ED cycles, ED being the number of cycles required for error detection and adjusting the clock frequency. Second, we can replicate the nine state registers (eight taps plus one redundant state) for $ED + 1$ cycles, and restore the entire filter to one cycle before the erroneous transition. In the first option, one rollback operation takes $8 + ED$ cycles while the second takes ED cycles. However, the first option has less storage requirements which reduce the energy overhead. Since error correction is expected to occur infrequently, we select the first option to trade-off delay overhead for lower energy.

Figure 5-5 shows the finite state machine which controls system operation. During the normal operating mode, the system behaves as an FIR filter with a throughput of one data sample per cycle. If an error is detected, the system transitions into ROLLBACK state and waits for the clock source to decrease the system frequency. Upon receiving an acknowledgement from the clock source, the system rolls back the input data and begins re-computation. If another error occurs during this state, the system transitions back to ROLLBACK to initiate further decrease in clock frequency. If no error occurs, the system resumes normal operation.

The design assumes that the FIR filter does not have a throughput requirement. If there is such a constraint, then the output buffer must be made larger to store enough data samples to maintain throughput during error correction. Similarly, the time taken for error correction can be shortened if there is a latency requirement, but a finite number of overhead cycles are still necessary.

The timing diagram of the error correction procedure is given in Figure 5-6. If an error occurs where indicated, the *ERROR* signal is asserted on the second rising clock edge, because of one pipeline stage in error detection. Buffered output data is

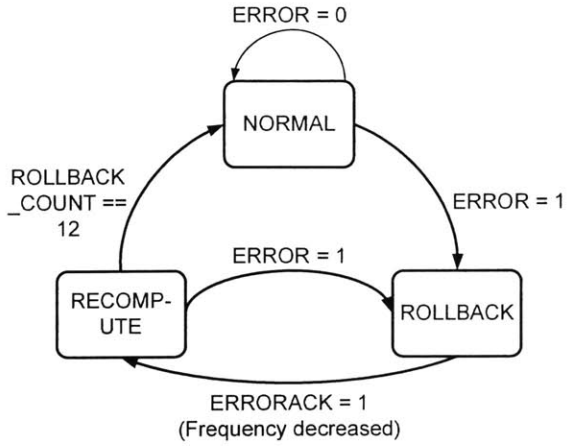


Figure 5-5: Finite state machine in fault-tolerant FIR control logic.

valid for one cycle past the assertion of *ERROR*. *DATA_VALID* is then de-asserted and the system enters the ROLLBACK state. *ERROR_ACK*=1 indicates that the system frequency has been decreased. The system then performs re-computation. If no errors occur in this phase, we obtain correct data output at the assertion of *DATA_VALID*.

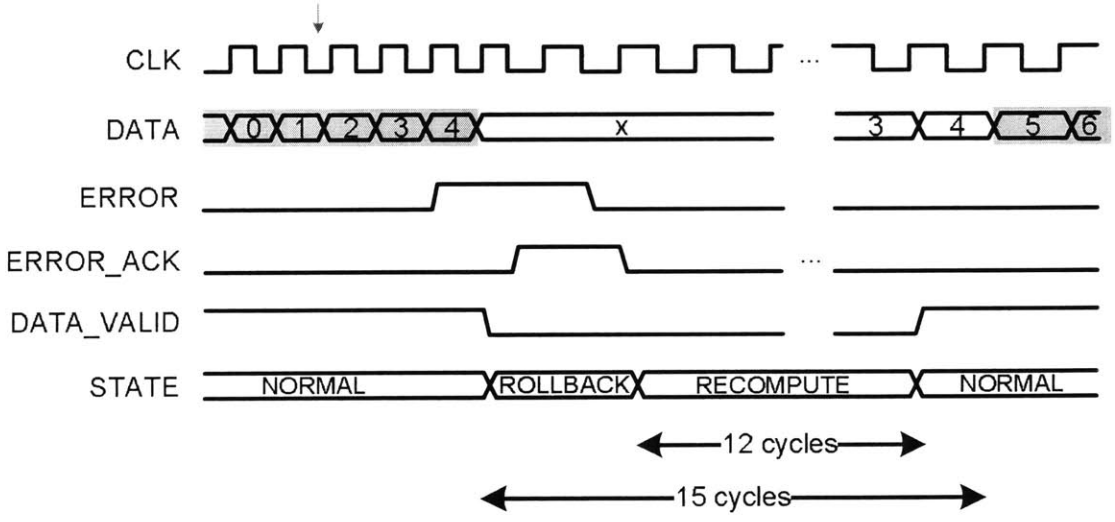


Figure 5-6: Timing diagram of error correction procedure.

Chapter 6

Sub-threshold Test Chip Results

A test chip implementing the fault-tolerant FIR filter of Chapter 5 using the sub-threshold standard cell library has been fabricated in a 65nm CMOS technology. This chapter discusses the test chip structure and then presents simulation results. At time of writing, the fabricated test chip is not yet available for measurement.

6.1 Test Chip Structure

The annotated test chip layout is provided in Figure 6-1. The test chip contains two 16-bit, 8-tap FIR filters, one with error correction mechanism and the other without, for comparison purposes. The subsequent discussion refers to the two filters as FIR-EC and FIR respectively. To facilitate testing, input data to both filters can be generated on-chip by either a linear feedback shift register (LFSR) or by a custom module that provides other input vectors of interest. The test chip can either be clocked with an external clock or by an on-chip ring oscillator with variable delay taps. When the ring oscillator is used, its control logic decreases the clock frequency automatically if an error occurs by adjusting the number of taps. For testability, an input signal to the chip can induce a bit error in the datapath, which triggers the error correction mechanism.

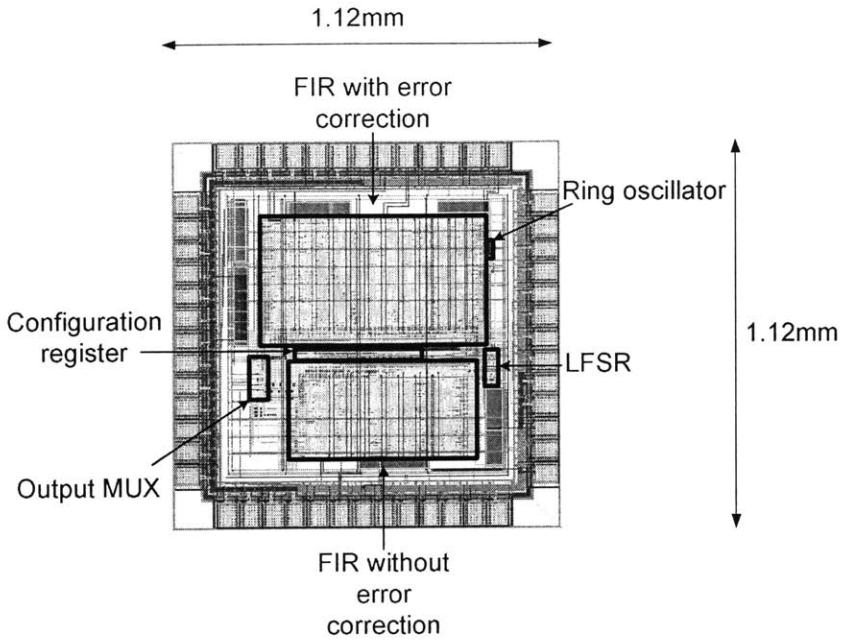


Figure 6-1: Annotated layout of sub-threshold test chip.

6.2 Simulation Results

The core logic was verified functionally using Synopsys Nanosim, a transistor-level simulation tool which allows long simulations on a large circuit to be completed in a reasonable time frame. The entire test chip, including core logic and pads, was simulated using SPICE for a short input data vector to verify connectivity.

Energy and delay performance are characterized using Nanosim. Comparing Nanosim and SPICE simulations for a small circuit shows that Nanosim gives delay results to within 10% of SPICE measurements. However, it is less accurate for sub-threshold leakage current, and can only provide an order-of-magnitude estimate (around 20% to 200%) of the SPICE measurement.

6.2.1 Delay Comparison

Figure 6-2 plots the simulated critical path delay for both FIR filters. The critical path is obtained from the post-layout netlist using timing analysis tool Primetime, then simulated in SPICE with extracted parasitics. The error correction circuitry

does not add to the critical path delay since it operates independently of the main FIR filter.

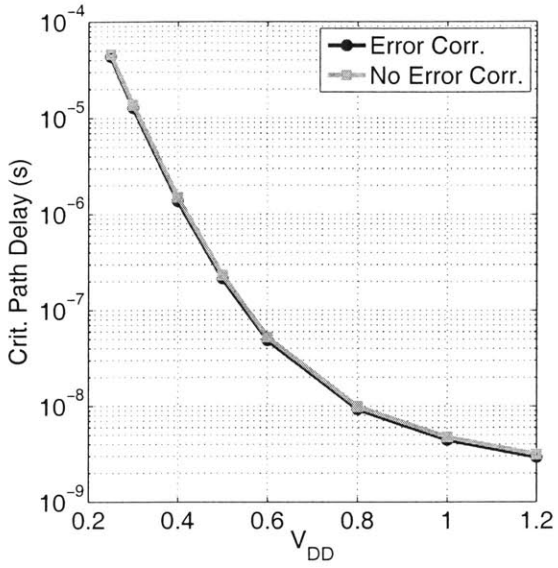


Figure 6-2: Simulated critical path delay for FIR filter, with and without error correction.

6.2.2 Energy Comparison

The total energy per cycle is plotted in Figure 6-3. Total energy is calculated as the sum of dynamic and leakage energy, as in Equation 4.1. Dynamic energy dominates in simulation such that the minimum energy supply voltage lies beyond 0.25V, the lowest operating point supported by the sub-threshold library. This observation may result from significant interconnect parasitics and must be verified from actual chip measurements. The energy overhead of the error correction circuitry stays constant over V_{DD} at approximately 80%. This overhead can be partially attributed to the input and output buffers necessary for the error correction scheme, which re-computes previous data starting from a known correct state. The buffers are currently implemented with D-type registers but can be more efficiently realized using a register file.

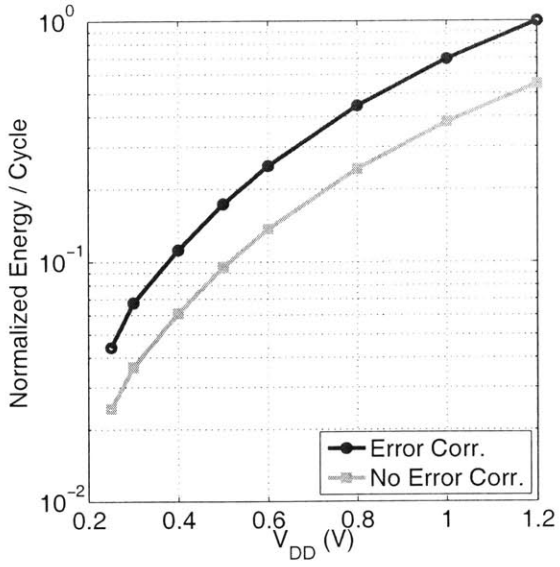


Figure 6-3: Simulated total energy per cycle for FIR-EC and FIR.

6.2.3 Error Correction Overhead Analysis

Using simulated data, we study the practicality of the FIR error correction scheme. In this discussion, we let V_{DDfail} be the supply voltage at which transient errors start to surface in the FIR filter. This scenario occurs when the ring oscillator no longer tracks the critical path delay, because of increased delay variability at lower voltages. Error correction allows the FIR filter to operate at a voltage $< V_{DDfail}$ by correcting transient timing errors that arise with voltage reduction. This results in energy savings if the fault-tolerant FIR consumes less energy at its operating point than a normal FIR working at V_{DDfail}.

To fairly compare energy consumption, we must also account for the timing overhead of error correction. The fault tolerance scheme does not increase the critical path delay during normal operation, but it does require 15 cycles to rectify one erroneous computation, assuming that decreasing the clock frequency effectively corrects the timing error. The overhead can be amortized over normal computation cycles. In terms of total energy per *valid* operation, this effectively increases the switching energy and the cycle time over which leakage current is integrated. Without detailed

analysis of input statistics, V_T mismatch, and data-dependent delays, it is difficult to simulate how frequently transient errors occur and the equivalent overhead. For purposes of illustration, we assume that error correction effectively adds 5, 10, and 20% to both C_{eff} and cycle time. This amounts to approximately one error occurring every 300, 150, and 75 cycles respectively.

From this discussion, we can compute the theoretical break-even point using the energy data from Figure 6-3. This is shown in Figure 6-4, where the fault-tolerant FIR (with overhead) operating at V_{DDcrit} would consume equal energy as a normal FIR at V_{DDfail}. For example, if the normal FIR fails at 0.35V, a fault-tolerant FIR with 10% timing overhead (T_{OH}) would need to operate at lower than 0.24V in order for error correction to provide an energy advantage.

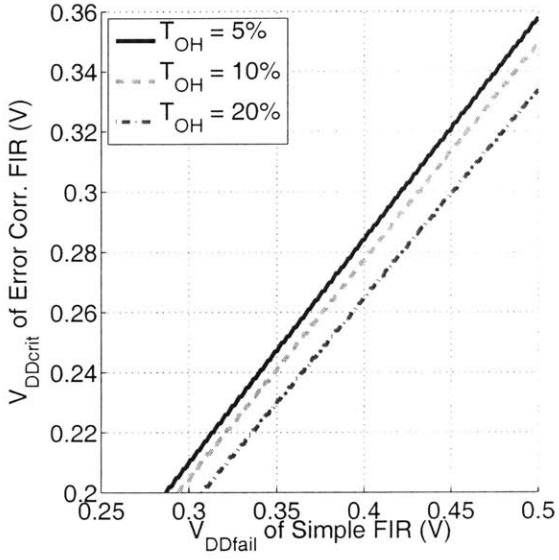


Figure 6-4: Overhead analysis of error correction scheme. If FIR fails at V_{DDfail} (x-axis), FIR-EC would need to operate at the corresponding V_{DDcrit} (y-axis), or lower, in order to provide energy savings.

Chapter 7

Conclusions

This thesis presents a standard cell library and design methodology for sub-threshold circuits. The library will serve as a platform for further exploration of sub-threshold operation, by allowing synthesis of arbitrary synchronous digital logic which is robust against process variation. This chapter summarizes the key lessons resulting from this research and discusses opportunities for future work.

7.1 Device Sizing and Library Implementation

In the device sizing analysis of Chapter 2, we have shown that while minimum size devices result in lowest energy and delay in sub-threshold, their increased sensitivity to random dopant fluctuation may cause functional errors. Functionality is no longer guaranteed even in static CMOS logic which is commonly assumed to be robust, although certain structures are less susceptible to local variation. As random V_T variation increases with process scaling, we should consider probabilistic models of circuit functionality and performance, contrary to the common practice of treating circuit behavior as deterministic.

Variability in sub-threshold motivates a new logic gate design methodology with yield, then low energy, as the primary goals. The functional yield of one gate does not relate in a straightforward way to the yield of a fabricated circuit. Further analysis is necessary to account for the many factors affecting die yield, for example circuit

logic depth, interconnection between gates, and global process conditions.

Library design considerations such as logic function and drive strength selection again underscore the difficulty of optimizing cells for an unknown application. In incorporating the sub-threshold library of this thesis in a commercial CAD flow, we observe that current tools do not adapt well to designing for within-die variation or operation at multiple supply voltages. Design tool functionality will need to be extended as technology continues to scale and dynamic voltage scaling becomes more widely used.

Timing verification remains a major challenge in a sub-threshold CAD methodology. Existing design tools assume worst case delays for all gates in a timing path, which is clearly impractical in sub-threshold given large delay variability. New techniques are necessary which treat the delay of each gate as a random variable, derive an overall distribution for each logic path, and verify that timing requirements are met to a given level of confidence.

7.2 Minimum Energy Operation With Yield Constraint

In Chapter 4, we examine whether reducing V_{DD} leads to energy savings, if it also requires device upsizing to satisfy a yield constraint. As illustrated by two case studies, upsizing provides energy benefits when the energy versus V_{DD} plot reaches a minimum within the range of V_{DD} where upsizing is required. The current analysis requires simulating a given design at various transistor sizing and V_{DD} to determine the optimum selections empirically. Analytical modeling to predict the best supply and sizing would be highly desirable. With an efficient model, the analysis can be more easily extended to find the optimum sizing when the power supply varies in a dynamic voltage scaling scenario.

7.3 Fault-Tolerant Architecture

Circuit simulations show that algorithm-based fault tolerance techniques can correct timing errors in sub-threshold. Although the test circuit implemented in this thesis represents one of the simplest ABFT schemes described in literature, the energy overhead of error correction is still significant. The proposed scheme would provide energy savings if error correction allows the circuit to operate at a considerably lower voltage than one without correction. Theoretical evaluation of ABFT overhead generally does not account for the ability to reduce timing errors by raising the supply voltage. When ABFT is applied to sub-threshold circuits, it is important to model the change in timing variability with V_{DD} in order to estimate the overhead more accurately. As such, many opportunities remain in the design of generally applicable, fault-tolerant architectures with low overhead for sub-threshold operation.

Appendix A

List of Standard Cells

Table A.1: List of standard cells in sub-threshold library.

Cell Name	Description
ADDFX1	Full adder
ADDHX1	Half adder
AFHCINX1	Full adder with inverted carry-in
AFHCONX1	Full adder with inverted carry-out
AND2X1 AND2X4	And
AOIB21X1 AOIB21X4	And-or-invert
BUFX1 BUFX2 BUFX4 BUFX8 BUFX12	Buffer
CBUF1	Top level clock buffer
CBUF21	Top level clock buffer
Continued on next page	

Table A.1 – continued from previous page

Cell Name	Description
CBUF22	Top level clock buffer
DFFRX1	D-register with asynchronous reset
DFFTGX1	Positive-edge D-register
DFFTGNX1	Negative-edge D-register
DFFTRX1	D-register with synchronous reset
DFFTSX1	D-register with synchronous preset
DLYX1 DLY2X2	Delay
INVX1 INVX2 INVX4 INVX8 INVX16	Inverter
MUX2X1 MUX2X4	2-1 Multiplexer
MUXI2X1 MUXI2X4	Mux-inverter
NAND2BX1 NAND2BX2	Invert-nand
NAND2X1 NAND2X2 NAND2X4 NAND3X1	Nand
NOR2BX1 NOR2BX2	Invert-nor
NOR2X1	Nor

Continued on next page

Table A.1 – continued from previous page

Cell Name	Description
NOR2X2 NOR2X4 NOR3X1	Nor
OAIB21X1 OAIB21X4	Or-and-invert
OR2X1 OR2X4	Or
TLATNX1 TLATNX4	Active low latch
TLATX1 TLATX4	Active high latch
XNOR2X1 XNOR2X4	Exclusive-nor
XOR2X1 XOR2X4	Exclusive-or

Bibliography

[1] E. Seevinck, F. List, and J. Lohstroh, "Static Noise Margin Analysis of MOS SRAM Cells," *IEEE Journal of Solid-State Circuits*, vol. SC-22, no. 5, pp. 748–754, Oct. 1987.

[2] G. Gerosa, S. Gary, C. Dietz, D. Pham, K. Hoover, J. Alvarz, H. Sanchez, P. Ippolito, T. Ngo, S. Litch, J. Eno, J. Golab, N. Vanderschaaf, and J. Kahle, "A 2.2W, 80MHz Superscalar RISC Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1440–1454, Dec. 1994.

[3] H. L. Yeager, M. J. Patyra, R. Reyes, and K. A. Bowman, "Microprocessor Power Optimization through Multi-Performance Device Insertion," in *Symposium on VLSI Circuits (VLSI) Digest of Technical Papers*, June 2004, pp. 334–337.

[4] R. M. Swanson and J. D. Meindl, "Ion-Implanted Complementary MOS Transistors in Low-Voltage Circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-7, no. 2, pp. 146–153, Apr. 1972.

[5] J. Burr and A. Peterson, "Ultra Low Power CMOS Technology," in *3rd NASA Symposium on VLSI Design*, 1991, pp. 4.2.1–4.2.13.

[6] A. Wang, A. Chandrakasan, and S. Kosonocky, "Optimal Supply and Threshold Scaling for Sub-threshold CMOS Circuits," in *IEEE Computer Society Annual Symposium on VLSI*, Apr. 2002, pp. 7–11.

[7] B. H. Calhoun and A. Chandrakasan, "Characterizing and Modeling Minimum Energy Operation for Subthreshold Circuits," in *International Symposium on*

Low-Power Electronics and Design (ISLPED) Digest of Technical Papers, 2004, pp. 90–95.

[8] H. Soeleman and K. Roy, "Ultra-Low Power Digital Subthreshold Logic Circuits," in *International Symposium on Low-Power Electronics and Design (ISLPED) Digest of Technical Papers*, 1999, pp. 94–96.

[9] H. Soeleman, K. Roy, and B. Paul, "Sub-Domino Logic: Ultra-Low Power Dynamic Sub-Threshold Digital Logic," in *International Conference on VLSI Design (VLSI-Design) Digest of Technical Papers*, Jan. 2001, pp. 211–214.

[10] H. Kim and K. Roy, "Ultra-Low Power DLMS Adaptive Filter for Hearing Aid Applications," in *International Symposium on Low-Power Electronics and Design (ISLPED) Digest of Technical Papers*, Aug. 2001, pp. 352–357.

[11] C. H. Kim, H. Soeleman, and K. Roy, "Ultra-Low-Power DLMS Adaptive Filter for Hearing Aid Applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 4, pp. 716–730, Aug. 2003.

[12] A. Wang and A. Chandrakasan, "A 180mV FFT Processor Using Sub-threshold Circuit Techniques," in *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 2004, pp. 292–293.

[13] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Device Sizing for Minimum Energy Operation in Subthreshold Circuits," in *Custom Integrated Circuits Conference (CICC) Digest of Technical Papers*, Oct. 2004, pp. 95–98.

[14] B. Calhoun and A. Chandrakasan, "Analyzing Static Noise Margin for Sub-threshold SRAM in 65nm CMOS," in *IEEE European Solid-State Circuits Conference (ESSCIRC) Digest of Technical Papers*, Sept. 2005, pp. 363–366.

[15] ——, "A 256-kbit Sub-threshold SRAM 65nm CMOS," in *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, Feb. 2006, pp. 628–629.

[16] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and Mitigation of Variability in Subthreshold Design," in *International Symposium on Low-Power Electronics and Design (ISLPED) Digest of Technical Papers*, 2005, pp. 20–25.

[17] J. Chen, L. T. Clark, and Y. Cao, "Robust Design of High Fan-In/Out Subthreshold Circuits," in *IEEE International Conference on Computer Design (ICCD) Digest of Technical Papers*, Oct. 2005, pp. 405–410.

[18] Y. Cao and L. T. Clark, "Mapping Statistical Process Variations Toward Circuit Performance Variability: An Analytical Modeling Approach," in *ACM/IEEE Design Automation Conference (DAC) Digest of Technical Papers*, June 2005, pp. 658–663.

[19] C. Enz, F. Krummenacher, and E. Vittoz, "An Analytical MOS Transistor Model Valid in All Regions of Operation and Dedicated to Low-Voltage and Low-Current Applications," *Journal on Analog Integrated Circuits and Signal Processing*, pp. 83–114, July 1995.

[20] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[21] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and Practical Limits of Dynamic Voltage Scaling," in *ACM/IEEE Design Automation Conference (DAC) Digest of Technical Papers*, 2004, pp. 868–873.

[22] D. Boning and S. Nassif, *Models of Process Variations in Device and Interconnect*. IEEE Press, 2001, pp. 98–115.

[23] J. A. Power, B. Donnellan, A. Mathewson, and W. A. Lane, "Relating Statistical MOSFET Model Parameter Variabilities to IC Manufacturing Process Fluctuations Enabling Realistic Worst Case Design," *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 306–318, Aug. 1994.

[24] K. Bowman, S. Duvall, and J. Meindl, "Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 183–190, Feb. 2002.

[25] A. Wang and A. Chandrakasan, "A 180-mV Subthreshold FFT Processor Using a Minimum Energy Design Methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, 2005.

[26] [Online]. Available: http://www-device.eecs.berkeley.edu/bsim3/ bsim4.html

[27] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching Properties of MOS Transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.

[28] G. Schrom and S. Selberherr, "Ultra-Low-Power CMOS Technologies," in *International Semiconductor Conference (CAS) Digest of Technical Papers*, 1996, pp. 237–246.

[29] J. Lohstroh, E. Seevinck, and J. D. Groot, "Worst-case Static Noise Margin Criteria for Logic Circuits and Their Mathematical Equivalence," *IEEE Journal of Solid-State Circuits*, vol. SC-18, no. 6, pp. 803–807, June 1983.

[30] C. Piguet, J.-M. Masgonty, S. Cserveny, C. Arm, and P.-D. Pfister, "Low-Power Low-Voltage Library Cells and Memories," in *International Conference on Electronics, Circuits and Systems (ICECS) Digest of Technical Papers*, 2001, pp. 1521–1524.

[31] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Prentice Hall, 2003.

[32] Y. Suzuki, K. Odagawa, and T. Abe, "Clocked CMOS Calculator Circuitry," *IEEE Journal of Solid-State Circuits*, vol. SC-8, no. 6, pp. 462–469, Dec. 1973.

[33] D. Markovic, B. Nikolic, and R. W. Brodersen, "Analysis and Design of Low-Energy Flip-Flops," in *International Symposium on Low-Power Electronics and Design (ISLPED) Digest of Technical Papers*, Aug. 2001, pp. 52–55.

[34] D. Markovic, "Analysis and Design of Low Energy Clocked Storage Elements," Master's thesis, University of California at Berkeley, 2000.

[35] I. Sutherland, B. Sproul, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.

[36] B. Calhoun and A. Chandrakasan, "Ultra-Dynamic Voltage Scaling (UDVS) Using Sub-threshold Operation and Local Voltage Dithering in 90nm CMOS," in *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, Feb. 2005, pp. 300–301.

[37] P. Zarkesh-Ha, T. Mule, and J. D. Meindl, "Characterization and Modeling of Clock Skew with Process Variations," in *Custom Integrated Circuits Conference (CICC) Digest of Technical Papers*, 1999, pp. 441–444.

[38] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani, "Analysis of the Impact of Process Variations on Clock Skew," *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, no. 4, pp. 401–407, Nov. 2000.

[39] A. Agarwal, V. Zolotov, and D. T. Blaauw, "Statistical Clock Skew Analysis Considering Intradie-Process Variations," *IEEE Transactions on Circuits and Systems*, vol. 23, no. 8, pp. 1231–1242, Aug. 2004.

[40] K.-H. Huang and J. A. Abraham, "Algorithm-Based Fault Tolerance for Matrix Operations," *IEEE Transactions on Computers*, vol. c-33, no. 6, pp. 518–528, June 1984.

[41] J.-Y. Jou and J. A. Abraham, "Fault-Tolerant FFT Networks," *IEEE Transactions on Computers*, vol. 37, no. 5, pp. 548–561, May 1988.

[42] C. G. Oh, H. Y. Youn, and V. K. Raj, "An Efficient Algorithm-Based Concurrent Error Detection for FFT Networks," *IEEE Transactions on Computers*, vol. 44, no. 9, pp. 1157–1162, Sept. 1995.

[43] S.-J. Wang and N. K. Jha, "Algorithm-Based Fault Tolerance for FFT Networks," *IEEE Transactions on Computers*, vol. 43, no. 7, pp. 849–854, July 1994.

[44] C. N. Hadjicostis, "Fault-Tolerant Discrete-Time Linear Time-Invariant Filters," in *IEEE International Conference on Acoustic, Speed, and Signal Processing (ICASSP) Digest of Technical Papers*, June 2000, pp. 3311–3314.

[45] ——, "Coding Approaches to Fault Tolerance in Linear Dynamic Systems," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 210–228, Jan. 2005.

[46] ——, "Nonconcurrent Error Detection and Correction in Fault-Tolerant Discrete-Time LTI Dynamic Systems," *IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications*, vol. 50, no. 1, pp. 45–55, Jan. 2003.

[47] ——, "Finite-State Machine Embeddings for Nonconcurrent Error Detection and Identification," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 142–153, Feb. 2005.

[48] T. Kailath, *Linear Systems*. Prentice-Hall, 1980.