The Pennsylvania State University

The Graduate School

Department of Electrical Engineering

CLOCK NETWORK AND PHASE-LOCKED LOOP POWER ESTIMATION AND EXPERIMENTATION

A Thesis in

Electrical Engineering

by

David Enrique Duarte

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2002

We approve the thesis of David Enrique Duarte.

Date of Signature

Mary Jane Irwin
Distinguished Professor of Computer Science and Engineering
Thesis Adviser
Co-chair of Committee

Kenneth Jenkins
Professor of Electrical Engineering
Head of the Department of Electrical Engineering
Co-chair of Committee

William E. Higgins
Professor of Electrical Engineering

Thomas N. Jackson
Professor of Electrical Engineering
Robert E. Kirby Chair Professor

Vijaykrishnan Narayanan
Assistant Professor of Computer Science and Engineering

ABSTRACT

The clock distribution network and the generation circuitry are critical components of current synchronous digital systems and are known to consume more than a quarter of the power budget of existing microprocessors. A high-level clock energy model that captures both the dynamic and leakage power components is formulated. The validation results show an average deviation within 5% of circuit-level simulations.

Furthermore, Phased Locked Loops (PLLs), which have been generally used in clock generation, are also crucial for the implementation of Dynamic Voltage Scaling (DVS) mechanisms employed in emerging power conscious processor designs. In order to devise architectural and compiler driven optimizations that exploit the dynamic frequency/voltage scaling features, accurate models that capture the performance and power characteristics of the PLL are essential. In addition, many emerging System-on-a-Chip (SOC) designs use multiple PLLs on the same die making it important to estimate the contribution of the PLL to the overall system power. A PLL energy and timing model that accurately estimates the power consumption during both lock and lock-acquisition states is also formulated. The applicability of PLLs as voltage regulators in support of leakage reduction by supply gating is briefly discussed.

The complete clock energy model is incorporated into a cycle-accurate energy simulator for an embedded architecture. This framework is used to study and quantify the influence on clock energy of several architectural-level decisions and their relative impact on the overall system power. These design choices include various cache architectures and clock gating at different levels (top-level distribution network, functional unit and gate level). From the software perspective, the influence on clock energy of power-aware memory-oriented compiler optimizations is assessed.

Finally, the model is used to predict the role that the clock will have in the total power budget of future designs while carefully capturing the impact of technology scaling. It is shown that as long as leakage power is kept under control, clock power will remain a significant contributor to the total system power.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am grateful and indebted to my thesis advisor Professor Mary Jane Irwin for the guidance and encouragement she has shown me and also for the financial support she provided me during my time at Penn State. I would like to thank Professor Vijaykrishnan Narayanan for his thoughtful suggestions and motivating discussions on various topics, as well as for his feedback during my research. I am also very grateful to Professor Kenneth Jenkins for accepting the task of being Co-Chair of my committee and to Professor Thomas Jackson and Professor William Higgins for her valuable comments on my work.

I would like to express my gratitude to Dr. Shizhuo Yin for his guidance during the early years of my graduate studies and to the EE Graduate Records Secretary, Diana Feltenberger, for all her help during my stay at Penn State. I would like to thank my lab mates Jeyran Hezavei and Hyun-Suk Kim for many enlightening and entertaining discussions, and my long time roommate Eugenio Zapata for his friendship and encouragement. I wish them the best in all their endeavors.

I am very thankful to my brothers and sister and their families for their continued support and understanding and also to Rebecca for all her love and help proofreading this manuscript. Finally, I am most grateful to my parents for their continued love and encouragement and for all they have done for me throughout my entire life.

Chapter 1

INTRODUCTION

It has become clear for architects and designers that performance, cost, testability, reliability and time-to-market must be accompanied by low power consumption as the main constraints that govern the design of integrated circuits. The way these constraints are balanced depends on the ultimate purpose that the device is aimed for, but low power remains important for the entire spectrum from high performance to embedded and mobile devices.

1.1 The importance of designing for low power consumption

Low energy dissipation is not only of interest for portable devices, where battery life is crucial for the success of a product, but also for non-mobile systems and supercomputers, where chip-level issues like power delivery and packaging, and system-level issues like integration, cooling and case design are important.

The implementation of the power delivery grid is becoming a very complex task due to the effects of technology scaling and the increasing complexity (i.e., increasing number of transistors per unit of area) of recent designs. In particular, the parasitic inductance and capacitance associated with the distribution wiring grids of the supply rails will form a resonant LC circuit, which, due to the pulsating nature of the current being drawn by the logic circuitry, would eventually oscillate. The effect is worsened by the continuing increase of the supply current needed by latest high-performance systems and also by the small series resistance required by the delivery grid to avoid an excessive voltage drop across it. This clearly becomes detrimental to reliability, noise immunity and performance. The issue can be expressed as follows: there is no point in having the most advanced microprocessor design in the world if there is no way to reliably deliver power to the entire chip. So, from this perspective, if would really help the power delivery design team if low power design rules are followed during the design stage.

As the amount of energy dissipated as heat is proportional to the current requirements of the circuit, the larger the power consumed, the more efficient the packaging must be to remove the generated heat. Normally, the latter implies a more expensive package. This is key in a competitive business where vendors buy in high volumes and a $10 price difference per unit (due to a costlier package) would add to several millions difference in a single shipment.

The above aspects come in conjunction with system level issues like integration, the cooling system and case design. The more power hungry a unit is, the more demanding its integration considerations are, the more expensive its cooling mechanisms need to be and the more sophisticated the case design is. As an example of how important these aspects are, consider the case of temperature-control mechanisms. Initially, contact with air was enough to take the heat away from the die. In recent years, the use of large heat metal dissipators was needed in order to provide a larger radiating area. When this was not enough, forced air was used by means of small fans attached to the die to cool down the device. The obvious question is, what is next? Will it get to the point where home computers feature water-based cooling mechanisms such as the ones used in Cray supercomputers? This is an alternative that the semiconductor industry is not likely to be attracted to. Additionally, the growing interest for environment-friendly devices is pushing even harder for the development of low power designs, as the discussion for energy conservation has been in the spotlight for several years now. Besides the above concerns that apply to all designs, portable and mobile systems bring into the picture the issue of maximizing battery life. It is clear that the lower the power consumption of a device is, the longer the battery will last.

So, it can be seen that reducing or keeping power consumption to a reasonable levels is vital as if this does not happen, all other design objectives (performance, cost and robustness) become harder to attain. *Figure 1–1* clearly shows that performance has been the predominant objective pursued by the microprocessor industry. Clock frequency is not a direct measure of performance but is a good indicator. This, however, has a very negative impact on power consumption, as *Figure 1–2* shows. Even though technology scaling implies supply voltage and switching capacitance reduction from one generation to the next, the increase in frequency and design complexity (die size and number of transistors) translates in even larger power consumption. The valleys in the power graph represent the transition of the same architecture from one technology generation to the following.

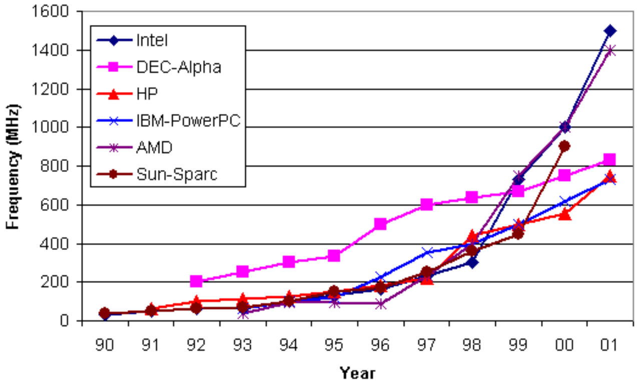


Figure 1–1: Evolution of the operating frequency for various microprocessor families. Data taken from [1].

Figure 1–3 shows the power density of various high performance microprocessors already fabricated. It can be observed that all current designs have surpassed the power density of that of a kitchen hot plate’s heating coil (which is rated at 0.1W/mm^2) and their power densities have been following an almost exponential trend. The excessive heat associated with such large power densities could cause silicon failure and hurt system reliability. Thus, the need for techniques that provide ways for power reduction is now clear and justified.

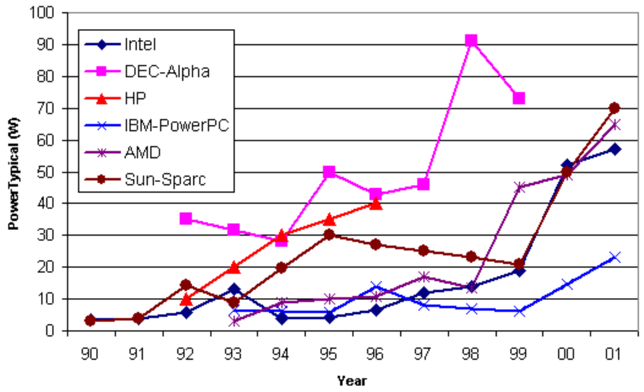


Figure 1–2: Evolution of the typical power consumption for various microprocessor families. Data taken from [1].

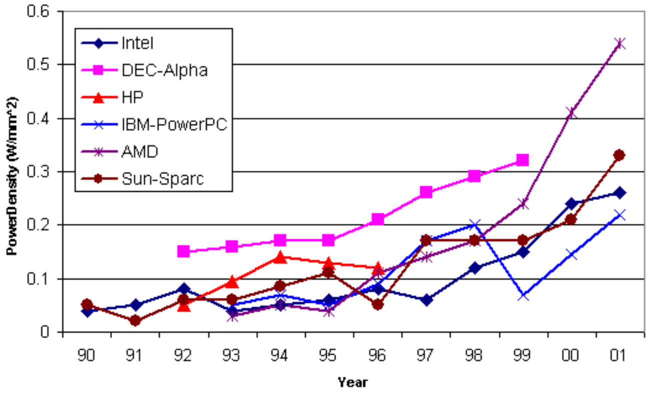


Figure 1–3: Power density of commercial microprocessor families. Data taken from [1].

1.2 The sources of power consumption in VLSI silicon circuits

Now that the importance of designing for low power has been presented, the causes of power consumption in silicon circuits can be introduced. As shown in *Figure*

1–4, there are four main sources of power dissipation, that when added, define the total power consumption of a digital gate as expressed in Eq. 1.1 [2].

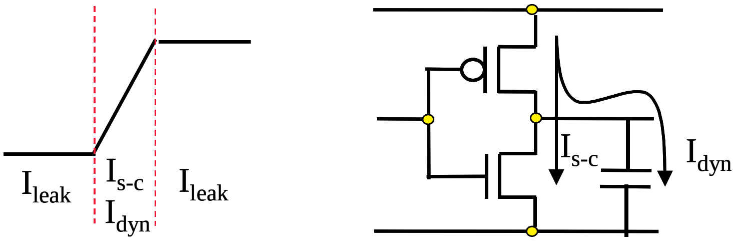


Figure 1–4: The sources of power consumption.

$$\begin{aligned} P_{avg} &= P_{dynamic} + P_{short-circuit} + P_{leakage} + P_{static} \\ P_{avg} &= [\alpha_{0\rightarrow 1} \cdot C_L \cdot V \cdot V_{dd} \cdot f_{clk}] + [I_{sc} \cdot V_{dd}] + [I_{leakage} \cdot V_{dd}] + [I_{static} \cdot V_{dd}] \end{aligned} \tag{1.1}$$

Traditionally, as CMOS technology started to be widely accepted (until recently when the minimum feature size reached the deep sub-micron regime), power consumption evaluation was limited to the estimation of the dynamic power ($P_{dynamic}$). For technologies like pure NMOS, where there is a constant current between the power supply and ground, the static component (P_{static}) of the total power is not negligible, but for the CMOS design style, P_{static} is basically zero. The total power consumption in CMOS technology has indeed been dominated by the dynamic component (normally 90% to 95% of the total power) which is basically the energy required to charge the transistor and interconnect parasitic capacitances as the signal value switches from low to high. In Eq. 1.1, α_{0-1} is the activity factor that represents the average fraction of nodes that make a power consuming transition in one period, C_L is the total capacitive load and V is the voltage swing.

Short-circuit power ($P_{short-circuit}$), which occurs when both the PMOS and the NMOS devices are ON during switching, has been responsible for the remaining 5% to 9% of the total power. This is due to the fact that the short-circuit time is usually very small (and becomes even smaller as the new technologies push the operating frequency into the GHz realm) and the rise and fall times are typically comparable.

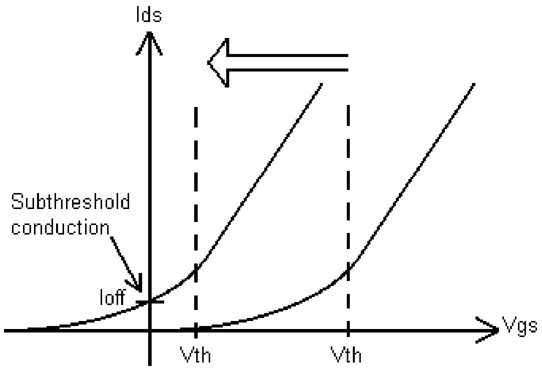


Figure 1–5: The increase in subthreshold leakage current as V_{th} decreases.

The leakage component ($P_{leakage}$) has been normally less than 1% of the total system power and was obviously disregarded. The two most significant sources of leakage power are diode junction leakage and sub-threshold conduction. The diode junction (or *pn* reverse bias) leakage occurs when a transistor is turned off and another device charges or discharges the drain with respect to the former bulk potential. It is independent of the reverse voltage and directly dependent on the reverse saturation current and it has been shown to contribute very slightly to the total leakage power [3]. The effects of sub-threshold currents, however, have been amplified by the continuous reduction of the supply voltage (required to reduce the effective dynamic power consumption and the power density in sub-micron technologies), which in turn calls for reduction of the transistor threshold voltage to maintain performance [2], [4]. As *Figure 1–5* shows, the drain source current behavior around the threshold voltage (V_{th}) is clearly not ideal and as the absolute value of V_{th} reduces, conduction at V_{gs} = 0 becomes more significant. This will accentuate the absolute influence of the leakage component, which will require designing not only for dynamic, but also leakage power reduction as well.

The equations given before can be re-written so that only the components of interest, for the target technology (CMOS), are captured as given below.

$$\begin{aligned} P_{dynamic} &= \alpha_{0\to 1} \cdot C_L \cdot V_{dd}^2 \cdot f_{clk} = C_{eff} \cdot V_{dd}^2 \cdot f_{clk} \\ P_{leakage} &= I_{off\,/\,device} \cdot N_{transistors} \cdot V_{dd} \cdot (1-\alpha_{0\to 1}) \\ P_{short-circuit} &= I_{sc\,/\,worst\_case} \cdot t_{sc} \cdot f_{clk} \cdot V_{dd} \cdot \alpha_{0\to 1} \end{aligned} \qquad (1.2)$$

1.3 Power consumption estimation methodologies

The proliferation of microprocessors operating at high clock frequencies and the wide acceptance of energy-constrained mobile devices has forced low power dissipation to be a goal in the design of VLSI systems. Efficient and accurate power estimation techniques are vital for identifying the causes of excessive power dissipation and also for evaluating optimizations to address them. Consequently, many power estimation techniques have been proposed at all levels of the design process (i.e., transistor, logic architectural and behavioral levels [63]). With some practical limits being already reached at the lower abstraction levels, the responsibility of achieving power savings has shifted to the system architect. To accomplish this, the appropriate tools must be available so that, early in the design process, it can be determined whether a new design, or an optimization on an already existing design, achieves the expected savings. In order to develop these tools, models need to be formulated such that the power consumption of a given design can be estimated without having to fabricate it, or perhaps, even without any layout steps.

Transistor and logic level power estimation techniques provide the best accuracy (within 5 to 15% of silicon) but pay for good accuracy with very long run-times, hence they lack the capacity for analyzing large designs. Behavioral methodologies are very fast but their accuracy is typically within 50-100% of silicon, while architectural level techniques trade off accuracy (20-25% of silicon) for speed. Thus, architectural level methods (RT techniques) provide a good balance between speed and accuracy and also a way to find if the design will meet the expected power and performance goals relatively early in the design process. In consequence, the development of models that can be used at this level is very important.

It is clear that power estimation techniques at higher levels of abstraction are needed, as there are more opportunities and flexibility in tuning the design at the early stages. Understanding this, various high-level estimation techniques have been proposed [57, 58, 59, 60, 61, 62, 64]. These techniques can be broadly classified as top-down or bottom-up according to the availability of structural information. The existing high-level estimation techniques can also be classified as input statistics oriented or primary input oriented. The latter characterizes power consumption caused by all possible input transitions, storing the information in a table indexed by the current and previous input. The primary input oriented estimation approach is preferred when the prediction of input statistics is difficult [61]. The number of entries required for accurately capturing the power consumption increases exponentially with an increase in the number of inputs to the circuit. Clustering mechanisms to overcome this have been proposed [60, 61].

Following the abundance of various power estimation techniques, various research groups have developed frameworks [26, 25] for making architectural-level decisions about the total power budget. The work presented in this thesis for accurately estimating clock generation and distribution power was incorporated into the cycle-accurate datapath and memory energy simulator described in [25]. This was a crucial extension as it allowed the influence of the clock network on the overall system energy to be accounted for.

1.4 The role of the clock-subsystem and related work

The focus of the work presented here is on the clock subsystem as it is a critical component of synchronous digital systems, and is also known to consume up to 40% of the total power budget of current high performance microprocessors [6, 5]. *Figure 1–6* shows the approximate power breakdown of a high performance CPU [5] and that of an embedded processor [51]. It is clear that, even though the percentages vary, in both cases the clock sub-system remains one of the units that deserve to be carefully studied and designed if power savings are to be obtained.

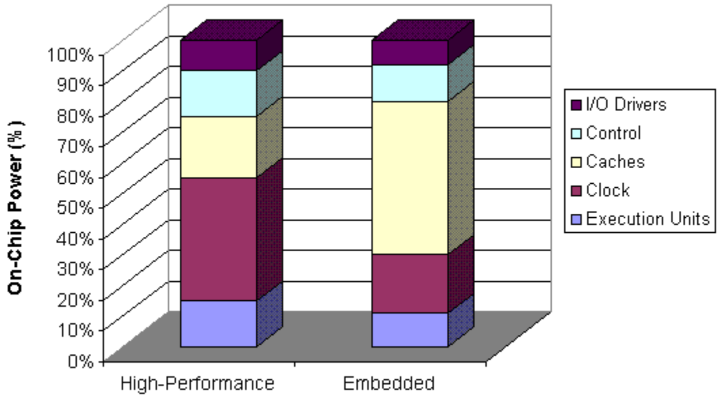


Figure 1–6: Reported power consumption breakdown for two processor types.

Initially, most of the interest in clock distribution networks was focused on the necessity of reducing clock skew, which was motivated by the objective of improving performance. This was pursued at multiple levels [7] that included circuit and layout techniques, automated synthesis tools, and the modeling of the timing characteristics of clock networks. The whole task reduced to fulfilling the architectural and functional performance specifications by defining the optimal timing behavior of the clock distribution network to be designed. From this perspective, many researchers have focused on proposing techniques and methods that allow the generation of virtually skew-free clock sub-systems. Reducing power consumption is now becoming another important design directive as well.

As the clock network delay is dominated by interconnect, it is clear that the network topology has a profound impact on its performance. The most common approaches that can be found in successful microprocessor designs are grids, trees and serpentines [8], each having particular advantages and drawbacks. Choosing one topology over the other is determined partially by the skew requirements to be met, and by other factors such as interconnect size and routing and buffer design and placement. Once the topology has been chosen, the remaining factors, namely interconnect and buffers, are considered. For the particular case of clock trees, many systematic methods have been proposed for synthesizing the placement of the clock distribution network from a clock skew schedule derived from the circuit timing information [9, 10, 11]. Other approaches have added constraints to the clock network routing problem, such as minimizing the wire length [12] and reducing power consumption by buffer insertion [13] or by switching activity reduction [14]. In a slightly different direction, some have relaxed the zero-skew requirement to a bounded-skew condition and enhanced the approach to account for semiconductor process-related factors such as parasitics and obstacle-avoidance clock routing [15]. And, as the designs complexity increases, others have presented ways to perform hierarchical clock network design [16]. Additionally, previous efforts have also been directed at proposing the use of phase-locked loops (PLL) for dynamic compensation of manufacturing and environmental variations such that self-calibrating networks are possible [17]. For the buffer placement problem, most of the previous research has focused on the distributed driver scheme (using a clock tree topology) as it provides a better power figure than the single driver scheme [18]. Although the buffer design conditions required to meet timing constraints are well known [19], many have proposed techniques to reduce the driver power [20, 21]. Similarly, others have addressed the problem of buffer placement [22].

With this quick review of clock network design techniques, one can see that a lot of effort has been devoted to designing the optimum clock network, both in terms of skew and, more recently power (the latter expected due to the big influence that it has in the total system power budget). But, the clock network has been taken as an independent unit without considering the influence that other units or even the application software has on it. Many other techniques have been proposed to reduce clock power consumption (including circuit-level methods [7, 14] and the derivation of clock distribution energy dissipation models [18, 24]) but these figures will be more meaningful if their impact is evaluated for the system as a whole.

1.5 Contributions of this thesis

The first step in the formulation of a detailed and accurate energy dissipation model is the study of all the components present in a clock network. The complexity of this task is increased by the presence of many possible network implementations, as it was discussed in the previous section. This implies that the model should be flexible such that it can be used for evaluation within different architectures. Once the basic network components have been identified, the modeling stage can proceed and the resulting power/energy expressions can be validated by comparing them against the power readings of a circuit level simulator like SPICE.

The formulated clock model provides estimates for both dynamic and leakage (sub-threshold) power consumption. The short-circuit power contribution is disregarded since it is expected to decrease relatively with respect to the dynamic and leakage components as technology is scaled down. The calculation of dynamic power basically requires the determination of the total effective switching capacitance (which depends on the switching activity factor and the total circuit capacitance). On the other hand, leakage power estimation requires knowledge about the total leakage current, which is expressed by Eq. 1.2 and is a function of both the switching activity factor and the total number of devices/gates in the circuit. These are the unknowns that must be determined by the model.

Dynamic Voltage Scaling (DVS) has been shown to provide significant dynamic energy savings by adaptively reducing voltage and frequency [39] and has been already used in commercial designs, [37, 38]. To implement such a technique, devices that provide support for frequency and power supply scaling are required. Phase-Locked Loops (PLLs) are an example of such devices and while they have usually been considered only for the frequency adjustment task, they can also be used as voltage regulators [40]. Thus, in order to exploit the DVS scheme at a higher level of abstraction, it becomes crucial to develop a model that effectively estimates the total PLL power and the incurred overhead, both in terms of delay and energy, when the PLL frequency is adjusted.

In summary, the basic contribution of this work can be stated as the definition of a clock power model that can be used at design time for evaluation of architectural trade-offs (i.e., evaluating the relative influence of each component's particular implementation and the effects of technology on the clock sub-system as an isolated unit) or that can be embedded into an architectural-level simulation tool so that the savings obtained after applying a particular clocking strategy (within the context of a real and complete clock network) can be accurately estimated. Further, a flexible and accurate PLL power consumption model would be vital to perform experiments involving DVS and to precisely analyze the overheads. Also, as many emerging SOCs employ multiple PLLs, the proposed PLL energy model will be useful in this domain as well as the contribution of the PLL to the total power budget grows.

1.6 Organization of the thesis

The remainder of the thesis is organized as follows. Chapter 2 presents the formulation and the validation of the power model for the distribution network and all related loads while Chapter 3 focuses on the clock generation circuitry. Chapter 4 justifies the helpfulness of the model by presenting various design scenarios in which valuable insights could have not been obtained without using the model. Chapter 5 discusses the expected impact of technology scaling on the power behavior of the clock subsystem while Chapter 6 presents some conclusions and formulates some issues for possible future research. Additionally, the appendices provide a set of additional information relevant to the main concepts as well as interesting results and situations that originated while performing the research presented here.

Chapter 2

AN ENERGY MODEL FOR THE CLOCK DISTRIBUTION NETWORK

The formulation of models for all the components in the clock distribution network is now presented. It is possible to differentiate three different contributing forces to the total clock load: the top-level interconnect wiring that distributes the clock signal to all areas of the chip, the buffers that amplify the clock signal locally and the actual loads that are being driven by these buffers and are composed of a mix of interconnect and active devices. There are two main factors that define the total capacitance of any of the mentioned components: design (structural) and technology-defined parameters. The first factor is evaluated on a case-by-case basis, but the technology dependence is only validated once since the formulated equations can then be used for all other cases.

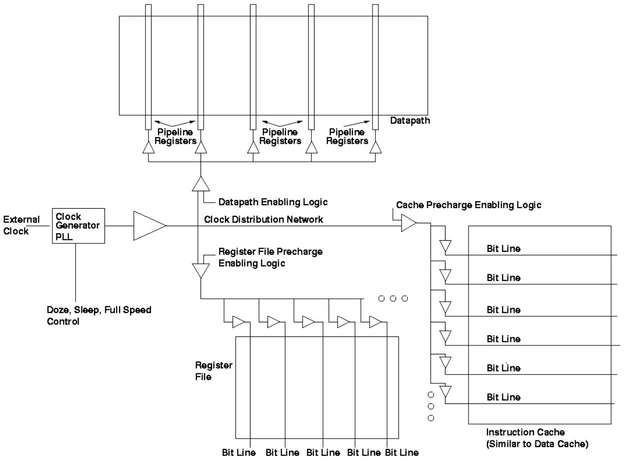


Figure 2–1: Basic loads present in the clock distribution network.

Figure 2–1 shows the usual loads present in the clock network of a modern processor (some other classic loads are not shown, such as the Translation Lookaside Buffers and the control section). As mentioned before, the estimation of dynamic power reduces to assess the total effective capacitance of the circuit, as voltage and frequency are process constants. If the estimated power is multiplied by the execution time, the energy is obtained, which is given by Eq. 2.1.

$$E = C_{clock} * V_{dd}^{2} \tag{2.1}$$

It is assumed that all components use static CMOS gates and a single phase clocking strategy for all the flip-flops and registers. The memory structures use the classic 6-transistor cell and a single PMOS transistor per bit line to provide the precharge voltage. Implicitly, the dynamic component of the total power is assumed to be the dominant one and the remaining short-circuit and leakage power contributions are disregarded, for now. Then, the task is to determine the term C_{clock}, as expressed by Eq. 2.2 where the corresponding gating factors are not shown.

$$C_{clock} = \begin{bmatrix} C_{I-cache} + C_{D-cache} + C_{I-TLB} + C_{D-TLB} + C_{RgFile} \\ + C_{PipeRegisters} + C_{PLL} + C_{Drivers} + C_{Wire} + C_{Control} \end{bmatrix} \tag{2.2}$$

2.1 Buffers

Buffers in the terminal points of the distribution network are usually built as a chain of variable size inverters and optimized for speed, as shown in *Figure 2–2* [19]. If x is the ratio of the output to the input capacitance in the chain (i.e., x = C_{out}/C_{in}), the required number of inverters can be written as in Eq. 2.3.

$$N = \frac{\ln(x)}{\ln(u)} \tag{2.3}$$

This implies that there are *N* inverters in the chain or *N+1*, if *N* happens to be odd and an extra minimum-size device is added. For this model, dynamic and static power models are presented next.

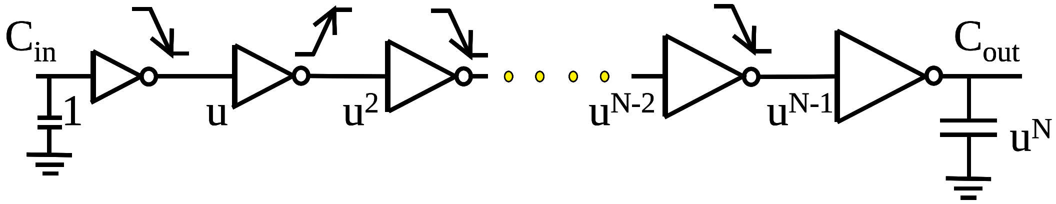


Figure 2–2: An inverter-chain based buffer.

2.1.1 Dynamic Power Model

The first inverter on the chain is built using minimum-size devices ($u^0 = 1$) while the last one, of size u^{N-1}, drives a capacitance (C_{out}) equivalent to that of an inverter of size u^N. For every transition in the input, only half of the inverters are actually demanding current from the power supply, while the other half are discharging to ground. But since in every clock cycle there are both rising and falling edges, each inverter in the chain charges its associated capacitive load once per cycle. Eq. 2.4 gives the estimation of the average sizing factor, which accounts for the situation described, while Eq. 2.5 gives the capacitance for a single driver.

$$u_{ave} = \sum_{i \in \text{even}} u^i \Big|_{\text{rising edge}} + \sum_{i \in \text{odd}} u^i \Big|_{\text{falling edge}} = \sum_{i=0}^{N-1} u^i \tag{2.4}$$

$$C_{driver} = C_{inv}u_{ave} = 4 * C_{tech}\sum_{i=0}^{N-1}u^{i} \qquad (2.5)$$

Since cascaded inverters are the basic building blocks, C_{tech} is calculated in Eq. 2.6, where W_n and W_p are the normalized widths of the N and P devices in the inverter and L_{local} is an estimate of the average length of the wire attached to a particular node. The latter increases as the width of the inverter increases and more parallel transistors are connected.

$$C_{tech} = (W_n + W_p)C_{gate} + W_n C_{drain\_n} + W_p C_{drain\_p} + C_{\text{int}} \cdot L_{local} \qquad (2.6)$$

By considering the gate (C_{gate}), drain (diffusion) (C_{drain}) and interconnect capacitance (C_{int} or C_{wire}), the effect of technology scaling is captured. The first component is estimated in Eq. 2.7 [32], where w and L_{eff} are the width and effective length of the minimum-size transistor, ε_{ox} is the dielectric constant of the oxide and t_{ox} is the oxide thickness.

$$C_{gate} = \frac{\varepsilon_{ox}A}{t_{ox}} = \frac{\varepsilon_{ox}(w \cdot L_{eff})}{t_{ox}} \qquad (2.7)$$

Similarly, C_{drain} is calculated as in Eq. 2.8 [19], where L_s is the diffusion zone length, C_j is the junction capacitance and C_{jsw} is the sidewall junction capacitance. The interconnect capacitance per unit length (C_{int}) is calculated using the area (C_{area}), fringe (C_{fringe}) and interwire (C_{intw}) components, such that $C_{int}=C_{area}.Width_{wire} + C_{fringe}+2C_{intw}$. Details about the calculation of C_{int} are given in section 2.4. *Figure 2–3* shows how the components of C_{tech} vary.

$$C_{drain} = C_{diff} = C_j L_S w + C_{jsw}\left[2L_S + w\right] \qquad (2.8)$$

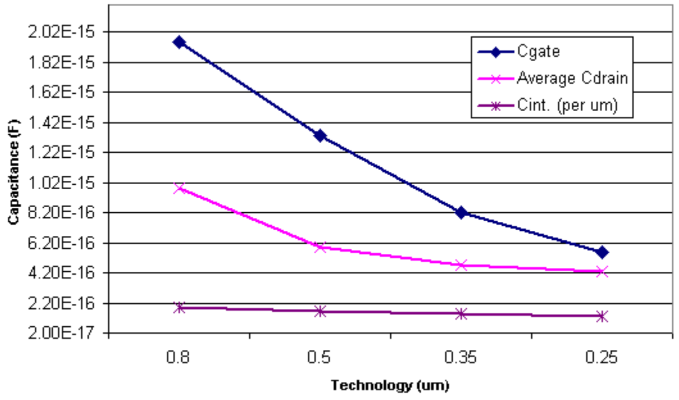


Figure 2–3: Variation of parasitic capacitances across technology generations. Data taken from SPICE technology files after estimating the effect of coupling capacitance.

Various buffers of different lengths were implemented using Berkeley's MAGIC CAD tool and simulated with SPICE. *Table 2–1* lists the technologies used for evaluation and the corresponding fabrication parameters.

Table 2–1: Parameters of the technologies used.

Tech.(um)	t_{ox} (nm)	C_{gate} (fF)	C_{drain} (fF)	V_{th} (n/p)	V_{dd} (V)
0.18	4.08	0.41	0.14	0.41/-0.43	1.8
0.25	5.80	0.55	0.43	0.43/-0.61	2.2
0.35	7.70	0.82	0.47	0.50/-0.68	2.5
0.5	9.70	1.33	0.59	0.65/-0.84	3.3
0.8	17.0	1.95	0.97	0.67/-0.84	3.3

For the validation of the model, the number of stages (N) and the technology used (C_{tech}) are the factors whose influence is verified. For the simulations, u was fixed to 3 and the results are presented in *Figure 2–4* (the curves of estimated values are so close that they are not shown for the sake of neatness). It was found that the average error across all technologies and design options (i.e., all values of *N*) was 5.29%. *Table 2–2* lists the average and maximum values for each case.

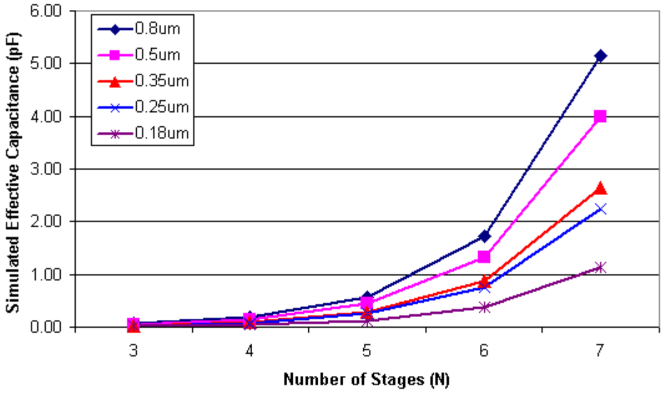


Figure 2–4: Simulated buffer capacitance as function of buffer depth.

Table 2–2: Buffer power deviation with respect to simulated values.

Tech. (um)	0.18	0.25	0.35	0.5	0.8
Average Error	2.53	9.48	3.61	5.63	5.22
Maximum Error	4.71	10.68	9.86	9.25	9.90

There are two aspects present during technology scaling that deserve to be mentioned. First, the continuous reduction of threshold voltage (V_{th}) influences the transistor's current driving capabilities, since its magnitude is a function of the difference (V_{dd}-V_{th}). The classic dependence on the square of this difference holds for long channel devices [19], but as the feature size shrinks below 1.0um, velocity saturation (v_{max}) effects under high electric fields begin to dominate. This situation forces the current to be a linear function of the voltage, which can be expressed as written in Eq. 2.9 [19].

$$I_{cell} = k(V_{ds})wC_{ox}v_{sat}(V_{gs} - V_{Tn}) \qquad (2.9)$$

In the above expression, the increment in C_{ox} and the decrease in w combine with the variations in (V_{dd}-V_{th}), such that, if the appropriate scaling is applied, the magnitude remains basically constant. Otherwise, the average gate current increases, which should be accounted for when estimating power consumption. The second aspect is the behavior of the short-circuit current, which can be estimated as in Eq. 2.10 [2], where τ and T_{clk} are the rise/fall time of the signal and its period (average time between transitions), respectively.

$$P_{short-circuit/inverter} = \frac{k}{12}\left(V_{dd} - 2V_{th}\right)^3 \frac{\tau}{T_{clk}} \qquad (2.10)$$

In a perfect scaling scenario where the supply and threshold voltage difference also scales down, and both τ and T_{clk} decrease as the operating frequency increases, it can be stated that short-circuit power also reduces, or in the worst case, remains proportionally constant. Appendix B discusses cases where some of the above conditions were found or introduced, which resulted in an increased contribution of short-circuit power to the total power consumption. Now, by matching the rise/fall times of the input and output signals, the power dissipation of short circuit currents is minimized and confined to be within 10% of the switching power dissipation [19]. For the particular case of the buffers, Eq. 2.11 presents an estimate of the rise/fall time for the n^{th} inverter in the buffer chain.

$$t_{rise/fall\_stage(n)} = \frac{C_{inv(n+1)}V_{dd}}{I_{inv(n)}} = \frac{u^{n+1}C_{inv\_\min}V_{dd}}{u^n I_{inv\_\min}} = u\frac{C_{inv\_\min}V_{dd}}{I_{inv\_\min}} \qquad (2.11)$$

In Eq. 2.11, C_{inv_min} and I_{inv_min} are the input capacitance and drive current of a minimum-size inverter, respectively. Since the above expression is independent of n, the input rise/fall times are identical to the output ones at all points in the chain, as long as the PMOS transistors are sized appropriately to compensate for their lower mobility.

Thus, by maintaining the sizing factor (*u*) across the chain, the short-circuit current contribution to the dynamic power is bound and could account for the error in *Table 2–2*.

2.1.2 Static Power Model

The buffers are expected to have the largest influence on the network's leakage due to their large width and absence of stacked devices in the inverter chain. Since it has been shown that the effects of sub-threshold currents dominate the diode junction (or PN reverse bias) leakage contribution [3], only the influence of the former is considered. Following a similar reasoning to that presented for the derivation of the buffer's dynamic power, Eq. 2.12 shows the leakage current modeled as a function of the number of stages in the buffer.

$$I_{buff-leak} = \left(\frac{I_{off-n} + I_{off-p}}{2} \right) w \sum_{n=0}^{N-1} u^n \qquad (2.12)$$

In Eq. 2.12, the first factor is the average sub-threshold current (I_{off}) for N and P devices (in A/um). When the input is high there is leakage through certain P and N devices, while when the input is low, the situation is reversed. As before, *w* is the width of a minimum-size transistor and the sum gives an average across all stages. Eq. 2.12 simply needs to be multiplied by the supply voltage to obtain the leakage power.

Figure 2–5 shows the data obtained from SPICE for three different technologies. As expected, leakage increases with decreasing feature size and the data follows the trend predicted using the model for a variable number of stages (the curves of estimated values are so close that they are not shown for the sake of neatness). The average error was 2% for 0.25um, 6% for 0.35um and 9% for 0.8um. This decreasing error for decreasing feature size can be explained by the fact that the magnitude of the leakage current increases and is then easier to differentiate from noise and transient fluctuations. These results justify the formulation of a technique for minimizing buffer leakage power.

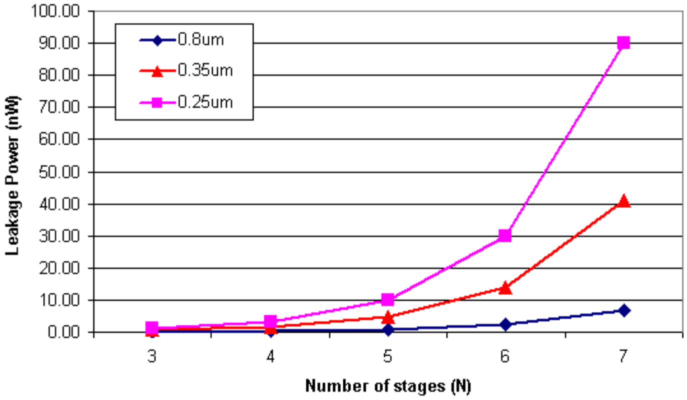


Figure 2–5: Buffer power leakage reported by SPICE as function of buffer depth.

2.2 Memory structures

Memory structures include data and instruction caches, data and instruction translation lookaside buffers (TLBs), register files, branch history tables in the branch predictor, instruction issue windows, the load/store queue and other similar constructs. Given the complexity of formulating a model for each particular case, a general model is presented which can be tailored to the particular features of a given construct.

2.2.1 A general memory structure precharge model

Figure 2–6 shows the schematic of the generic memory design, which can also be seen as a classic cache design [27]. Since only Static RAM (SRAM) is assumed to be present on-chip, the classic 6-transistor cell is used for all estimates, assuming a single PMOS transistor per bit line to provide the precharge voltage, unless stated otherwise.

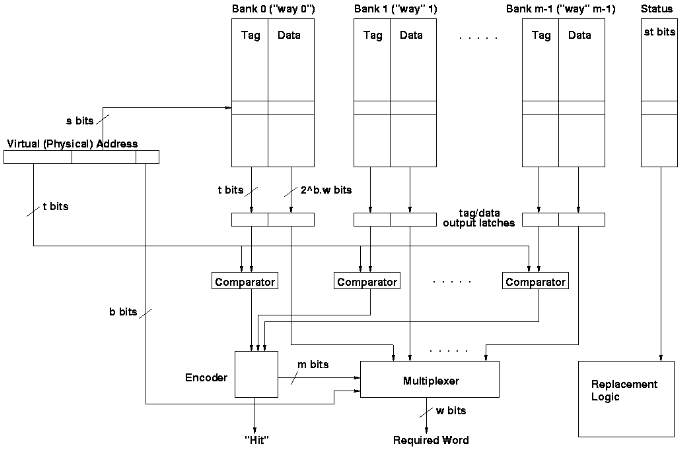


Figure 2–6: Schematic of an m-way set associative memory structure

Assuming an address with *t* bits of tag, *s* bits for the set index and *b* bits for the word index, the following observations result:

- 2^s line-tag pairs per way, so there are $N_{ic} = 2^s$ rows.
- 2^b words constitute the line (block) size and assuming a word size of *w* bits, the number of columns per way is then $n_{ic} = [(2^b w) + t]$.

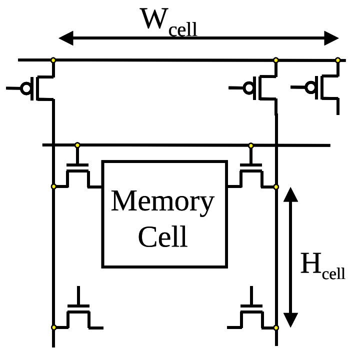


Figure 2–7: A detailed schematic for the bitline load capacitance extraction.

The gate capacitance of the precharge transistor is assumed to be proportional to the bit line capacitance but scaled down by a factor β. This factor is normally chosen to be around 30, as circuit-level simulation shows that a bitline of 32 cells can be driven effectively with a minimum size PMOS transistor. Note that during precharge, there is no rail-to-rail swing as bitline isolation is assumed (and also widely used to reduce cache power consumption). For faster memory structures, like register files, this β factor can be in the range of 15 to 20. The bit line capacitance per way is then given by Eq. 2.13, based on the structure shown in *Figure 2–7*.

$$C_{bl\,/\,way} = N_{ic}\left[C_{wire}H_{cell} + C_{drain}W_{tr}\right] \qquad (2.13)$$

In Eq. 2.13, the first term represents the interconnect capacitance (H_{cell} = cell height and C_{wire} expressing the wire capacitance per unit length) and the second represents the capacitance due to the pass transistor (where W_{tr} is the size factor and C_{drain} is the drain diffusion capacitance). The total precharge capacitance per way is given by Eq. 2.14, where the first term represents interconnect capacitance (W_{cell} = cell width) and the second captures the gate capacitance of the precharge transistors (one per bit line, for a total of 2).

$$C_{prech/way} = n_{ic}\left[C_{wire}W_{cell} + 2\left(\frac{C_{bl/way}}{\beta}\right)\right] \qquad (2.14)$$

For the whole array, Eq. 2.15 gives the total capacitance, where the *k* factor accounts for the additional power overhead if intermediate buffers are included between ways, which are required if a technique like sub-banking is implemented. The last term is considered only if these buffers are present (W_{way} is the width of a bank or “way”, as shown in *Figure 2–8*).

$$C_{prech} = \begin{Bmatrix} m[C_{prech/way}](1+k) + C_{I\$-status}(1+k) + \\ (m-1)C_{wire}W_{way} \end{Bmatrix} \tag{2.15}$$

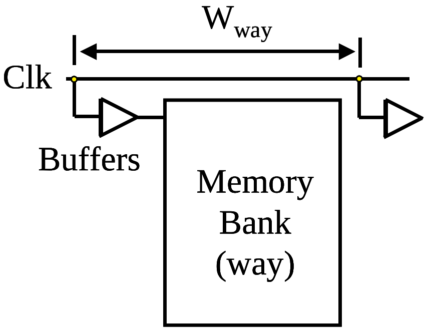


Figure 2–8: Details of the parameters associated when buffers are inserted between ways.

According to the rules presented in the previous section, if the buffer is sized to be *u* times smaller than the $C_{prech/way}$ load it is driving, then $k = u^{-1}$. Thus, for the nominal case of *u* = 3, then *k*= 0.33. The second term in the equation above is the additional load due to the storage required for the status bits that support the replacement policy, if any. The effective capacitance per memory read can simply be expressed as the factor C_{prech}, if the influence of the control logic is discarded given that the size of the memory is large, which applies to current cache designs. Since a write usually requires a read to verify data status (only the tag might actually be read, but the whole array might be precharged), the model can also be used in this case. The models can be easily tailored to the specific cases of caches, register files and TLBs as discussed in the next section.

Figure 2–9 shows both the estimated and simulated bitline capacitances ($C_{bl/way}$), for the indicated number of rows. The estimated values were calculated using Eq. 2.13 for $C_{bl/way}$, taking W_{tr} equal to 1, C_{drain} as given in *Table 2–1*, C_{wire} as determined for a 0.35μm process and H_{cell} as measured from the actual layout. The simulated values were obtained using SPICE. The average (maximum) deviation is 5%(12%) with respect to the simulated values. The error decreases as the number of rows increases (1.31% for 32 rows and 0.96% in the case of 64 rows). For the other dimension of the memory array, a similar equation was obtained ($C_{prech/way}$) and the validation results provided an accuracy similar to that reported above for the bitline capacitance.

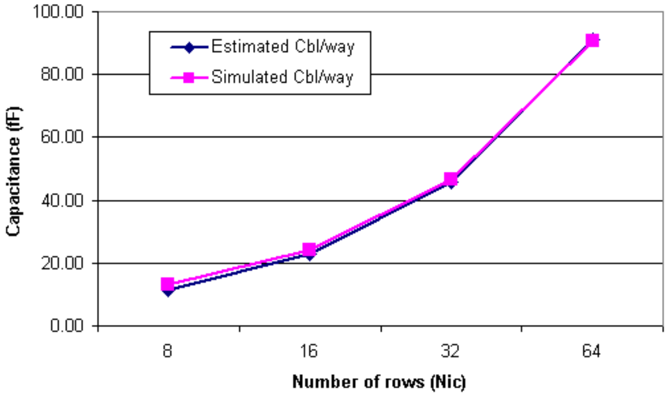


Figure 2–9: Estimated and simulated bitline capacitance.

Regarding leakage power, the only active device present (besides the buffers, which were already addressed) is the precharge transistor. In general, this device does not have a large width, but there are many of them. They are, however, part of the actual memory and not of the clock network, as only its capacitive load is of interest. Then, the leakage contribution due to the memory structures load is assumed to be negligible.

2.2.2 Adapting the Model

The caches, TLB's and register file are the structures initially considered for study as they are the most common constructs present in single-issue general purpose and embedded processors but any other similar unit can be modeled in an analogous way.

$$C_{rf-bl} = N_{rf}\left[C_{wire}H_{cell} + C_{drain}W_{tr}\right] \qquad (2.16)$$

The register file can be considered a fully associative cache, distributed as a square array of 6-transistor cells, with n_{rf} columns and N_{rf} rows. The presence of X read-ports and Y write-ports is assumed. Using the same assumptions made for the memory read energy consumption estimation, Eq. 2.16 gives the bit line capacitance, where all the terms were described earlier. The total precharge capacitance per read access, per port, is given by Eq. 2.17, where, now an α factor is used instead of β, as the register file might be expected to be faster than the caches. It should be noted that the multiplicative factor of 2 is dropped, as it is common to use only one bit line per read port [28].

$$C_{rf-prech} = n_{rf}\left[C_{wire}W_{cell} + \left(\frac{C_{rf-bl}}{\alpha}\right)\right] \qquad (2.17)$$

The total effective capacitance for a register file read is given by Eq. 2.18, where the capacitance due to the control logic might not be negligible, as the size of the register file is usually small compared to the size of the cache. In the case of a write, it can be assumed that $C_{rf/write}$ is approximately equal to $C_{rf\text{-}control}$. An estimate of the $C_{rf\text{-}control}$ is difficult to obtain as various circuit implementations are possible, but it must be noted that this term captures the load that the control circuitry produces on the clock network, and not the power consumed by the switching of the control circuitry itself.

$$C_{rf/read} = X \cdot C_{rf-prech} + C_{rf-control} \qquad (2.18)$$

For the TLB analysis, an m-way set associative design is assumed, following a similar structure to the one used for the general case analysis [29]. Now, an address has t_b bits of tag and s_b bits for the set index, which results in:

- 2^{Sb} data-tag pairs per way, so there are $N_{it} = 2^{Sb}$ rows
- There is only one data word (one physical address) per line (block) size and assuming a word size of w_b bits, the number of columns per way is $n_{it} = [w_b + t_b]$.

From previous analysis, and after renaming the scale-down factor to γ, so that a different value can be defined for the TLB design, the total precharge capacitance per way is given by Eq. 2.19. For the whole array, assuming that no distributed buffers are used, Eq. 2.20 gives the total TLB precharge capacitive load.

$$C_{prech/way} = n_{it}\left[C_{wire}W_{cell} + 2\left(\frac{N_{it}\left[C_{wire}H_{cell} + C_{drain}W_{tr}\right]}{\gamma}\right)\right] \tag{2.19}$$

$$C_{TLB-prech} = m \cdot C_{prech/way} + C_{TLB-status} \tag{2.20}$$

The second term is the additional load due to storage required for the status bits and is essentially equal to $C_{prech/way}$ but with n_{it} replaced by s_{tb} (the number of status bits per line).

2.3 Dynamic and sequential Circuitry

The pipeline registers capture the sequential aspect of the base design, as the basic assumption is that all logic is built using static CMOS. However, this does not hold true for recent high performance systems, which use dynamic logic aggressively. The impact of pipeline registers is first discussed and the impact of other sequential loads is addressed later.

2.3.1 Pipeline registers

Assuming that there are M_n flip-flops for register n and a total of N_p registers, the effective capacitance per register is given by Eq. 2.21, where C_{ff} is the flip-flop clock input capacitance and H_{ff} is the average height of the flip-flop (which is used to determine the interconnect capacitance).

$$C_{reg-n} = M_n (C_{ff} + C_{wire} H_{ff}) \qquad (2.21)$$

In general, the flip-flop clock load is calculated as $C_{ff} = N_{t\_clk} C_{gate}$, where N_{t_clk} is the number of clock-driven transistors per flip-flop and also accounts for any transistor sizing. There is one buffer per register, such that gating functionality is provided. Eq. 2.22 gives the effective capacitance for all registers in the pipeline.

$$C_{pipe-registers} = \sum_{n=1}^{N_p} [C_{reg-n}(1+k)G_n + C_{wire} W_{pipe-stage-n}] \qquad (2.22)$$

Where k represents the additional power consumed by the buffers, G_n is the gating factor (i.e., the fraction of flip-flops being gated, 0 = all, 1 = none) and $W_{pipe-stage-n}$ accounts for the width of stage n, so that the interconnect capacitance is also included, as shown in *Figure 2–10*.

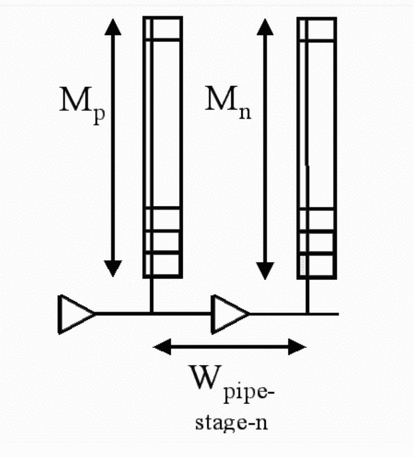


Figure 2–10: Schematic of the pipeline registers.

The validation is performed for the case of $N_p = 1$. The estimated effective capacitance (as predicted by Eq. 2.22) and the simulated effective capacitance (as inferred from SPICE power readings) are shown in *Figure 2–11*, for a non-buffered implementation (k = 0). The variable is the size of the register (M_n), with C_{ff} estimated as 2 times the technology-defined capacitance (C_{tech}) and the height of the flip-flop (in lambdas) taken from the layout. The maximum deviation from the measured values was 2.21% with an average error of 1.21%, which is reasonable due to the regular structure of the register. The influence of the driver, can be estimated in a similar way to that presented earlier, where $k = u^{-1}$, u being the buffer scaling factor.

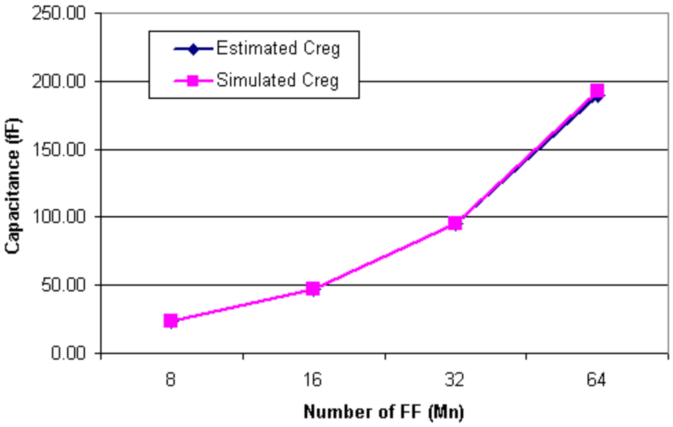


Figure 2–11: The estimated and simulated pipeline register capacitance.

In terms of leakage, a similar situation to that present with the memory structures occurs, as the clock inputs of the flip-flops in the registers are used to estimate the load, but the flip-flops themselves are not really part of the clock network, thus no leakage can be calculated.

2.3.2 Dynamic logic and other sequential circuitry

If there are dynamic gates on the design, the associated load on the clock network can be estimated simply as the number of transistors being clocked times the technology capacitance (C_{tech}) given in Eq. 2.6. The number of transistors clocked can be estimated as the number of dynamic gates times the average number of clocked transistors per gate. As gates are grouped depending on which functional unit they are part of, the calculation of the associated buffers would be only an approximation as this type of information might not be available in the early design stages.

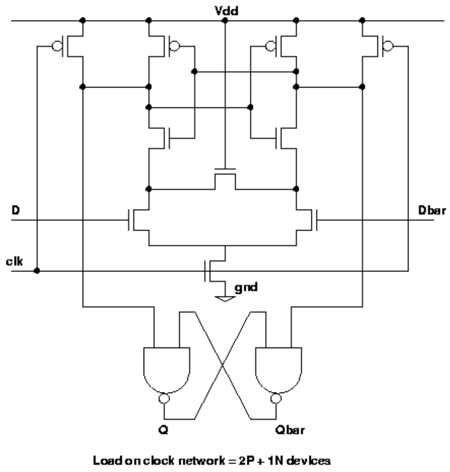


Figure 2–12: Schematic and clock load of a StrongArm flip-flop.

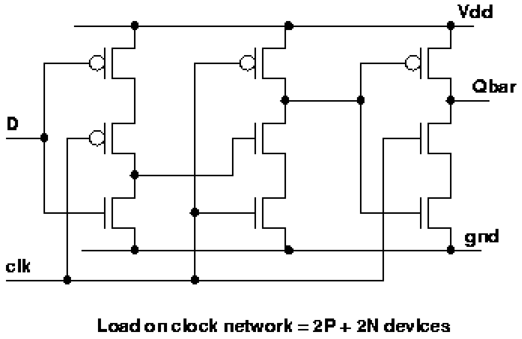


Figure 2–13: Schematic and clock load of a TSPC flip-flop.

For additional sequential components, besides the pipeline registers, the basic requirement is the determination of the number of clocked devices per flip-flop (N_{t_clk}), such that $C_{ff} = N_{t\_clk}C_{gate}$, as stated earlier. This is obviously dependent on the type of flip-flop used. *Figure 2–12* and *Figure 2–13* show the designs of two widely used flip-flops and the estimated load that each design present to the clock network.

2.4 Distribution network

As mentioned earlier, there are various topologies that can be used to implement a clock distribution network. The H-tree topology employs a fractal-like style, where at each subsequent level, the H-tree shape is replicated until the load driven at the terminating points of the network meets the target. The grid approach attempts to make the clock available at every point in the die, regardless of whether it is needed or not. The serpentines-based topology represents the opposite of the grid approach since, in this case, each load is driven by a single point-to-point wire and lengths are matched (for skew equalization) using a serpentine structure as shown in *Figure 2–14*.

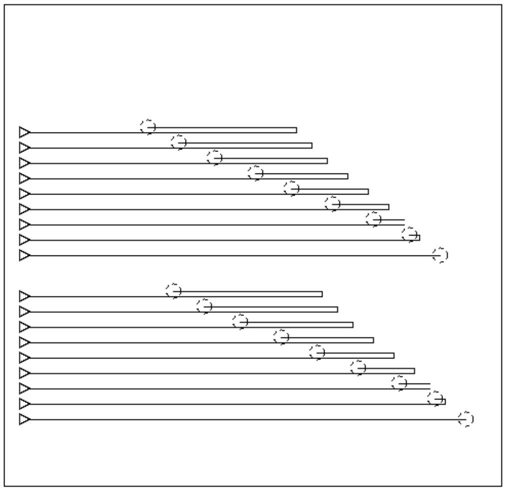


Figure 2–14: Schematic of a length-matched serpentine-based clock distribution network.

For a uniform load distribution, trees provide low power consumption and skew while grids minimize delay at the expense of a larger capacitive load. Serpentines also provide low skew but they incur larger power consumption. An interesting comparison between these three topologies can be found in [8]. In some recent designs, trees and grids have been used simultaneously in order to combine the advantages of each technique and produce clock signals with very low skew, minimum delay and high tolerance to process variations [23].

The effective capacitive load produced by the interconnect network is an important factor and expressions to estimate it are discussed next. Given that it is impossible to determine the layout of a serpentine distribution scheme before the actual implementation, only trees and grids are considered.

2.4.1 Grids

For this analysis, it will be assumed that C_{pp} defines the wire parallel-plate capacitance per unit length, which includes both the area and the fringe components while C_{intw} defines the interwire or coupling capacitance. From [65], C_{pp} and C_{intw} can be estimated using the equations below.

$$C_{pp} = \varepsilon_{ox}\left[C_{area}(W_{wire}) + C_{fringe}\right] = \varepsilon_{ox}\left[1.15\left(\frac{W_{wire}}{T_{fox}}\right) + 2.8\left(\frac{T_{wire}}{T_{fox}}\right)^{0.222}\right] \quad (F/um) \qquad (2.23)$$

$$C_{\mathrm{int}\,w}(W_{sp}) = \varepsilon_{ox}\left[0.03\left(\frac{W_{wire}}{T_{fox}}\right) + 0.83\left(\frac{T_{wire}}{T_{fox}}\right) - 0.07\left(\frac{T_{wire}}{T_{fox}}\right)^{0.222}\right]\left(\frac{W_{sp}}{T_{fox}}\right) \quad (F/um) \qquad (2.24)$$

In the above equations, W_{wire} is the wire width, W_{sp} is the separation between adjacent lines and T_{fox} and T_{wire} are the field oxide and wire thickness, respectively. If an interconnect line is surrounded by two lines, then the total wire capacitance per unit length is $C_{wire} = C_{pp} + 2C_{intw}$. This, however, is not the general case.

For the estimation of the total C_{pp} for the grid, the contribution of all lines is added while neglecting the impact of the intersecting points and assuming that the same wire width is used for all lines. For the general case of a rectangular grid, the total C_{pp} is given by Eq. 2.25.

$$C_{pp\_grid}(\Delta L) = C_{pp}\left[(n_y + 1)n_x \Delta L_x + (n_x + 1)n_y \Delta L_y\right] \tag{2.25}$$

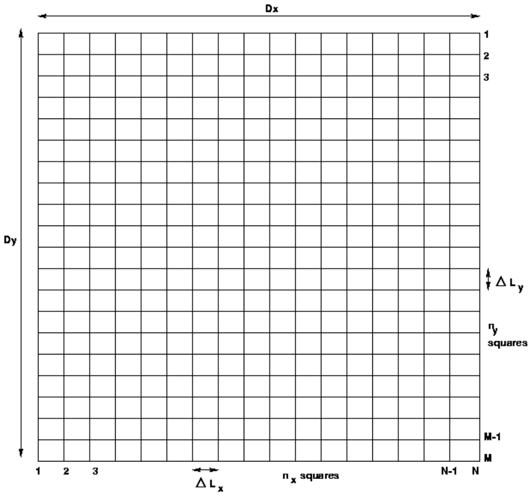


Figure 2–15: Schematic of a grid-based clock distribution network.

Where ΔL_v is the side length of the minimum-size square and n_v represents the number of squares along the *v* (= *x* or *y*) dimension, as shown in *Figure 2–15*. If the grid is symmetric in both dimensions ($n_y = n_x$ and $\Delta L_x = \Delta L_y$) and given that $n_x = D_x/\Delta L_x$, the total parallel plate capacitance for a grid can be simplified to the expression below.

$$C_{pp\_grid}(\Delta L) = 2C_{pp}\left[(n_x + 1)n_x \Delta L_x\right] = 2C_{pp}\left[\left(\frac{D}{\Delta L} + 1\right)D\right] \tag{2.26}$$

For the interwire component, it is clear that W_{sp} could be equal to ΔL_x or ΔL_y. Under the same assumption that the grid is symmetric and after considering that the four lines on the boundary only present coupling to one other line each, the interwire capacitance for the grid is given by Eq. 2.27, where C_{intw} is calculated for $W_{sp}= \Delta L$.

$$C_{cp\_grid}(\Delta L) = C_{\mathrm{int}\,w}\left[4n_x^2\Delta L\right] = C_{\mathrm{int}\,w}\left[4\frac{D^2}{\Delta L}\right] \qquad (2.27)$$

2.4.2 Trees

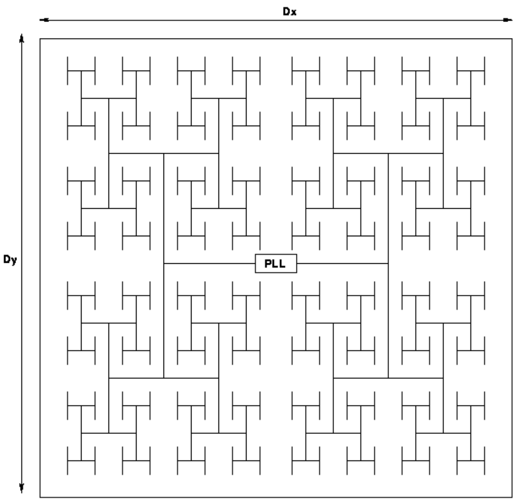


Figure 2–16: Schematic of a tree-based clock distribution network.

The estimation of the parallel plate capacitance for a tree (shown in *Figure 2–16*) requires the addition of the contributions from all branches. It is assumed that the tree has *p* levels and the wire width of the top level is *2p* times the minimum wire width, while the lowest level uses the minimum width. Because of this change in width from one level to the next, the determination of the area and fringe components of the parallel plate capacitance must be done independently. For the area capacitance, the sum of all contributions is given by Eq 2.28.

$$C_{area\_tree}(p) = 3\frac{D_x}{2}C_{area}\sum_{j=1}^{p} 2^{(p+1-2j)}4^{j-1} = 3\frac{D_x}{2}C_{area}\sum_{j=1}^{p} 2^{p-1} = 3\cdot C_{area}\frac{D_x}{2}2^{p-1}p \qquad (2.28)$$

A similar analysis with the fringe component yields the expression below.

$$C_{fringe\_tree}(p) = 3\cdot C_{fringe}\frac{D_x}{2}\left(2^p - 1\right) \qquad (2.29)$$

For the interwire component, the analysis complexity increases significantly. It was found that for trees with more than 2 levels, which is the usual case, an iterative formula can be established. The interwire capacitance of the tree can be approximated to that of the lowest level only, as its branches have an average separation of half that of the previous level. Under this assumption and after a lot of algebraic manipulation, the coupling capacitance for a p-level tree is estimated by Eq. 2.30.

$$C_{\text{int}\,w}(p) = D_x 4^{p-1}\left\{\begin{array}{l} C_{\text{int}\,w}(D_x/2^{p+1})\cdot\left(\frac{1}{2^p}+\frac{3}{2^{p+2}}\right)+C_{\text{int}\,w}(D_x/2^p)\cdot\left(\frac{3}{2^p}+\frac{p-1}{2^{p+3}}\right)+ \\ C_{\text{int}\,w}(D_x/2^{p-1})\cdot\left(\frac{1}{2^{p+1}}+\frac{p-1}{2^{p+3}}\right) \end{array}\right. \qquad (2.30)$$

The equations derived for the distribution network are not validated given that the base expressions (Eqs. 2.23 and 2.24) have already been verified and are similar to those used by circuit-level simulators for gate-level interconnect capacitance estimation. Thus, the above equations will be used for future experiments under the assumption that they were correctly derived. Since the top-level distribution network has been reported to contribute less than 10% of the to the total clock power, the accuracy of the above expressions is important but not crucial.

Chapter 3

PHASE-LOCKED LOOP POWER MODELING

3.1 The PLL dynamic behavior

The dynamic behavior of the PLL has been widely studied [43, 46], and only the most relevant equations are repeated here. Initial PLL implementations omitted the charge pump circuit, but most recent designs use it because of two main reasons: the PLL capture (pull-in) range is now limited only by the VCO frequency and the static phase error is practically zero. The latter is crucial for clock distribution because it means that the output frequency follows the input without phase difference, independent of the frequency value. *Figure 3–1* shows the schematic of the base PLL design.

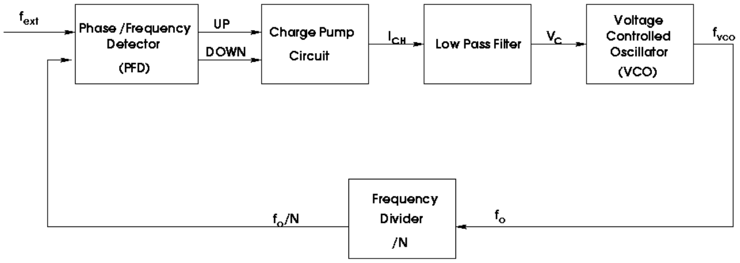


Figure 3–1: Schematic of a phase-pocked loop.

Assuming that the VCO has a gain K_V (Hz/V), the charging/discharging current in the charge pump is I_{CH} and the low pass filter is defined by a resistance (R) and a capacitance (*C*), the output frequency (which is referred to as f_o or f_{vco}) as function of the input frequency (called either f_i or f_{ext}) is expressed in the *S* domain by Eq. 3.1 [44].

$$H(s) = \frac{F_o(s)}{F_i(s)} = N \frac{1 + 2(\xi / \omega_n)s}{1 + 2(\xi / \omega_n)s + (s / \omega_n)^2} \qquad (3.1)$$

It is clear that the PLL can be described as a 2nd order continuous-time system, including the well known factors, ξ (damping factor) and ω_n (natural frequency). These two parameters, given by Eq. 3.2, are important because the first one determines the amount of overshoot in the response when a step is applied to the input, while the second dictates the stability limits of the PLL.

$$\xi = \frac{R}{2}\sqrt{\left(\frac{1}{N}\right) K_V I_{CH} C} \qquad \omega_n = \sqrt{\frac{K_V I_{CH}}{CN}} \qquad (3.2)$$

It must be noted that, in reality, charge-pump PLLs are discrete-time systems [45]. It has been shown [46], however, that when the PLL complies with the classic stability requirement (i.e., ω_n is approximately ten times smaller than the minimum frequency at which the PLL operates), the continuous-domain frequency response given above accurately describes the PLL behavior. If there is an instantaneous change (a step of magnitude A_{step}) in the input frequency, the time response of the output frequency is given by Eq. 3.3 while Eq. 3.4 defines some parameters used in the former.

$$f_o(t) = A_{step} N [1 - (1 / \omega_d) e^{-bt} \cos(\omega_d t + \varphi)] \qquad (3.3)$$

$$b = \xi \omega_n \quad ; \quad \omega_d = \omega_n \sqrt{1 - \xi^2} \quad ; \quad \varphi = \sin^{-1}(b) \qquad (3.4)$$

If ξ is too large, the response is slow while if ξ is too small, the overshot and oscillation become significant. The optimum value has usually been taken to be 0.707.

Using both the time and frequency responses of the PLL, it is possible to define a set of parameters that accurately describe the PLL dynamic behavior. These include the hold, lock, pull-in (capture) and pull-out range and the reader is referred to specialized literature for further details [47]. Depending on the application the PLL is intended for, some or all of the above factors must be considered during the design stage. For the particular case considered here, the lock range is the parameter of greatest interest since it defines the maximum instantaneous input change that will not cause the PLL to lose lock. An approximation of the lock range is given by Eq. 3.5.

$$\Delta\omega_{lock} \approx 4\pi\xi\omega_n \tag{3.5}$$

Similarly, time-domain equivalents of these parameters have been defined, the lock and capture times being the most useful ones. Here, the focus is on time-domain parameters as they are needed in order to estimate the energy spent by the lock and capture process transients. These timing issues are covered in Section 3.3.

3.2 Design parameters: definition and validation

The expressions introduced in the previous section are all dependent on four parameters: K_V, I_{CH}, R and C. Thus, it is important that these parameters are accurately calculated from the technology and implementation dependent factors, which is performed next. For an example of how the parameters are defined, refer to the design example given in Appendix C.

The parameter considered first is the VCO gain. Most VCO designs are based on the widely used ring oscillator implementation that employs current-starved inverters. The stages on the ring oscillator can be either differential or non-differential, with the former having a better noise immunity. For low power, the number of stages (n) should be small but at least four stages are recommended [31]. As will be shown later, this limit also applies for stability purposes. *Figure 3–2* shows the non-differential cell while *Figure 3–3* shows the differential basic cell. Note how the load of the differential cell is built as the parallel connection of two devices (this is called a symmetric load): one biased in the resistive region and the other in saturation. This helps to obtain a more linear V vs. I load characteristic [41] and a more stable voltage swing across all frequencies.

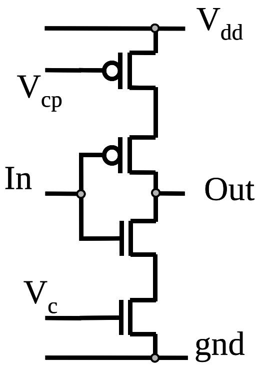


Figure 3–2: A single-ended VCO cell.

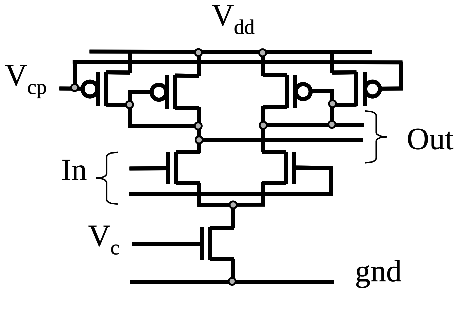


Figure 3–3: A differential VCO cell.

The oscillation frequency for an n-stage (i.e., n-cell) VCO is given by Eq. 3.6, where V_{sw} is the voltage excursion at each cell (normally close to V_{dd}), C_{cell} is the capacitive load of a cell and I_{cell} is the current that charges/discharges this load.

$$f_{vco} = \frac{1}{2nt_d} = \frac{I_{cell}}{2nV_{sw}C_{cell}} \tag{3.6}$$

The cell current is varied such that the frequency can be controlled accordingly and it follows the relationship for the transistor's drain-source current. As mentioned in the previous chapter, the effects of velocity saturation and mobility degradation, which are inherent to deep sub-micron technologies (i.e., 0.35um and smaller technologies), must be considered. Eq. 3.7 is repeated here for convenience where *w* is the transistor width, v_{sat} is the saturation velocity and $k(V_{DS})$ is given by Eq. 3.8.

$$I_{cell} = k(V_{ds})wC_{ox}v_{sat}(V_{gs} - V_{th}) \tag{3.7}$$

$$k(V_{ds}) = \frac{1}{1+(E/E_{sat})} = \frac{1}{1+V/(LE_{sat})} \tag{3.8}$$

To determine K_V, the expression df_{vco}/dV_C must be calculated, resulting in Eq. 3.9 after replacing V_{gs} with V_C. Various versions of differential and single-ended VCOs for different values of *n* were implemented using Berkeley's MAGIC CAD Tool in a 0.35um technology with a 2.5V supply voltage. The number of stages is kept small so that high VCO operating frequencies can be reached. *Figure 3–4* shows the resulting frequency vs. voltage characteristic for non-differential VCOs. A similar plot was also obtained for differential designs.

$$K_V = \frac{df_{vco}}{dV_C} = \frac{k(V_{ds})wC_{ox}v_{sat}}{2nV_{sw}C_{cell}} \tag{3.9}$$

Figure 3–4 clearly shows the nonlinear behavior of the VCO frequency versus the control voltage. It is possible to calculate two values for K_V: one for V_C below the limit value of V_{CL} and other for V_C above V_{CL}. In this way, straight-line approximations can be made for the curve segments present in each of the two regions mentioned. This threshold can be estimated by calculating the control voltage that causes the bias circuitry (a current mirror) to leave the saturation regime and move into the resistive region, condition given by Eq. 3.10.

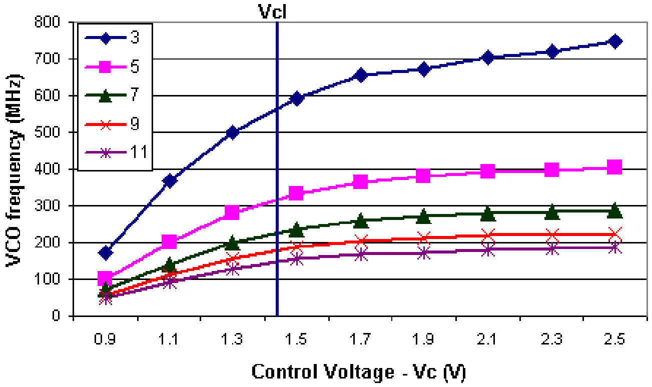


Figure 3–4: Frequency dependence with respect to V_c for non- differential VCOs with various number of stages.

$$V_C = V_{tn} + \frac{(V_{dd} - V_{tp})}{1 + (K_n / K_p)} \tag{3.10}$$

After plugging in all numbers, V_{CL} equals 1.5V. However, an improved design of the bias circuitry can effectively increase the value of V_{CL}, making it feasible to work with only one average value of K_V for the whole dynamic range of operation. Under this assumption, *Figure 3–5* shows the average simulated (i.e., obtained from SPICE simulations) and calculated K_V values, using the equation above. It must be highlighted that, even in the case where V_{CL} remains around a value of 1.5V, an average for the region of $V_C > V_{CL}$ can be derived from the calculated K_V by dividing it by a factor. On non-differential VCOs this factor is 5.5 and for differential VCOs, it is 4. An average value across both regions can also be obtained by dividing the calculated K_V by 2, for both single-ended and differential. The average error of the estimated K_V with respect to the data obtained from SPICE simulations was 2.2% and 3.2% for non-differential and differential implementations, respectively. This error drops to 0.9% and 1.1% if the implementations for *n* = 3 (non-differential) and for *n* = 5 (differential) are not considered, assuming that they operate beyond the limits of frequency stability, which results in unreasonable jitter performance.

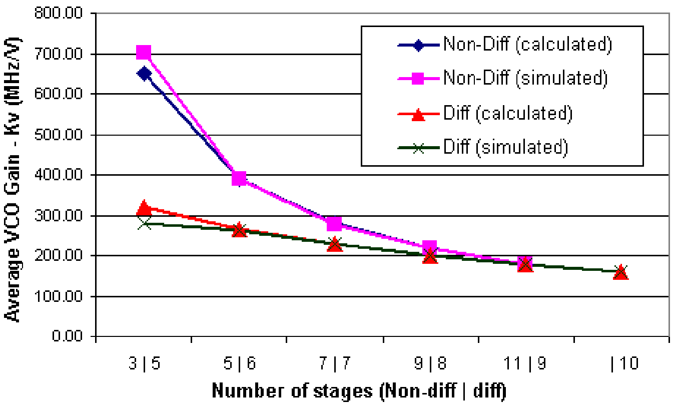


Figure 3–5: Simulated and calculated VCO gain.

The characterization of the resistor is performed next. The resistor was not built as a diffusion strip, but a transmission gate with both devices always enabled was used instead. The key assumptions here are that: (1) the drop across the resistor is small and (2) the voltage conditions that force the devices to work in the resistive region exist. For example, for the N device, V_G is equal to V_{dd} and V_D (which is also the voltage across the filter capacitor) is normally in the range below 2V, which is the expected normal operating range of the VCO and satisfies the condition $V_{DG} < V_{th}$. Similar analysis can be done for the P device. The transistor current in the resistive region is given by Eq. 3.11 [19] while Eq. 3.12 gives the expression for the resistance after assuming that V_{DS} is small and V_{GS} is on average equal to $V_{dd}/2$.

$$I_D = k(V_{DS})\mu_n C_{ox}\left(\frac{W}{L}\right)\left[(V_{GS} - V_{th})V_{DS} - \left(\frac{V_{DS}^2}{2}\right)\right] \qquad (3.11)$$

$$R_T = \frac{V_{DS}}{I_D} = \frac{2}{k(V_{DS})\mu_n C_{ox}(V_{dd} - 2V_T)}\left(\frac{L}{W}\right) \tag{3.12}$$

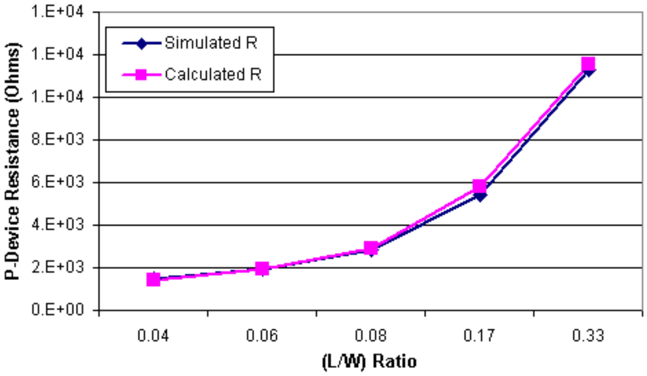


Figure 3–6: Simulated and calculated resistance value (P device).

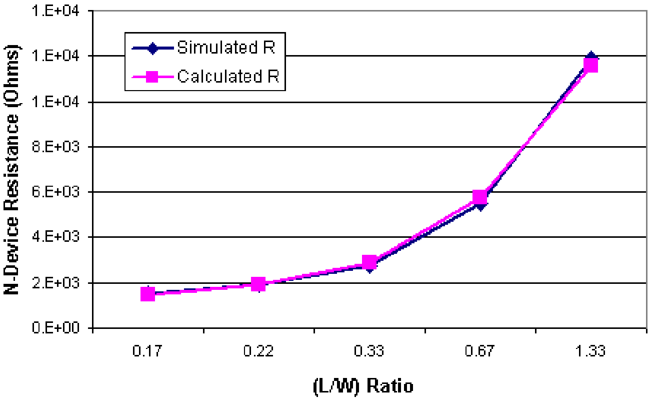


Figure 3–7: Simulated and calculated resistance value (N device).

Although Eq. 3.12 is still not a constant due to its dependence on *k(V_{DS})*, an average value for this parameter was calculated. *Figure 3–7* shows both the simulated and calculated values of *R* for the P device while *Figure 3–6* shows the same curves for the N device. The average deviation with respect to the simulated values was 3.77% and 3.2%, for N and P devices respectively. It must be noted that very large resistors can be built in a very small area using this technique.

The two remaining components, I_{CH} and C, are accurately described by equations already available. In particular, I_{CH} can be calculated with Eq. 3.7. On the other hand, since C is built via the gate capacitance of transistors with both source and drain grounded, then the total capacitance is calculated by multiplying C_{ox} times the total area of the gate.

3.3 PLL timing characterization

After validating a mechanism for extracting the PLL key parameters, the focus is now on modeling the PLL's performance behavior, which is determined by the time required by the PLL to re-acquire lock. If the input frequency is changed abruptly but its final value is within the lock range (see Eq. 3.5), then the time it takes the PLL to acquire lock again is given by Eq. 3.13 [47]. The PLL is in locked condition when no phase difference exists between input and output frequencies.

$$t_{lock} \approx \frac{k_f}{\xi\omega_n} \tag{3.13}$$

It was found that to reduce jitter to tolerable levels, the value of the constant k_f should be around 5, following the classic assumption that 99% of the final value of an RC-like system, is reached after a time equal to 5τ (i.e., five times the system's time constant) has passed. If the frequency step takes the input frequency beyond the lock range, then a capture (pull-in) process takes place first. The time required to bring the VCO frequency within the vicinity of the new input is given by Eq. 3.14 [44], where $f_{vco}(0)$ is the VCO frequency at the start of the process ($f_{vco}(0)=N_{old}.f_{ext}$).

$$t_{capture} = \left| f_{vco}(t_{capture}) - f_{vco}(0) \right| \frac{2C}{K_V I_{CH}} \tag{3.14}$$

For charge-pump PLLs, the capture process is linear, which contrasts with the nonlinear nature of the same process in classic realizations [47]. This holds as long as the parameters in the expression above do not change drastically during run-time. The frequency at the boundary of the capture and lock ranges is given by Eq. 3.15.

$$f(t_{capture}) = N_{new} f_{ext} - \frac{\Delta f_{lock}}{2} = N_{new} f_{ext} - \frac{4\pi\xi f_n}{2} \tag{3.15}$$

By using the stability condition (i.e., $\omega_n \leq \omega_{ext}/10$) in Eq. 3.14 and by replacing $f(t_{capture})$ as given in Eq. 3.15, the following expression is obtained.

$$t_{capture} = \left(\frac{2 f_{ext}(N_{new} - N_{old}) - 4\pi\xi(f_{ext}/10)}{N_{new} f_{ext}^2} \right) \frac{100}{(2\pi)^2} \tag{3.16}$$

After some manipulation and assuming that ξ is around the optimum value such that $4\pi\xi/10 \cong 1$, the total acquisition time is given in Eq. 3.17. In this equation, k_c and k_l are constants that perform a correction on the calculated times due to the discrete-time nature of the charge-pump PLL [45] which makes the times calculated without this correction slightly smaller than they actually are.

$$t_{acq} = t_{capture} + t_{lock} \approx k_c \left(\frac{2(N_{new} - N_{old}) - 1}{N_{new}} \right) \frac{100}{(2\pi)^2 f_{ext}} + k_l \frac{10 k_f}{2\pi\xi f_{ext}} \tag{3.17}$$

In particular, the values of k_c and k_l used were 1.7 and 2.7, respectively. Although, it was not possible to analytically derive these factors, it seems that they are mainly affected by the technology used as the values above worked well for all the design cases considered. If the PLL is initially off (i.e., not switching), the equation can also be used by simply making $N_{old} = 0$.

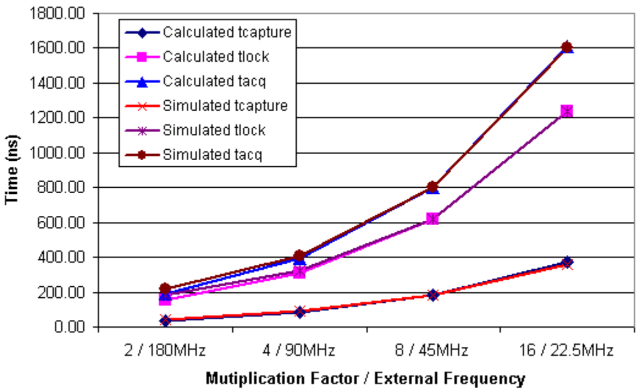


Figure 3–8: Simulated and estimated PLL response time for varying *N* (non-differential).

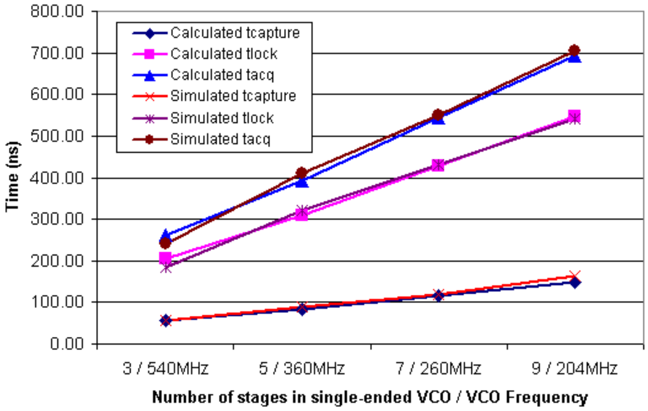


Figure 3–9: Simulated and estimated PLL response time for varying *n* (non-differential).

In order to verify these trends, a set of experiments was conducted for various PLL designs on a 0.35um technology, where the cases studied included varying *N* and K_v (by changing the number of stages, *n*). The obtained results are shown in *Figure 3–8* and *Figure 3–9* for the single-ended implementation and in *Figure 3–10* and *Figure 3–11* for the differential designs. For both cases, N_{old} = 0 and N_{new} is set to the value indicated (in the variable *N* case) while for the K_v varying case, N_{new} is always 4. Notice that the acquisition time increases with the multiplicative factor (i.e., external frequency decreases while VCO frequency remains constant) and decreases inversely with the VCO frequency (which depends on *n*). Effective ways to reduce the acquisition time are increasing f_{ext} and decreasing N (optimally, N =1), but as will be discussed later, performance improvements will have a negative effect on power consumption.

Table 3–1: Average and maximum error of estimated PLL response times.

Varying….	Error (%)	Single-Ended VCOs	Differential VCOs
Multiplication Factor	Maximum	13.35	7.55
	Average	4.55	3.61
VCO Number of Stages	Maximum	9.22	11.15
	Average	3.98	4.93

Table 3–1 summarizes the accuracy of the proposed equations. The error slightly increases for the cases where the PLL operates at higher frequencies or the input frequency increases, possibly caused by unexpected transients or glitching events. These appear when the device is forced to operate at a frequency where the linear behavior under which the above equations were formulated, no longer applies.

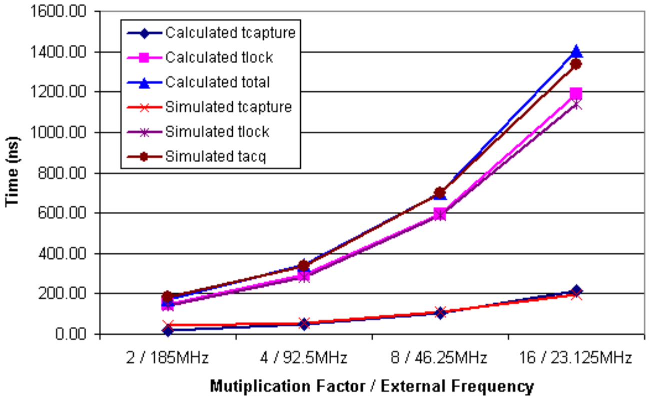


Figure 3–10: Simulated and estimated PLL response time for varying *N* (differential).

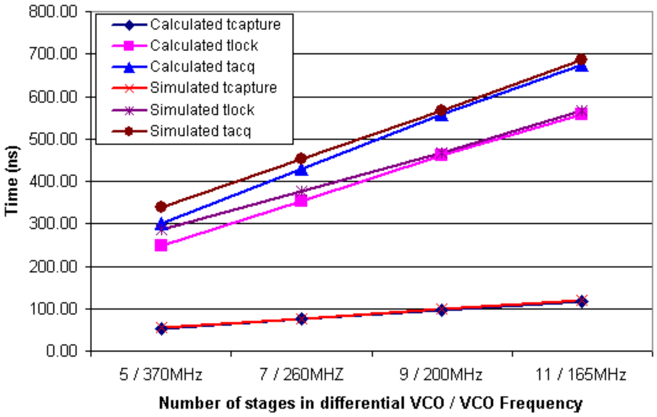


Figure 3–11: Simulated and estimated PLL response time for varying *n* (differential).

3.4 PLL power modeling

For the modeling of the PLL power consumption, the task has been divided in two. The power consumption of the PLL during lock is considered first and the power utilized during capture and lock-in processes are analyzed later.

3.4.1 PLL power consumption during lock

The locked condition can be considered a stable state. There are no drastic changes and most of the variables (V_c, f_{vco}, etc.) can be assumed constant. Then, the total power consumed can be estimated as the sum of the power required by each of the main elements. Expressions that define the effective capacitance of each PLL component (VCO, PFD, FDIV and charge pump) are presented briefly. The VCO is considered first. For a differential implementation, independent of the input values, there is always a branch that conducts current from the supply rail to ground while for a non-differential implementation, at any given time, just one cell is switching providing an excursion equal to the supply voltage. For n cells, the average current in each case is given by Eq. 3.18, where V_{sw} is the voltage swing that causes switching in the next cell (normally, $V_{sw} = kV_{dd}$, with k between 0 and 1), C_{cell} is the input capacitance of the next stage and t_d is the delay per cell.

$$I_{avg\_diff} = nI_{cell} = n\left[\frac{V_{sw}C_{cell}}{t_d}\right] \quad ; \quad I_{avg\_nodiff} = I_{cell} = \left[\frac{V_{dd}C_{cell}}{t_d}\right] \qquad (3.18)$$

Since the oscillation frequency is given by $f_{vco}=(2n.t_d)^{-1}$, and after some manipulation, the effective capacitances for each case are given in Eq. 3.19 and 3.20, where W_p and W_n, are the width of the N and P devices, respectively and C_{gate} and C_{drain} are the gate and diffusion capacitances of a minimum size transistor.

$$C_{vco-diff} = 2n^2kC_{cell} = 2n^2k\left[W_n(C_{gate} + C_{drain\_n}) + 2W_pC_{drain\_p}\right] \qquad (3.19)$$

$$C_{vco-nodiff} = 2nC_{cell} = \left[W_n(C_{gate} + C_{drain\_n}) + W_p(C_{gate} + C_{drain\_p})\right] \qquad (3.20)$$

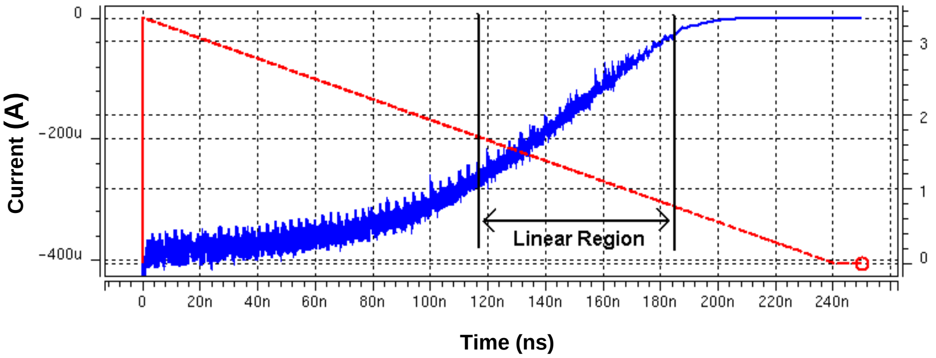


Figure 3–12: Supply current as function of the control voltage for a single-ended VCO.

Since initial validation attempts yielded relatively large error margins by operating the devices outside their dynamic linear range, the validation presented here ensures that the control voltage remains in the valid linear region, which is shown in

Figure 3–12 for the single-ended case. This region is narrower in differential designs due to their higher gain. This condition is also important for stable operation of the PLL. *Figure 3–13* shows average supply current readings given by SPICE, for different values of the control voltage. It can be seen that, for both cases, the measured supply currents follow the expected trends. In particular, the average (maximum) error was below 2%(5%) with respect to the constant behavior for the single-ended VCOs and below 0.10%(0.25%) with respect to the linear behavior of differential VCOs. The other remaining factor that influences the total effective capacitance is the technology impact, which was addressed earlier.

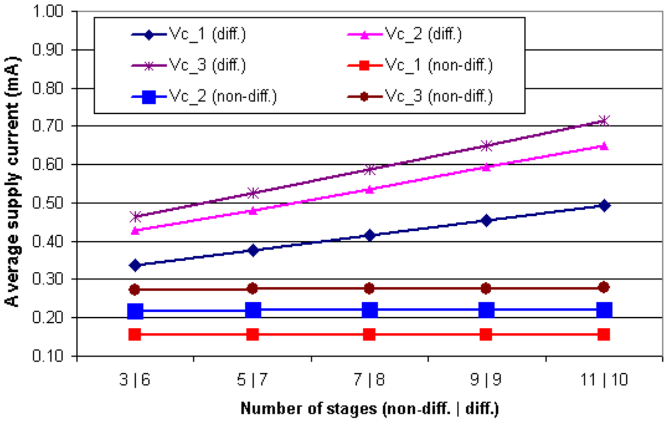


Figure 3–13: Supply current as function of the number of stages (*n*) for both VCO types and various values of the control voltage.

The widely popular Phase/Frequency Detector (PFD) design used [42, 30] is shown in *Figure 3–14*. The outputs (UP and DOWN) do not change if both input signals are in perfect phase and frequency synchronization. But if the VCO input is ahead of the REF input, either in phase or frequency, the DOWN signal switches at a rate proportional to the slowest input signal (REF in this case) and with a duty cycle proportional to the phase difference. The UP signal switches if the scenario reverses. After carefully analyzing the behavior of the circuit under all possible input combinations, it was found that all nodes in the design switch at a rate determined by the slowest input signal. This analysis permitted an estimation of the effective capacitance per transition of the reference input during both lock and acquisition as given by Eq. 3.21.

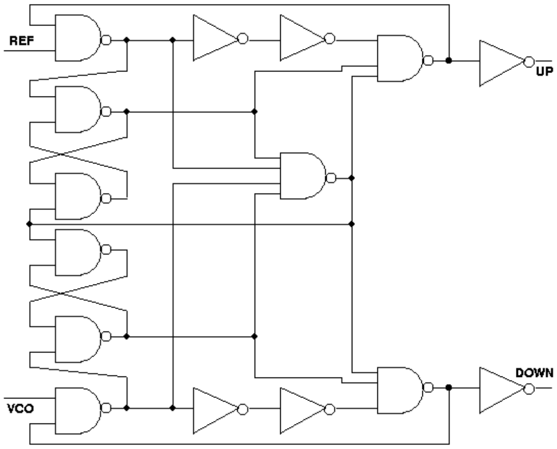


Figure 3–14: Schematic of the PFD used.

$$C_{pfd\_locked} = 108C_{gate} + 22C_{drain\_n} + 42C_{drain\_p}$$
$$C_{pfd\_acq} = 122C_{gate} + 30C_{drain\_n} + 64C_{drain\_p} \qquad (3.21)$$

By calculating an activity factor G_{pfd} as the ratio of $C_{pfd\text{-}locked}$ to the total capacitance calculated as the product of the total number of minimum-size transistors in the design (N_{tpfd}), and C_{tech} (i.e., $G_{pfd} = C_{pfd\text{-}locked} / N_{tpfd}.C_{tech}$), the effective capacitance can be expressed in a simpler way by Eq. 3.22. C_{tech} is the same expression given in Eq. 2.6, with the only exception that L_{local} is made equal to 10λ, since connections are only to neighboring cells, which makes the interconnect contribution virtually negligible for the technologies considered.

$$C_{eff-pfd} = N_{tpfd} G_{pfd} C_{tech} \qquad (3.22)$$

The calculations described above yielded G_{pfd_locked} = 0.74, G_{pfd_acq} = 0.83 and N_{tpfd} = 146. The average error of the estimated values with respect to SPICE data is presented in *Table 3–2*.

Table 3–2: PFD effective capacitance average error with respect to simulated values.

Tech. (um)	0.18	0.25	0.35	0.5	0.8	Average
PFD	8.11	15.89	7.20	3.97	4.43	7.92
FDIV	7.62	0.99	10.69	3.60	5.40	5.66
Average	7.86	8.44	8.94	3.78	4.19	6.79

The frequency divider (FDIV) is basically a ripple counter, built using four cascaded TSPC flip-flops [19] connected in toggle configuration plus a transmission-gate-based multiplexer and a few inverters. Since each of the successive flip-flops is running at half the frequency of the preceding one, the FDIV effective capacitance can be estimated using Eq. 3.23, where *N* is the multiplication factor that determines the switching activity of the output inverters.

$$C_{eff-fdiv} = \left[1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}\right]C_{ff} + \frac{2C_{inv}}{N} = 1.875C_{ff} + 2C_{tech} \tag{3.23}$$

After assuming that $C_{inv} = 4C_{tech}$, the flip-flop effective capacitance is given below, where the ½ factor captures the fact that this total capacitance is only charged once every two input frequency cycles, which was determined after carefully analyzing the flip-flop behavior.

$$C_{ff} = \tfrac{1}{2}\left\lfloor 20C_{gate} + 6C_{drain\_n} + 21C_{drain\_p} \right\rfloor \tag{3.24}$$

Following a similar analysis to the one presented for the PFD, a value of G_{fdiv} equal to 0.78 was estimated after setting N_{tfdiv} = 104, such that the effective capacitance can simply be expressed as given below.

$$C_{eff-fdiv} = N_{tfdiv}G_{fdiv}C_{tech} \tag{3.25}$$

The average error across technologies is presented in *Table 3–2*. Since the influence of the design style is difficult to generalize, it must be considered on case-by-case basis. It was a priority, however, to be as general as possible by modeling designs that are widely used. Signal glitching will also affect power estimates, particularly in deep-submicron technologies.

The last component is the charge pump, which is implemented as a single-ended switch-in-source design [49]. As long as the PLL is in locked condition, the outputs of the PFD do not change and no current flows to the filter capacitor. But the overhead of the bias circuitry present in the charge pump and VCO must be considered, so that the total PLL power can be expressed by Eq. 3.26. For the contribution of the bias circuitry, the only requirement is an estimate of the total current (I_{bias}), which in this case is proportional to I_{CH}.

$$\begin{gathered} P_{PLL} = (C_{PFD} + C_{VCO} + C_{FDIV})V_{dd}^2 f_{vco} + P_{bias} \\ P_{PLL} = [N_{tpfd}G_{pfd} + N(2n(6) + N_{tfdiv}G_{fdiv})]C_{tech}V_{dd}^2 f_{ext} + I_{bias}V_{dd} \end{gathered} \tag{3.26}$$

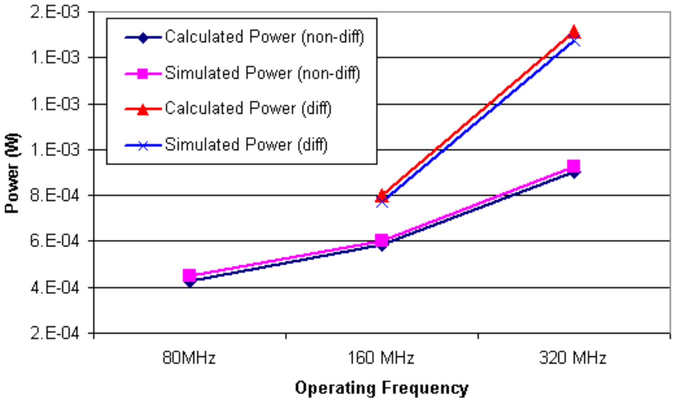


Figure 3–15: Simulated and calculated PLL power under locked state for both PLL types.

Preliminary experiments were performed and *Figure 3–15* shows the calculated and simulated total PLL power values of the base PLL design (which included a 5-stage non-differential VCO) running at various frequencies. Additional simulations were performed with a design that employed a 6-stage differential VCO and the obtained values are also captured by *Figure 3–15*. The average (maximum) deviation from the simulated results was 3.5% (4.8%) and 3.4% (4%) for the single-ended and differential cases, respectively.

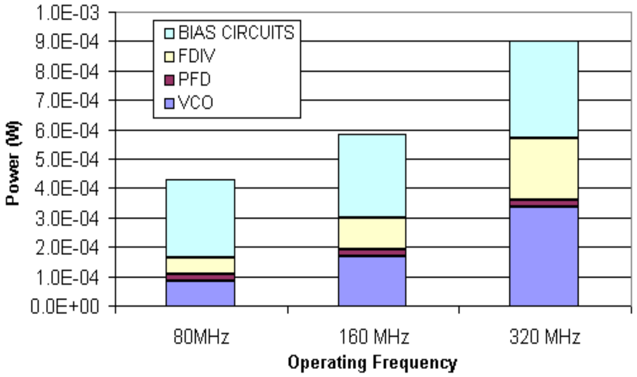


Figure 3–16: Calculated PLL power breakdown for a non-differential PLL.

Figure 3–16 and *Figure 3–17* show the power breakdown across all PLL components for single-ended and differential implementations, respectively. While the contribution of the VCO, the FDIV and the bias circuits increase with the operating frequency, the PFD share remains constant. When comparing the power breakdown, the VCO contribution becomes dominant for differential implementations (given the quadratic dependence on the number of stages), while for the non-differential case all components (except the PFD) take comparable shares of the total power budget. This also implies larger power consumption for the same operating frequency, which is the price that must be paid for better noise immunity in differential designs. However, this does not restrict non-differential VCOs from being used in commercial products [42].

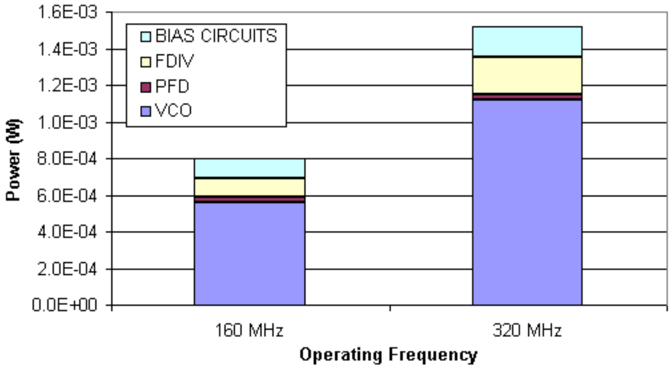


Figure 3–17: Calculated PLL power breakdown for a differential PLL.

Further experiments were performed under two clearly distinct circumstances: first, the multiplication factor was varied (i.e., possible values were 2, 4, 8 and 16) and second, the number of VCO stages was modified from 3 to 9 and 5 to 11 (in steps of 2) for single-ended and differential PLLs, respectively. These operating parameters are identical to those employed in the experiments in Section 3.3. *Figure 3–18* and *Figure 3–19* show the simulated and calculated data for the two cases mentioned above, while *Table 3–3* summarizes the results.

Table 3–3: Average and maximum error of estimated PLL lock power with respect to values obtained from simulation.

Varying….	Error (%)	Single-ended VCOs	Differential VCOs
Multiplication Factor	Maximum	8.53	6.41
	Average	4.25	4.56
VCO Number of Stages	Maximum	6.98	9.40
	Average	4.29	4.93

From *Figure 3–18* and *Figure 3–19*, it is observed that a differential PLL consumes 35.7% more power (on average) than its single-ended counterpart for the same operating frequency (an adjustment was made to compensate for the slightly higher operating frequency of the differential design). This difference relies completely on the type of VCO, as all other components remain unchanged from one design to the other.

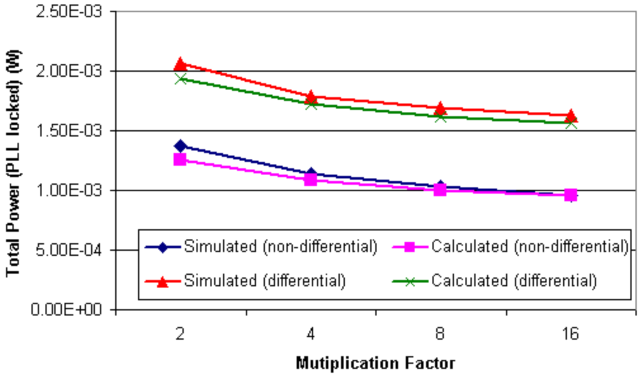


Figure 3–18: Simulated and calculated total power under locked condition for non-differential and differential PLLs for varying *N*.

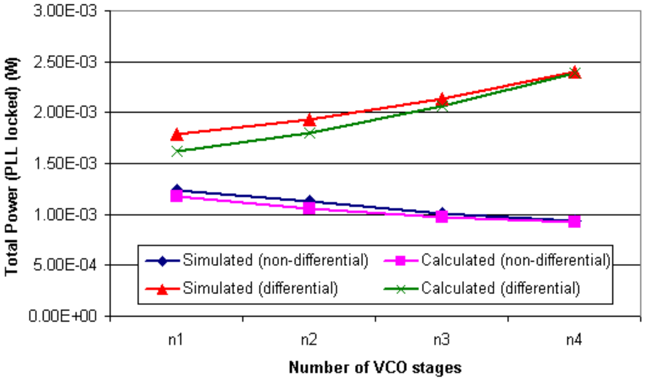


Figure 3–19: Simulated and calculated total power under locked condition for non-differential and differential PLLs for varying *n*.

Figure 3–18 also shows an expected trend in power as *N* increases, which is caused by a reduction in the PFD power. For the non-differential implementations, a multiplicative factor of 4 reduces the total power by 17.3% with respect to the case when N = 2 (for N equal to 8 and 16, the reductions are 24.6% and 29.8%, respectively). For the differential implementations, the reductions with respect to N = 2 are 13.8%, 18.2% and 21.6% for N equal to 4, 8 and 16, respectively. The price to pay, however, is a sluggish response time as discussed in the previous section. Note the contradictory trend observed between the two types of PLLs when the number of stages in the VCO increases, as shown in *Figure 3–19*. This is caused by the quadratic dependence of the differential VCO power with respect to the number of stages (as formulated earlier), which dominates the linear reduction associated with the change in the operating frequency. For the single-ended case, the frequency decline dominates the increase in the VCO effective capacitance and it also affects all other PLL components.

Now, the leakage power behavior of the PLL is discussed. The VCO is not expected to contribute strongly to the total PLL leakage, as the number of transistors on it is usually small. Simulations for the 0.25um case revealed that leakage power is more than 1000 times smaller than the dynamic power measurements. For the FDIV and the PFD, although their transistor counts are larger, the relative difference is even greater than the order of magnitude given above. A closer examination of the designs reveals a lot of stacking present, which reduces leakage by a factor close to 10. This phenomenon is called the “stacking effect”. For now, all leakage contributions to the PLL total power will be disregarded, at least for the technologies considered.

3.4.2 PLL Power consumption during acquisition

For the estimation of power in this case, the equations that describe the PLL dynamic behavior must be used. As mentioned earlier, when an instantaneous change of the input frequency (or the dividing factor) occurs, it is possible that in addition to a lock process, a capture process also occurs if the frequency step is beyond the lock range. From Section 3.3, the time required for the PLL to lock to the new frequency can be expressed as $t_{acq} = t_{capture} + t_{lock}$. The length of the capture time is such that the final voltage brings the VCO frequency within the lock range and a *linear* lock process can then take place. Here, *linear* refers to the behavior that can be described by the equations in Section 3.1. For the estimation of the power consumption during the capture process, it is assumed that the VCO frequency follows the variation on the control voltage and that the PFD switching activity increases with respect to the locked condition but remains independent of the VCO frequency variation. Eq. 3.27 gives the expression for the average PLL power consumption during capture, where for the frequency dependent components, an average frequency ($f_{capture\_ave}$) is used which is basically the mean value of the start and stop frequencies during the capture process.

$$P_{PLL\_capture} = \left(C_{PFD} + C_{VCO} + C_{FDIV}\right)V_{dd}^{2} f_{capture\_ave} + P_{bias} + P_{ch\_pump} \tag{3.27}$$

Since there is a portion of the bias current that increases with frequency (i.e., the VCO portion of the bias circuitry), its effective load is also adjusted by calculating an average current, which is the mean value of the starting and the ending currents, as given by Eq. 3.28.

$$I_{bias\_ave} = \frac{I_{start} + I_{start}}{2} = k\frac{\left[\left(\frac{f_{start}}{K_V}\right) - V_T\right] + \left[\left(\frac{f_{stop}}{K_V}\right) - V_T\right]}{2} \tag{3.28}$$

For the charge pump, both the energy delivered to the capacitor and the energy dissipated in the resistor are considered. These are estimated as a function of the total variation of the capacitor voltage *(ΔV)* during the capture process (i.e., the total energy delivered) as in Eq. 3.29.

$$P_{CH-PUMP} = \left(\frac{\Delta V^{2} C}{t_{capture}}\right) + \left(\frac{\Delta V \cdot C}{t_{capture}}\right)^{2} R = \left(\frac{\Delta V \cdot I_{CH}}{2}\right) + \left(\frac{I_{CH}^{2} \cdot R}{4}\right) \tag{3.29}$$

Now, after the VCO frequency has been brought within the lock range, or in the case that the instantaneous change is not beyond the lock range, the PLL initiates a lock process and the power consumed is estimated as in Eq. 3.30, where the average frequency was estimated using the definition of average value in Eq. 3.31.

$$P_{PLL\_lock} = \left(C_{PFD} + C_{VCO} + C_{FDIV}\right)V_{dd}^2 f_{lock\_ave} + P_{bias} + P_{ch\_pump} \tag{3.30}$$

$$f_{lock\_ave} = \frac{1}{t_{lock}} \int_0^{t_{lock}} f_o(t)dt = A_{step}N\left[1 - \frac{1}{\omega_d t_{lock}} e^{-bt_{lock}} \sin(\omega_d t_{lock})\right] \tag{3.31}$$

Eq. 3.31 can also be obtained by calculating the RMS value while disregarding second order exponential terms (which are very small) and then applying the approximation given by Eq. 3.32.

$$\sqrt{1-x} \approx 1-\left(x/2\right) \qquad \text{for} \quad \text{small } x \tag{3.32}$$

It was found, however, that without applying the above approximation, the values obtained using each equation were within 1% of each other. The calculated average was within 1% of the final target frequency, which means that all components, besides the PFD and the charge pump, contribute similarly as they do while the PLL is locked to that final frequency. The PFD switching activity is assumed to be G_{pfd_acq}. To determine the charge pump contribution, the time intervals shown in *Figure 3–20*, when energy is actually being delivered to the filter (i.e., by charging the filter capacitor), must be found. During the intervals in which the voltage decreases, no power is drawn from the power supply as the charge is removed from the capacitor and given a path to ground.

In order to determine these times, the derivative of $f_o(t)$ is calculated and then equaled to zero. These times, as given by Eq. 3.33, represent the points where the function reaches either its maximum or minimum values, for n = 1,2,3, etc.

$$t_n = \frac{1}{\omega_d}\left[\tan^{-1}\left(\frac{-2\omega_d b}{\omega_d^2 - b^2}\right) + n\pi\right] \tag{3.33}$$

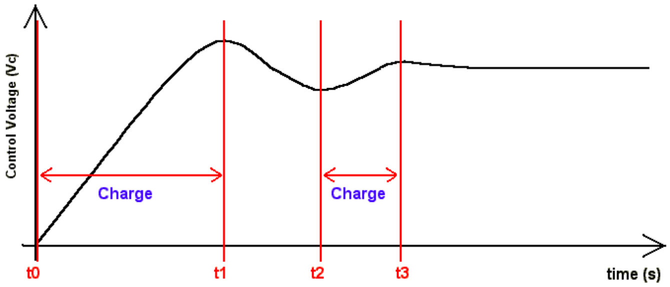


Figure 3–20: Definition of the capacitor charging intervals.

The changes in the capacitor voltage are estimated as in Eq. 3.34, where *n* = 1,3,5, etc., and the frequency at t_0 ($f_o(t_0)$) is the frequency at the start of the lock process. Using Eq. 3.34, the charge-pump power is given by Eq. 3.35.

$$\Delta V_n = \frac{1}{K_V}\left[f_o(t_n) - f_o(t_{n-1})\right] \tag{3.34}$$

$$P_{CH-PUMP_n} = \left(\frac{\Delta V_n^2 C}{t_n - t_{n-1}}\right) + \left(\frac{\Delta V_n \cdot C}{t_n - t_{n-1}}\right)^2 R \tag{3.35}$$

Since the lock time is the reference time frame for these calculations, then the total charge pump power is rewritten and now estimated using Eq. 3.36.

$$P_{CH-PUMP} = \frac{1}{t_{lock}} \sum_{n=1} \left(\Delta V_n^2 C + (\Delta V_n \cdot C)^2 R\right) \tag{3.36}$$

Considering both capture and lock processes, the total PLL power consumption during lock acquisition is given below.

$$P_{PLL\_acq} = \frac{1}{t_{acq}}\left(P_{PLL\_lock-in} t_{lock} + P_{PLL\_capture} t_{capture}\right) \tag{3.37}$$

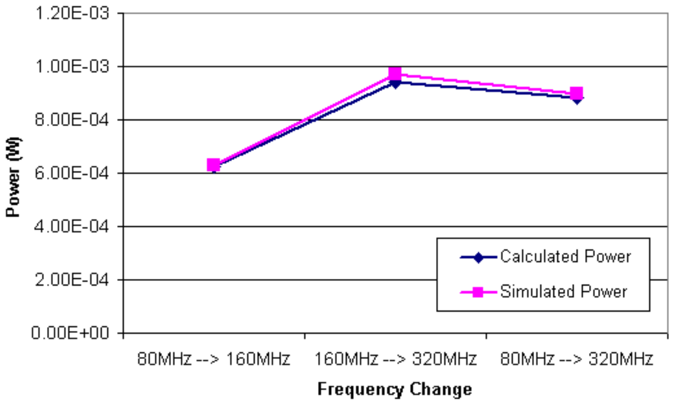


Figure 3–21: Simulated and calculated PLL power during acquisition for a non-differential PLL.

Figure 3–21 presents initial results, including estimates using the equations derived above, for three frequency steps (i.e., changes in *N*) applied to the base non-differential design. The average (maximum) deviation with respect to SPICE readings was 1.7% (2.8%). Additional experiments with the differential design resulted in an average (maximum) error margin of 0.7% (1.11%). The fact that power is larger for the transition from 160 to 320MHz than for that of 80 to 320MHz can be misleading. But, the energy readings, however, are consistent and show a higher value for the larger step.

Table 3–4: Average and maximum error of estimated PLL acquisition power with respect to SPICE simulation values.

Varying….	%	Single-Ended VCOs	Differential VCOs
Multiplication Factor	Maximum	4.53	4.36
	Average	1.89	2.39
VCO Number of Stages	Maximum	4.21	8.32
	Average	2.11	3.41

Further experiments considered the two cases presented in the previous section (i.e., varying the multiplication factor and varying the number of VCO stages). The simulation framework remains unchanged. *Figure 3–22* and *Figure 3–23* show the simulated and calculated data while *Table 3–4* presents average and maximum deviation for all cases.

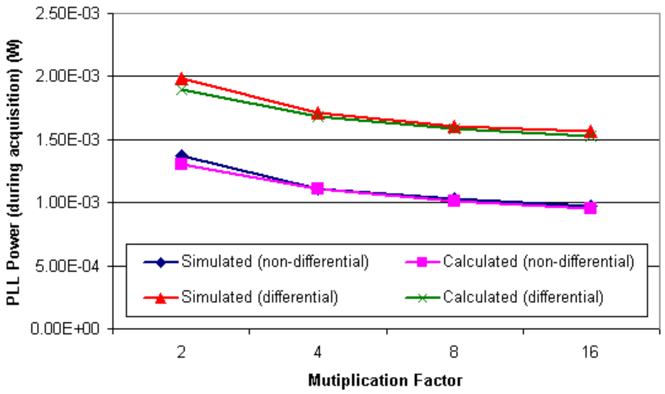


Figure 3–22: Simulated and calculated total power during lock acquisition for non-differential and differential PLLs for varying *N*.

As expected, differential PLLs present a larger lock-acquisition power than single-ended designs, averaging an increase of 35.7%, which is very similar to the difference found in the locked state case. *Figure 3–22* also shows the already noted trend of reduced power as *N* increases and the percentage reductions are also close to those calculated earlier for the locked state of the PLL. What becomes important now is to calculate how the lock-acquisition power and power consumed during lock (at the target frequency) compare to each other. Note that for all non-differential designs, the power during acquisition is larger than the power consumed under lock, by 2%, on average. For non-differential designs, the situation reverses, as now the lock power is larger than the acquisition power by a margin of 2.1%, on average. This may be caused by the dominant contribution of the differential VCO to the total PLL power. Note that for these experiments, the PLL starts from a non-switching condition (N_{old} = 0), but for those presented in *Figure 3–21*, N_{old} is always different from zero. In those cases, acquisition power is larger than lock power by average margins of 2.1% and 2.53%, for non-differential and differential designs, respectively.

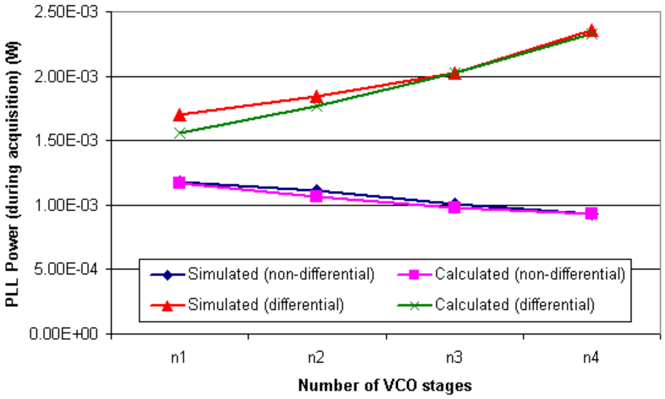


Figure 3–23: Simulated and calculated total power during lock acquisition for non-differential and differential PLLs for varying *n*.

In *Figure 3–23* the contradictory trend between the two types of PLLs when the number of stages in the VCO increases is observed again. But, when comparing acquisition and lock power, it was found that, for both implementations, the power during lock is larger (1.78% and 4.09% for single-ended and differential implementations, respectively) than the acquisition power. This seems to be a consequence of the fact that the PFD contribution to total power is now constant since *N* remains unchanged. Appendix D discusses additional PLL design issues and the applicability of the PLL as voltage regulator.

Chapter 4

CLOCK ENERGY OPTIMIZATIONS

Once it has been shown that the model is reasonably accurate, it can be used to evaluate the impact of the different loads present in the clock network as their organization or architecture is changed. In other words, since the model is parameterizable, various configurations can be studied and assessed, so that the total clock power is minimized. Also, the model can be a very useful tool in showing behaviors that might not be, in general, easy to detect. To illustrate this, two main cases are considered next: the impact of design-time choices and run-time optimizations. For the latter, not only techniques that reduce clock energy are considered but also those that reduce energy of other system components that are tightly coupled to the clock sub-system.

4.1 Design-time optimizations

The first experiment is set to determine where the energy hotspots in the clock sub-system are, and to do this, three different architectures were considered: a superscalar RISC processor, a DSP (which is basically a VLIW machine) and a Network Processor (NP). The relevant architectural parameters of each design, given in *Table 4–1*, were substituted in the model, so that the clock capacitance of each component is determined. *Figure 4–1* captures how the different loads contribute across the three designs. It was assumed that all designs were implemented in a 0.18um technology and used an H-tree topology. Structures particular to each design, such as branch history tables, issue windows, L2 caches and similar structures, are not captured.

Table 4–1: Three processor configurations.

Feature	RISC	DSP	NP
Address Bus Width	32	21	40
Data Bus Width	64	32	64
GP Registers (GPRs)	32	64	32
FP Registers (FPRs)	32	NO	32
Number of Pipelines	5	8	6
- Number of stages	5	6	10
L1 D-Cache (I-Cache)	32 KB	16 KB	32KB
- Mapping (D / I)	8-way	2-w /dir	4-way
- Block size - bytes	32	32	32
D-TLB (I-TLB) - entries	128	NO	64
- Mapping	2-way	--	Fully
- Virtual address bits	52	--	40

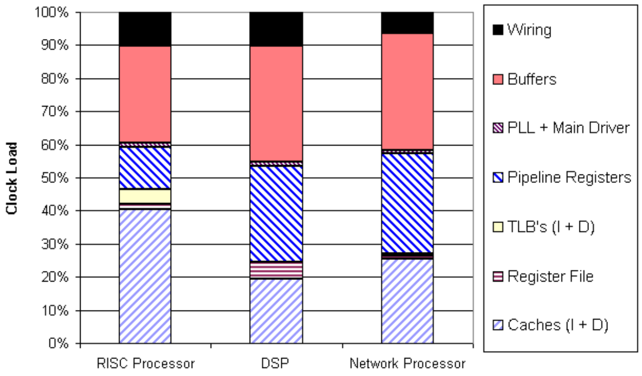


Figure 4–1: Total clock capacitance breakdown for three processor configurations.

There are many interesting insights given by this very simple experiment. The first thing to notice is that in all cases, although the configurations are different, the main contributors to the clock power budget are the buffers, the caches and the pipeline registers, with the percentages varying from one architecture to the other. All other loads are almost negligible (except for the TLB in the RISC and the register file in the DSP) with respect to these main contributors, but it is important to keep in mind that the wiring component (i.e., the distribution network) can play a major role, especially for grid implementations in more aggressive technologies. This experiment helps in focusing attention on the loads that are causing the largest power consumption, so that power-reduction strategies and optimizations that provide the most significant savings can be developed. Possible optimizations in the memory structures' load and also in the buffer design style are considered next.

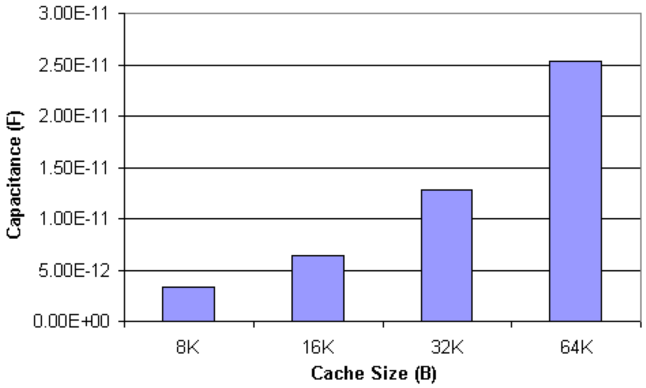


Figure 4–2: Influence of cache size on the cache clock load.

First, the effects on the cache precharge component for different cache sizes are presented in *Figure 4–2*. As expected, the load on the clock network per cache access increases almost linearly with size. It is important to recognize that this increase might be worthwhile in exchange for a reduction of the cache's miss ratio, but this must be evaluated at runtime (i.e., using an architectural-level simulation tool). The impact on the cache precharge component for different cache associativities follows the assumption that the direct-mapped case provides the least load per access. When compared to the direct-mapped case, the precharge load per access increases by just 1%, 2% and 4% for 2-way, 4-way and 8-way implementations, respectively. This is caused by the decrease in the number of rows affected by the scaling factor of the precharge transistor and the increment in the tag size. This behavior, however, would give some freedom to the architect, as now he/she can basically select the associativity of the cache based on the cache's own power consumption and system performance requirements.

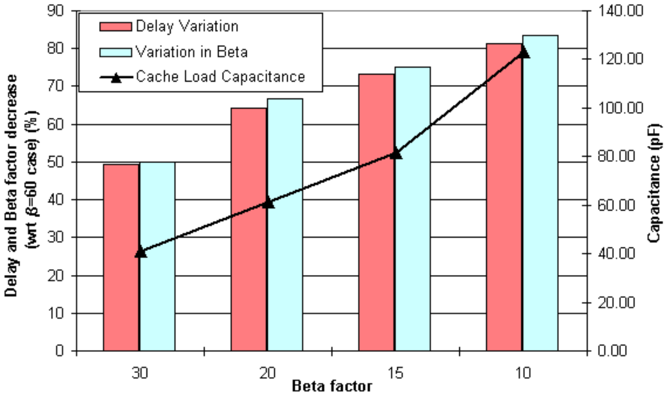


Figure 4–3: Influence of (β) beta factor on cache clock load and on cache performance.

Now, the effect of the β factor on memory-dependent clock energy will be considered and it will be shown how the model provides a means to evaluate what the real performance-power trade-offs incurred by choosing a given sizing factor are. *Figure 4–3* shows the percentage decrease in delay with respect to a base implementation using β equal to 60, as measured from SPICE signal traces. Clearly, a smaller β means a reduction in the precharge time at the expense of increased clock load as calculated using the model (see line plot). However, the actual variation in β, which is presented by the light bars, can be used to estimate the changes in delay with respect to the case β = 60 (i.e., if β = 30, there is a reduction of 50%). The plot shows that the β factors ratio provides a good estimate of the performance change since the maximum error of this metric with respect to the actual delay measurements is 2.5%.

As another example of the usefulness of the model, *Figure 4–4* shows the impact of different values of the sizing factor on the buffer power. In this experiment, the number of inverters in the chain must be changed as *u* changes, as illustrated in the same figure, such that the clock signal is delivered to the load without incurring a very high performance penalty. Note that two conflicting trends contribute to the uncharacteristic behavior of the buffer dynamic power. As *u* increases, which implies larger devices for any given stage, the number of stages decreases, which means that a lower number of contributions are added, although they are actually larger. One would expect a parabolic behavior of the dynamic power curve, which can actually be observed between the values obtained for *u* = 2.33 and 3.66. Beyond these two extremes, the trend changes as the variation in the number of stages is no longer linear. This is reasonable given that the number of required stages is calculated using the ratio of two non-linear expressions (see Eq. 2.3).

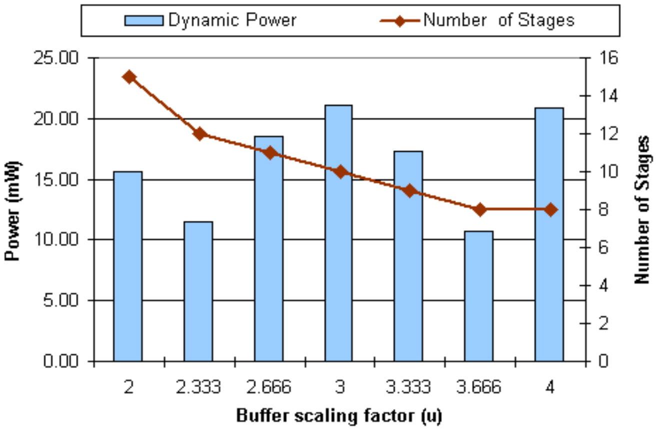


Figure 4–4: Buffer dynamic power and required number of stages as function of the scaling factor (*u*).

The interesting result about this simple experiment is that, although a value between 2.66 and 3 provides the optimum behavior in terms of timing (which is expected and was confirmed by SPICE simulations, as shown by *Figure 4–5*), the optimum value for power consumption is found at a different value of *u*. In particular, from *Figure 4–4*, it can be seen that the least power is consumed for values of 2.33 and 3.66. The increase in delay incurred by using *u* = 3.66 is 14.8% with respect to the case where *u* is 3, which is an acceptable overhead for a reduction of 49% in buffer power. For the case of *u* = 2.33, a smaller penalty is incurred but the savings are also proportionally less.

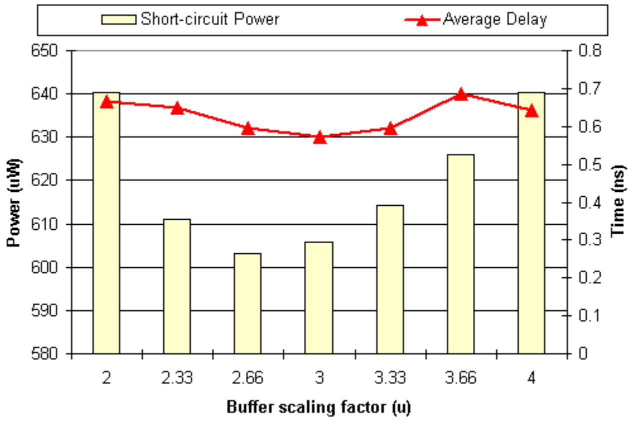


Figure 4–5: Short-circuit power estimates and delay numbers as function of the scaling factor (*u*).

The figure also shows an estimate of the behavior of the short-circuit power, which was discussed in Section 2.1.1. The short-circuit power variation across the possible values of *u* is confined within the percentage variation of the *u* factor (about 6%). Since this translates into less than 1% of the overall dynamic energy, the model estimates are virtually unaffected. A decision on whether a different value of *u* should be used depends heavily on the influence that this change would have on the timing characteristics of the driver. One possibility is to use *u* = 3.66 in drivers that control sections where larger clock skews are allowed.

Now, besides architecture and spatial distribution, process technology is another important factor that affects the clock energy behavior. As interconnect capacitance will increase significantly for deep submicron technologies, it is worthwhile to be aware of the implications that this will bring to the complete clock energy picture. *Figure 4–6* shows the expected clock energy distribution when the single-issue, five-stage datapath design of the power simulator described in [25] is scaled down from 0.35um to a 0.18um technology. The first thing to notice is the increase in the relative contributions of the precharge load, the wiring and the main driver. A significant portion of these components is basically interconnect capacitance, which increases for smaller technologies mostly due to the reduction of distance between wires. The datapath (pipeline registers) presents a mixed load of transistors and interconnect capacitance and at the end, its contribution remains almost constant. The power from buffers reduces since they are basically transistors, which scale down with technology.

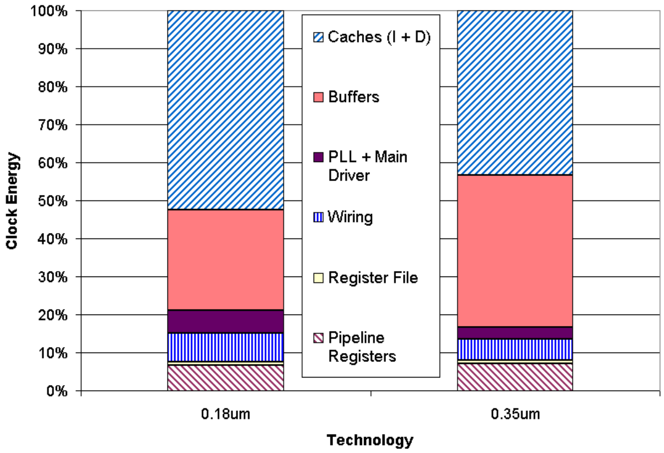


Figure 4–6: Technology dependence of the clock energy distribution.

4.2 Run-time clock power reduction schemes

In order to perform the experiments presented here, the clock model was incorporated into an existing cycle-accurate datapath and memory energy simulator [25]. This is an important extension as the influence of the clock network on the overall system energy can now be accounted for. The architecture used in [25] is a single-issue 5-stage pipeline datapath, which describes one of the most common architectures used in embedded processors and runs the integer subset of SimpleScalar's instruction set architecture [35]. The estimates from this simulator were shown to be within a 10% error margin from circuit-level simulation.

Figure 4–7 shows a schematic of the clock network of the target architecture. It must be noted that all circuitry on the design is built using static CMOS devices (no dynamic circuits are used), so that the clock is only needed in those places shown in the figure. This is the trend in embedded systems where the focus is on power and a similar argument applies to SOC designs, where the complexity of the system limits power consumption. It can be seen in the figure that the impact of gating, at different levels and for different units, has been modeled. The gating of the caches precharge load during a cache miss is mandatory, especially for a low power system (mid-level gating). The datapath gating is not shown, but it is clear that the system cannot proceed until the cache fill requests have been serviced.

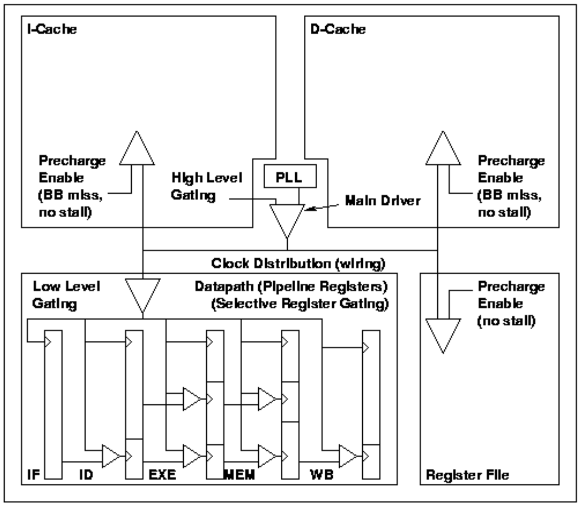

I-Cache
D-Cache
Precharge Enable (BB miss, no stall)
High Level Gating
PLL
Main Driver
Precharge Enable (BB miss, no stall)
Clock Distribution (wiring)
Low Level Gating
Datapath (Pipeline Registers) (Selective Register Gating)
Precharge Enable (no stall)
IF
ID
EXE
MEM
WB
Register File

Figure 4–7: The energy simulator clock network topology.

Two additional gating levels were added. First, additional clock power savings are sought through selective gating of the pipeline registers, which is called low-level gating since it is implemented at the lowest end of the distribution network. In this scheme, selective portions of a register are disabled depending on whether the information they hold is required by the next stage. This, however, should not be confused with the datapath gating mentioned earlier. The overhead incurred by implementing this technique is minimal [36] and since the impact on datapath energy of such a scheme has already been evaluated, the influence of such finer grain gating on clock load is now assessed.

Secondly, support for gating of the whole distribution network was added at the main driver (high-level gating). This eliminates unnecessary switching of the main driver and the top-level clock wiring during cache misses. Note that this will work only for an architecture like the one used here, where a cache miss translates into a complete stall of the pipeline. This level of gating may incur some overhead in a single-processor embedded system, but in SOCs, support for this may already be available, especially if each processor core is turned ON/OFF in response to processing requirements. Shutting down the PLL would be costly in terms of performance due to the time required to regain its locked condition, so this is not done. Thus, by simply enabling the gating signal at the main driver, the processor core is allowed to restart instantly.

Table 4–2: The details of the benchmarks used.

Benchmark	Source	Input Size (KB)	Execution Cycles
adi	Livermore	241	8,251,639
apsi	Perfect Club	1,605	53,472,741
bmcm	Perfect Club	126	108,042,830
btrix	Spec95	4,312	54,197,641
eflux	Perfect Club	297	18,354,453
mxm	Spec95	120	61,982,751
tomcat	Spec95	119	7,029,092
tsf	Perfect Club	204	21,168,771
vpenta	Spec95	114	3,329,809
wss	Perfect Club	125	107,193,410

To perform the experiments presented next, the array-dominated benchmarks listed in *Table 4–2* were executed in the energy simulator. Note that many applications in video and signal processing domains are array-dominated and both embedded and SOC designs have been widely used in portable communication devices that execute frequently this type of code. The base configuration (architecture) had 8KB, 2-way set associative instruction and data caches with a latency L_c for every cache access miss. The mentioned mid-level gating is also implemented in the base system.

4.2.1 Cache architecture impact on clock energy

A considerable portion of the clock energy is expended in the memory subsystem. Thus, it is important to evaluate the impact that different memory architectures have on the energy consumed by the clock network. If static evaluation is performed using the model, a trend of increased cache load with increasing associativity is observed. It was also found that the effect of cache size on the capacitance per access follows the intuitive linear relationship (i.e., the cache precharge load doubles as the size doubles). However, these static calculations cannot be assumed to correlate completely with the behavior during runtime. Thus, it is necessary to assess the effect of cache performance on the clock energy's dynamic behavior. *Figure 4–8* shows the clock energy variation while *Figure 4–9* presents the clock energy consumed per cycle (EPC) for each benchmark, with respect to changes in cache associativity.

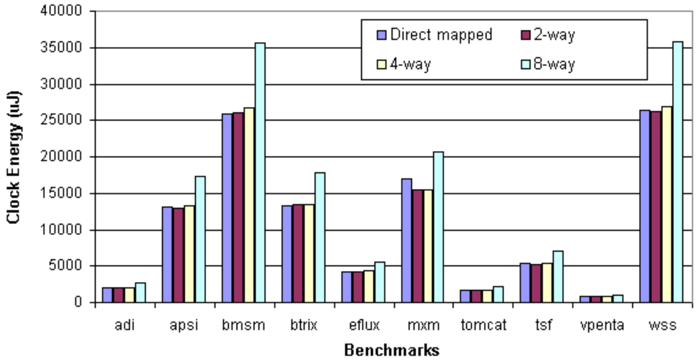


Figure 4–8: Clock energy as function of cache associativity for the benchmarks used.

The tradeoffs in clock energy with varying associativity (cache size is always 8KB) are the increase in precharge load per access, as mentioned before, and the reduction in the total number of clock cycles due to the potential improvement in cache hit rates with the increased associativity. Some benchmarks follow the trend presented by the static evaluation but others exhibit lower energy consumption for 2-way or 4-way than for the direct-mapped case. This clearly brings cache performance into the clock energy picture, as indicated before. The energy consumption for the 8-way case is much higher than the others, which leads to the conclusion that, in this case, the combination of an increased precharge load and a poor hit ratio seriously harms the clock energy behavior.

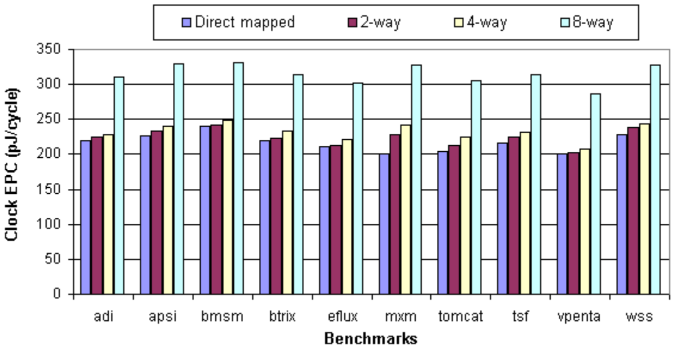


Figure 4–9: Clock EPC as function of cache associativity for the benchmarks used.

This behavior contrasts appreciably with the trend observed in the cache energy, where the cache energy increases monotonically with increasing associativity, for all benchmarks. This is expected since the number of accesses to the cache is constant, independent of its size. Thus, by observing the clock energy behavior, the cache's miss performance can be assessed.

The results in *Table 4–3* show that the on-chip savings are driven by clock energy optimization, with the maximum average savings being reached with a 2-way configuration. The maximum savings on cache energy occur with the direct mapped case but they are not enough to overcome the increase on clock energy due to the unnecessary clock cycles caused by a larger miss rate. The energy per cycle (EPC) increases with associativity for all cases but with the actual percentage variation being a function of each benchmark. It must be recalled that the caches are responsible for up to 60% of the total power consumed in an embedded system. In SOCs, where the number of processing cores is large, more dramatic savings can be obtained at the chip level.

Table 4–3: Average energy change calculated with respect to the 2-way case.

Variation (%)	Direct	4-Way	8-Way
Clock	0.74	1.52	33.49
Caches	-0.10	0.50	1.42
Total On-Chip	0.24	0.45	5.10

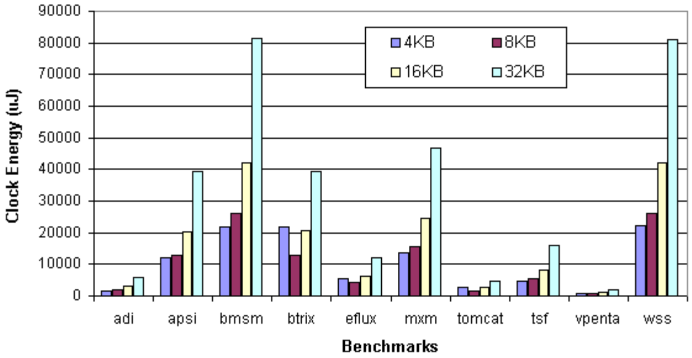


Figure 4–10: Clock energy as function of cache size for the benchmarks used.

Now, the effects of variations in cache size are evaluated, while fixing the associativity to 2-way. *Figure 4–10* shows that only when the cache storage is large enough (i.e., 16KB and above), a linear increase in clock energy is observed. For smaller caches, miss performance becomes dominant. There are benchmarks like *btrix* or *tomcat*, for instance, where a 4KB cache consumes as much clock energy as a 16KB cache. This contrasts again with the measured behavior of the cache energy, which increases when size increases (i.e., doubling the cache size doubles the cache energy). A comparison, including total on-chip energy is presented in *Table 4–4*. In contrast to the energy figure, the EPC metric in *Figure 4–11* shows a behavior that is dominated by the impact of the array size instead of the miss performance. The linear increase with size can be observed except for the 4KB implementation.

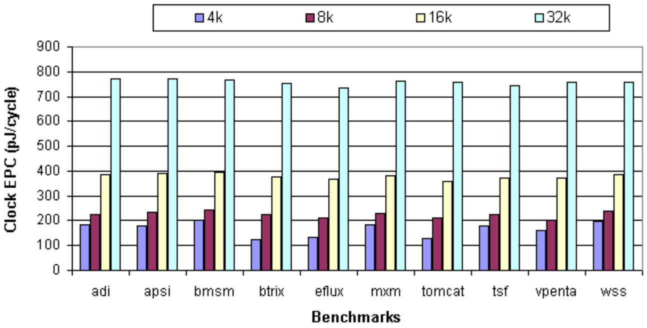


Figure 4–11: Clock EPC as function of cache size for the various benchmarks used.

Note that when reducing the cache size to 4KB, the predicted halving in clock energy by the static model is overcome by the miss rate increase and the clock energy actually increases by 5.90% (on average), with respect to the base 8KB-cache system. For caches larger than 8KB, both cache and clock energy increase monotonically but at different rates. This clearly reaffirms the importance of these types of experiments at the design stage, so that architectural decisions can be done with the understanding of the requirements of the application software. From the average clock energy numbers given in *Table 4–4*, an 8KB cache constitutes the optimum size for the application framework used after considering performance as well. It was also observed that the total on-chip energy increases with increasing cache size, with no exceptions but contradictory optimum points are found again for clock (8KB) and cache (4KB) energy. Here, however, the off-chip energy will ultimately determine the optimum choice.

Table 4–4: Average energy change calculated with respect to the 8KB case.

Variation (%)	4KB	16KB	32KB
Clock	5.90	54.40	193.73
Caches	-49.45	98.87	205.83
Total On-Chip	-32.10	77.35	235.72

Next, two classic memory energy optimizations are evaluated from the clock energy perspective: sub-banking and block buffering [23]. In sub-banking, the effective pre-charge capacitance per access reduces by half. This requires a clock gating feature that avoids the pre-charging of the sub-bank not being utilized, which must be decided early during the memory access stage. For block buffering, the traditional approach requires that the entire array be precharged, independent of whether there is a hit on the buffer, which has the advantage of completely eliminating any performance penalty but has a negative impact from the clock energy perspective. A modified approach considered here, requires array pre-charging only if there is no valid information on the buffer (buffer miss), which in turn causes a performance penalty of an additional cycle on a buffer miss.

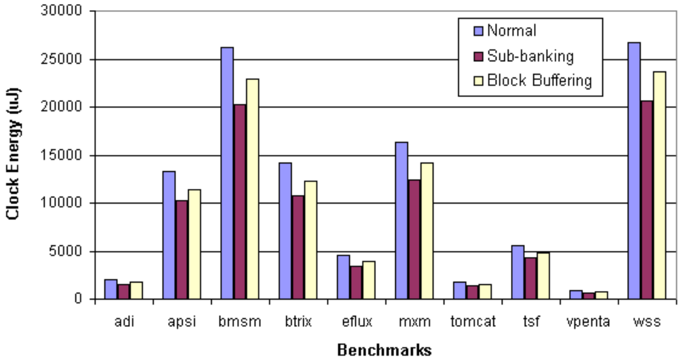


Figure 4–12: Clock energy for cache power optimizing hardware schemes.

Figure 4–12 shows clearly that both block buffering and sub-banking are effective ways of reducing clock energy in addition to cache energy. It must be pointed out that these savings come at no performance cost when using sub-banking, resulting in a more energy-efficient alternative than block buffering. However, note that the block buffer scheme would be preferred over sub-banking, if the objective is to minimize EPC (see *Figure 4–13*) rather than overall energy. This is due to the smaller load that the block buffer presents to the clock network rather than the sub-banked cache array on block buffer hits. The performance penalty due to block buffer misses that results in more execution cycles also affects the observed EPC trend.

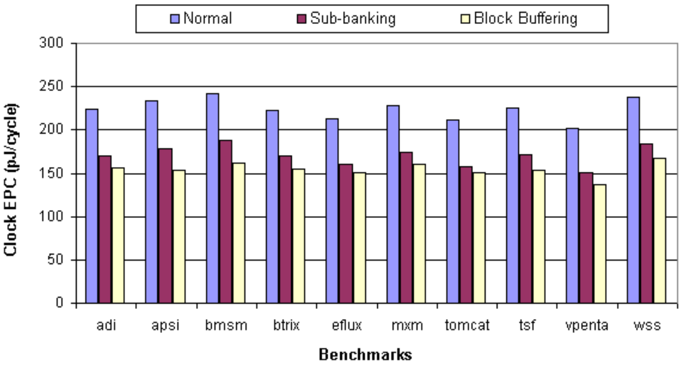


Figure 4–13: Clock EPC for cache power optimizing hardware schemes.

4.2.2 Clock gating at various levels

In the experiments performed so far, only mid-level clock gating has been used where the cache load is disabled during cache misses. Now, the influence of gating at a higher and lower level of granularity is evaluated. For the first case, an additional performance parameter was added: the latency of the off-chip (main) memory. Further experiments considered three cases, which defined an average speed memory (AM), a fast memory (FM) that reduced latency of the previous by half, and the same fast memory with the addition of a write buffer (FM +WB).

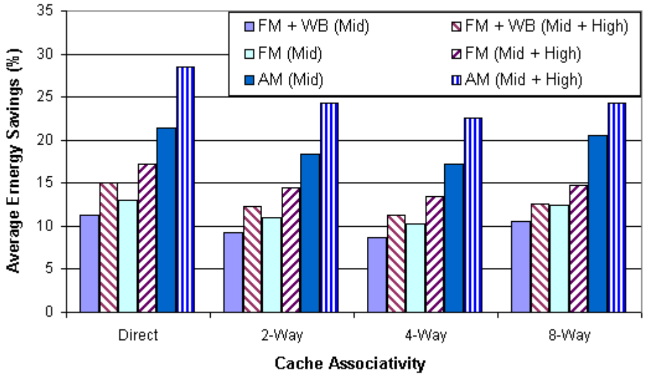


Figure 4–14: Average clock energy savings when implementing mid-level and high-level gating, as function of associativity.

Figure 4–14 and *Figure 4–15* show the resulting clock energy savings for the three cases above mentioned when high-level clock gating is used. Additionally, to establish a reference, the energy savings already provided by mid-level gating are captured. From both figures, it is clear that the larger the main memory latency and/or the miss ratio are (in particular, see the values of the 4KB configuration), the larger the improvement on clock energy savings obtained by high-level gating. In general, independent of the configuration, high-level gating provides further energy savings beyond those obtained by mid-level gating without performance penalty.

Table 4–5 summarizes the results obtained. Looking at the base case (AM), significant savings can be obtained (up to 22.11%) by adopting this simple mechanism. Note that for all three main memory configurations, the maximum savings are obtained when there is no associativity (direct mapped cache) and the cache size is the smallest, which represents the cases where the cache miss ratio is maximized. As associativity and size increase, the energy savings lessen.

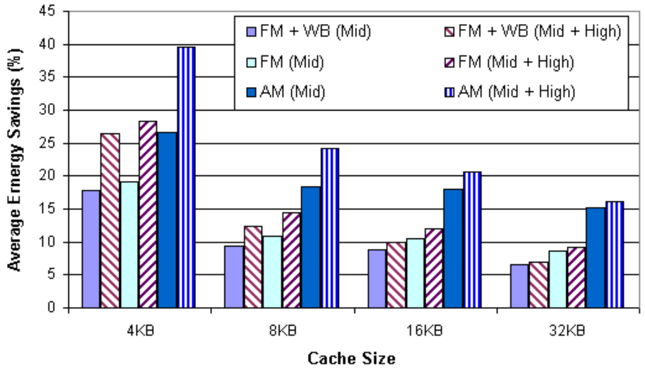


Figure 4–15: Average clock energy savings when implementing mid-level and high-level gating, as function of cache size.

Table 4–5: Impact of high-level gating on clock energy.

Improvement	Savings (%)	AM	FM	FM+WB
Cache Associativity (Size is 8KB)	Average	7.50	3.97	3.30
	Max. (Direct)	9.64	5.14	4.37
	Min. (8-way)	5.41	2.83	2.31
Cache Size (Associativity is 2W)	Average	8.67	5.10	4.55
	Max. (4KB)	22.11	13.84	12.88
	Min. (32KB)	1.17	0.59	0.44

Now, the effects of low-level gating are evaluated by enabling the selective pipeline register gating strategy described earlier. *Figure 4–16* shows the reported clock energy with and without low-level gating while *Table 4–6* summarizes the average results across all benchmarks. At this finer level of granularity, clock gating is able to provide larger average savings on clock energy than those obtained by high-level gating. Since the reduction in clock load is obtained in every cycle where the system is doing useful work (which is the dominant case), its contribution to lessening clock (and system) energy is larger than that obtained through a higher level of gating that only yields savings during the time that the units are not used (which ideally should be minimal). The clock load in this case is smaller than that present at higher levels of the clock distribution hierarchy, so there is a trade-off between the size of the load and how often it is gated.

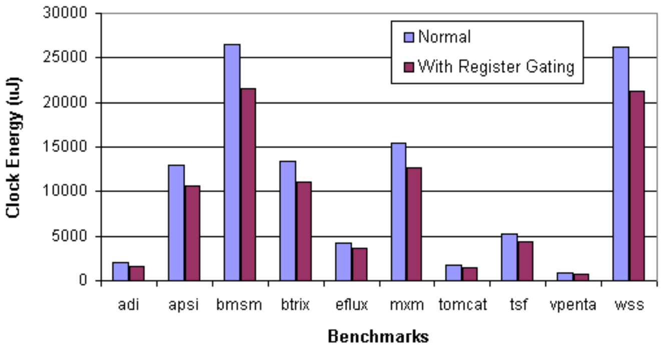


Figure 4–16: Low-level clock gating impact on clock energy.

Table 4–6: Impact of low-level gating on clock energy.

Savings (%)	Average	Maximum
Datapath	25.45	31.90
Clock	18.47	19.52
On-chip	6.12	7.11

4.2.3 Impact of memory-oriented software optimizations

Next, the effect of memory-oriented software optimizations on clock energy is evaluated in the original base architecture. Such optimizations are particularly useful for array-dominated media applications. The loop optimizations used in this study include loop interchange (which permutes the order of two loops in a nest to enable temporal reuse and/or unit-stride accesses in inner loops), loop tiling (which automatically creates the blocked version of a nested loop, thereby exploiting temporal locality across multiple, not necessarily innermost, loops), and unroll-and-jam (which basically creates multiple copies of loop bodies by repeating the statements in the loop for different iterations and reducing the trip count of the enclosing loop). The data transformations employed, on the other hand, modify the memory layouts of multidimensional arrays to achieve high spatial locality in cases where loop transformations fail. Although these optimizations improve data reuse and cache locality [52] (which is a benefit from the overall memory/cache energy perspective), they also tend to increase the complexity of array subscript expressions and transformed loop bounds, which in turn, can increase the arithmetic and branch operations executed by the processor (i.e., an increase in datapath energy). Their influence is of interest as the cache precharge and associated buffer energies contribute significantly to the total clock power.

It is of interest to evaluate the impact of: (1) the increase on the number of datapath operations performed during the execution of the optimized codes and (2) the improvement on the cache hit rates, which reduces the number of memory stall cycles. *Figure 4–17* shows the changes in clock energy after the optimizations described were evaluated. It can be observed that for most cases, the increase in execution cycles dominates any reduction in stall cycles (which waste energy in the PLL, main driver and distribution tree) as far as clock energy is concerned. Additional increments were found in the datapath and cache energy.

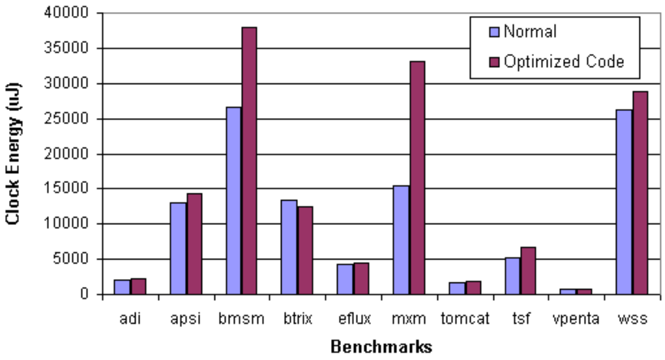


Figure 4–17: Software optimizations impact on clock energy.

When the off-chip main memory is also considered, an average energy reduction of 37.8% was obtained at the system (on- and off-chip) level. *Figure 4–18* and *Figure 4–19* compare the energy distribution before and after the compiler optimizations are applied, while *Table 4–7* summarizes the average and maximum variations across all system components.

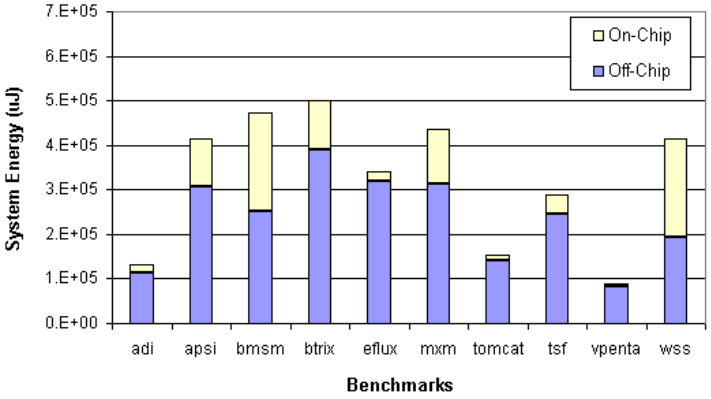


Figure 4–18: The total system energy before applying code optimizations.

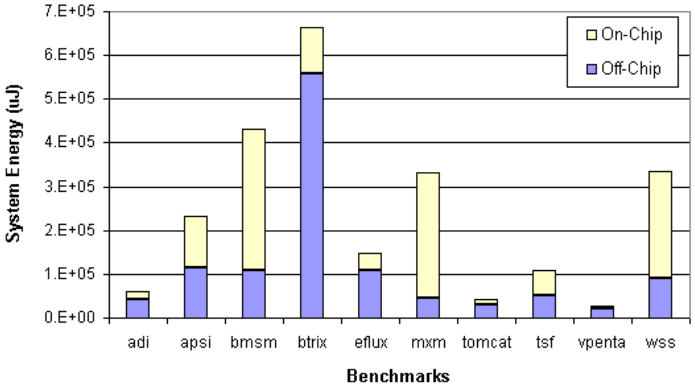
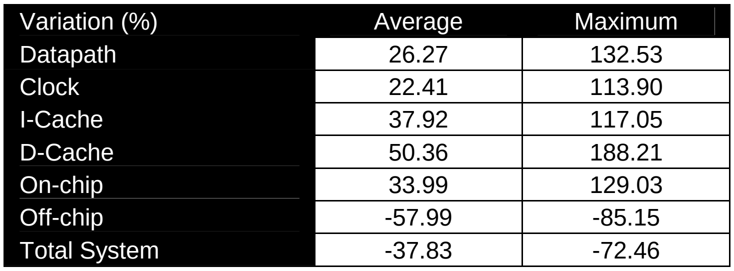


Figure 4–19: The total system energy after applying code optimizations.

Table 4–7: Energy variations across all units after implementing software optimizations.



Variation (%)	Average	Maximum
Datapath	26.27	132.53
Clock	22.41	113.90
I-Cache	37.92	117.05
D-Cache	50.36	188.21
On-chip	33.99	129.03
Off-chip	-57.99	-85.15
Total System	-37.83	-72.46

Chapter 5

EFFECTS OF TECHNOLOGY SCALING ON CLOCK POWER

Now that the model has been used for evaluating the impact of various design and run-time optimizations, the effect of technology scaling can be studied. In this way, attention is focused on determining whether future hotspots of clock energy are different from those identified so far.

5.1 Effects of scaling on system power consumption

The first step in predicting the behavior of the clock subsystem as technology shrinks is to scale the base design such that it will resemble a machine characteristic of a given future technology, since improvements obtained from technology scaling are usually tied to architectural enhancements. Eq. 5.1. gives the expression for the system-level dynamic power, where N_t is the number of transistors in the design, C_{avg} is average capacitive load of a transistor, (taken as that of an average-size device), V_{dd} is the power supply, f_{clock} is the operating frequency and Act is the activity factor, which accounts for the number of devices that are actually switching (drawing current from the power supply) at a given time.

$$P_{act} = N_t C_{avg} V_{dd}^2 (Act) f_{clock} \qquad (5.1)$$

The speed at which the circuit operates is determined by the time it takes to charge the capacitive load that the average device is driving. Thus, the time required to raise the capacitor voltage to V_{dd}, assuming that the charging current is constant, is given in Eq. 5.2.

$$V_c = \frac{1}{C_{load}} \int I_{dyn} dt = \frac{I_{dyn} t_{gate}}{C_{load}} \qquad \text{thus} \qquad t_{gate} = \frac{V_{dd} C_{load}}{I_{dyn}} \qquad (5.2)$$

In recent microprocessor designs, pipelining has been aggressively used and, in such cases, the operating frequency is determined by the slowest pipeline stage of the design. Thus, Eq. 5.2 can be adjusted so that the cycle time is estimated as given in Eq. 5.3 [53], where I_{on} is the drive current for an average-size device and L_D is the logic depth (i.e., number of gate delays) of the slowest pipeline stage.

$$T_{cycle} = \frac{1}{f_{clock}} = \frac{L_D C_{avg} V_{dd}}{I_{on}} \qquad (5.3)$$

If the expression f_{clock} is replaced in Eq. 5.1, the new expression for dynamic power is given by Eq, 5.4. L_D captures architectural improvements, in which the number of pipeline stages has been continuously increased in order to improve performance. Starting with a reference number given in [53] for a 0.6um technology, L_D is scaled down by a constant factor up to the point were deeper pipelining is basically non-feasible, due to the recovery penalty incurred when such a long pipeline is flushed.

$$P_{act} = \frac{N_t V_{dd} I_{on} (Act)}{L_D} \qquad (5.4)$$

Similarly, the activity factor is scaled up as a way to capture architectural improvements for enhanced Instruction Level Parallelism (ILP). The assumption used here is that *Act* increases by 10% as long as L_D decreases, and once L_D stalls, *Act* increases by 30% assuming dramatic improvements in compilation techniques that improve hardware utilization. The value given in [66] of *Act* equal to 0.015 is used here as the starting point. *Figure 5–1* shows how L_D and *Act* change for the technologies considered, while *Figure 5–2* shows how f_{clock} is expected to change according to the equation above. The scaling factor of 25% percent for L_D was chosen for consistency with industry data. For instance, for a Pentium processor in a 0.6um technology, L_D was in the order of 50 while for current designs like the Pentium 4 (0.18um), it is below 10.

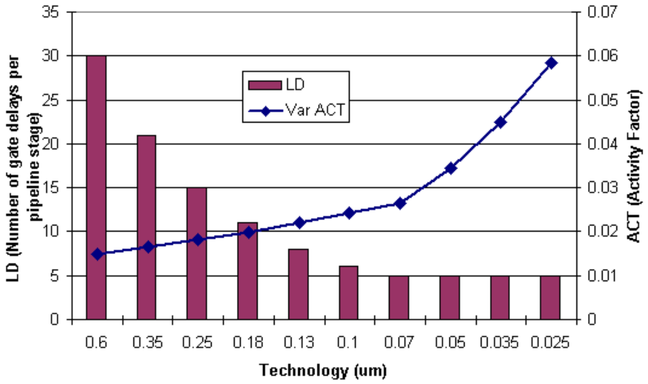


Figure 5–1: L_D and *Act* changes across technology generations.

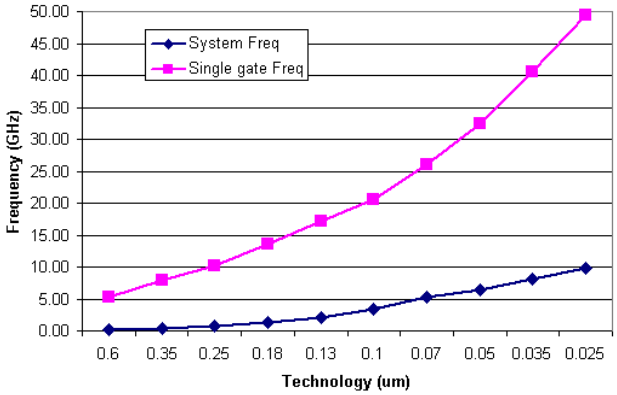


Figure 5–2: Projected operating frequency behavior across technology generations.

Besides L_D and *Act*, all other factors in Eq. 5.4 scale with technology but at a predictable rate depending on the scaling laws followed. It was shown [53] that industry was using Quasi-Constant Voltage (QCV) rules up to the end of the last decade, but as practical limits were found, the scaling rules had to be adjusted, such that scaling is still feasible well into the deep-submicron regime. *Table 5–1 s*hows some of the basic parameters for the technologies considered. The scaling models presented in [53] were used to predict what would happen in future designs and some older and current technologies were consider in order to compare the estimates against real numbers. A brief review of the scaling models is given in Appendix E.

Table 5–1: Main parameters for the technologies considered.

Feature \ Tech.	**0.6**	**0.35**	**0.25**	**0.18**	**0.13**	**0.1**	**0.07**	**0.05**	**0.035**	**0.025**
V_{dd} (V)	5	3.3	2.5	1.8	1.4	1.15	0.85	0.65	0.5	0.4
V_{th}@0V (V)	0.6	0.47	0.41	0.35	0.31	0.27	0.23	0.20	0.17	0.15
V_{th}@Vdd (V)	0.43	0.34	0.30	0.26	0.23	0.21	0.18	0.15	0.13	0.12
t_{ox} (nm)	9.8	7.5	6.3	5.4	4.6	4.0	3.4	2.8	2.4	2.0
Min. feature - λ (um)	0.35	0.2	0.15	0.11	0.08	0.06	0.04	0.03	0.02	0.01
Trans. W_{min} (um)	1.05	0.61	0.44	0.32	0.23	0.18	0.12	0.09	0.06	0.04

Now, each of the remaining items in Eq. 5.4 will be considered individually. It is assumed that short-channel effects (SCE) dominate and the drive current (I_{on}) no longer can accurately be described in terms of the classic quadratic expression, but in the form given in Eq. 3.7 which is repeated below for convenience. Eq. 5.6 gives *k(V_{ds})*, which provides a measure of the degree of saturation.

$$I_{on} = k(V_{ds})WC_{ox}v_{sat}\left(V_{dd} - V_{th}\right) \tag{5.5}$$

$$k(V_{ds}) = \left(1 + \left(\frac{V_{ds}}{L_{eff}E_{sat}}\right)\right)^{-1} \tag{5.6}$$

C_{ox} and W are scaled according to the design rules, while v_{sat} is assumed to remain constant and $k(V_{ds})$ is evaluated for $V_{ds} = V_{dd}$. V_{dd} and V_{th} are scaled as shown in *Figure 5–3*, which also captures the estimated effect of Drain Induced Barrier Lowering (DIBL) on the nominal threshold voltage. DIBL is accentuated at shorter channel lengths and higher drain voltages and it captures the reduction in the potential barrier at the device's source terminal due to the proximity of the drain's depletion region [3]. However, it is important for high performance that $V_{dd} > 3V_{th}$, so that enough current drive is available.

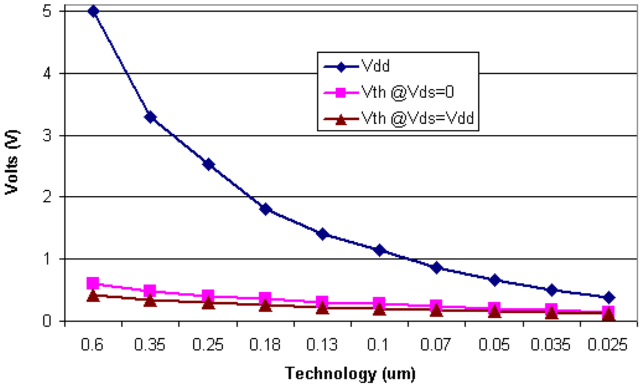


Figure 5–3: V_{dd} and V_{th} scaling across technology generations.

Finally, the number of transistors (N_t) is estimated by dividing the total die area by the area taken by a minimum-size device with individual contacts and some spare area around it, that was calculated as $126\lambda^2$ (λ is the minimum feature size given in Table *5–1*). This approach attempts to balance the effect of very compact structures like memories and other structures not so regular. Figure *5–4* shows how the transistor count varies for two cases: the first case assumes that the die size remains constant at a value of 80mm^2 while the second assumes an increase of 14% in die size from one technology to the next [67]. The first assumption comes from the fact that 0.5um designs from five years ago and current 0.18um designs exhibit similar die sizes. This could be a practical trend as power density, power delivery and clock skew synchronization problems may prevent the size increase proposed in [67].

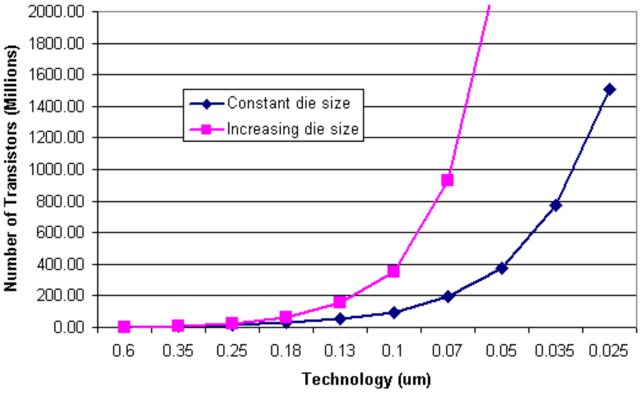


Figure 5–4: Transistor count per die as technology scales down.

Eq. 5.7 gives an estimate of the leakage power consumption, where at the system level, where K_L is a factor that accounts for the distribution/sizing of P and N devices, the stacking effect, the idleness of the devices and the design style used. The expression in Eq. 5.7 is similar to that given in [54]. In a CMOS-based style, at most half the transistors are actually leaking while the remaining are ON (resistive region).

$$P_{leak} = N_t V_{dd} I_{off} K_L \qquad (5.7)$$

Transistor stacking can further reduce leakage but its net effect is hard to determine as it depends on the actual signal states during execution. The stacking effect is estimated by Eq. 5.8, where N_{stack} captures the number transistors present in optimized leakage stacks while s_{stack} represents the ratio of I_{stack}/I_{device} (approximately 0.1 [68])

$$K_{stack} = 1 - \frac{N_{stack}}{N_t}\left(1 - s_{stack}\right) \qquad (5.8)$$

Conflicting with the savings due to stacking, there is additional leakage in large non-stacked devices like clock buffers and I/O buffers. Thus, it is possible to define a factor $K_{drivers}$, which is expressed in a similar fashion to K_{stack}, with the difference that $N_{drivers}$ captures the equivalent number of minimum-size transistors used as buffers while $s_{drivers}$ represents the average increase in width that causes a direct increase in leakage current. The logic depth factor (L_D), that provides a measure of the idleness of a gate, can be used to estimate the portion of the clock cycle that is actually used by the devices for state switching. That is, if L_D = 5, during one fifth of the cycle period the gate is either switching high to low or low to high, while during the rest of the cycle, the signals remain steady at the final intended state. Eq. 5.9 uses L_D for the definition of the idle factor K_{idle}.

$$K_{idle} = 1 - L_D^{-1} \tag{5.9}$$

In [54], experiments showed that K_{style} is around 10 for logic, while for memory structures it is close to 1. Given that future designs (and even current designs) are equally dominated by memory structures and logic, then the choice of K_{style} = 5 seems reasonable. Then, the total K_L factor is given by Eq. 5.10.

$$K_L = K_{sizing} K_{stack} K_{drivers} K_{idle} K_{style} \tag{5.10}$$

Subthreshold conduction is not the only leakage mechanism but is by far the most influential. Other mechanisms include increased leakage caused by DIBL effects, gate leakage (gate oxide tunneling), Gate Induced Drain Leakage (GIDL), PN reverse bias current leakage, punchthrough, narrow width effect and hot carrier injection [56]. For this study, the effect of the first three mechanisms was considered. *Figure 5–5* illustrates how the various leakage mechanisms compare to each other, which reinforces the statement that subthreshold conduction is the dominant mechanism.

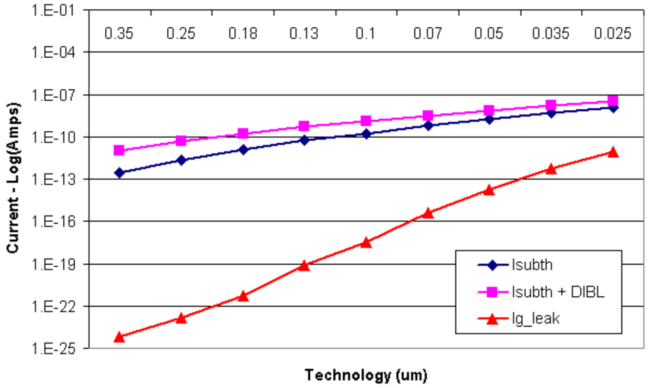


Figure 5–5: Leakage mechanisms comparison.

Eq. 5.11 was used to estimate subthreshold leakage [69], where μ is the carrier mobility, ϕ_T is the thermal voltage (= *KT/q*) and C_{dep} is the capacitance of the channel depletion region. In the above plot, the subthreshold leakage current per transistor is calculated for both V_{th} at V_{ds} = 0 and V_{th} at V_{ds} = V_{dd}, with the latter capturing the DIBL effect of reduced V_{th}.

$$I_{sub} = \frac{\mu C_{ox} W}{L_{eff}} (\eta - 1) \phi_T^2 \exp\left(\frac{V_{gs} - V_{th}}{\eta \phi_T}\right) \qquad \text{where} \qquad \eta = 1 + \frac{C_{dep}}{C_{ox}} \qquad (5.11)$$

For gate leakage only the effects of direct oxide tunneling are considered, as the effects of Fowler-Nordheim (FN) tunneling were found to be basically negligible for the technologies studied. This was expected because FN tunneling normally occurs at higher field strengths [3] than those present in the cases studied. The direct tunneling current density was calculated using Eq. 5.12 [70], where A_{dt} is calculated as 6.17uA/V, B_{dt} is 5.52/nmV^{½} and ϕ_B is the oxide energy barrier (= 3.25V). The impact of GIDL has been difficult to quantify, but initial estimates indicate its influence as basically insignificant.

$$J_{dt\_tunn} = \frac{A_{dt}}{T_{ox}^2}\left[\phi_{dt}\exp\left(-B_{dt}T_{ox}\sqrt{\phi_{dt}}\right)-\phi_B\exp\left(-B_{dt}T_{ox}\sqrt{\phi_B}\right)\right] \quad \text{where} \quad \phi_{dt} = \phi_B - \frac{V_{ox}}{2} \qquad (5.12)$$

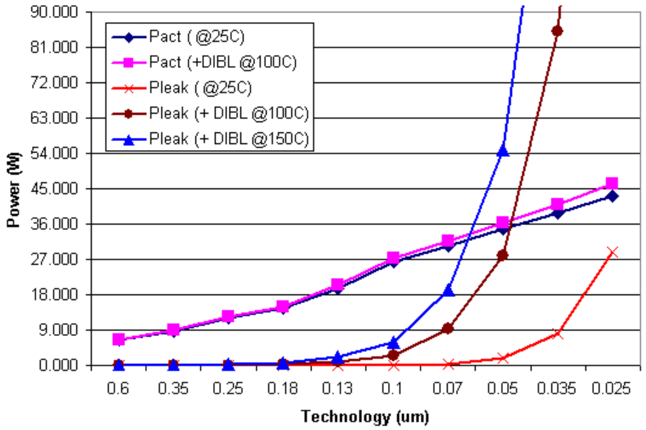


Figure 5–6: Active and leakage on-chip power for constant die size.

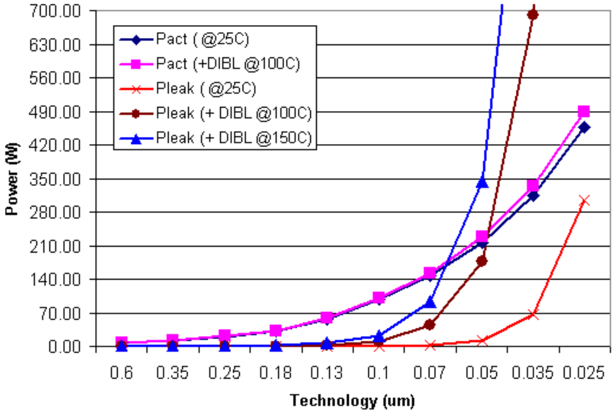


Figure 5–7: Active and leakage on-chip power for increasing die size.

Now that all the significant contributors have been analyzed, both the active and leakage power components can be estimated. *Figure 5–6* and *Figure 5–7* illustrate how dynamic and static (leakage) power vary across the technologies considered. Only the influence of subthreshold currents has been captured as they are the dominant leakage mechanism. The impact of DIBL is more significant in increasing leakage power than active power.

Further, the effect of temperature has also been captured. It is known that mobility decreases as temperature increases but this effect is minimized as the doping density increases [32] which is expected as technology scales [53]. The threshold voltage is also affected by temperature changes, in particular those on ϕ_F, which represents the voltage difference between the intrinsic energy level and the Fermi level. The total net impact on ϕ_F is difficult to visualize since temperature increases both the intrinsic carrier concentration (n_i) and the thermal voltage (ϕ_T) and these changes have opposite effects as Eq. 5.13 shows [55], where N_a represents the doping concentration.

$$\phi_F = \phi_T \ln\left(\frac{N_a}{n_i}\right) \tag{5.13}$$

In general, the net effect is a reduction in ϕ_F, which translates into a lower threshold voltage. The assumption here is that this effect (which increases I_{on}) compensates with any possible changes in the mobility, so that no adjustments are required for the active component of power. This is also justified given that scaling of T_{ox} is expected to reduce the effect of temperature in V_{th} [3]. In terms of subthreshold leakage, besides the decrease in ϕ_T, the transistor log(I_{ds}) versus V_g subthreshold slope (S_t) increases linearly with temperature. It has been shown [1] that S_t has not increased (and is not expected to increase) as technology scales due to scaling of T_{ox} and improvements in doping profiles. These effects are captured in Eq. 5.11 where ϕ_T is calculated at the new temperature of 100^0C (373^0K). These effects are captured in Eq. 5.11 where ϕ_T is calculated at temperatures of 100^0C (373^0K) and 150^0C (423^0K). Note in *Figure 5–6* how the leakage power contribution doubles when the temperature increases from 100^0C to 150^0C.

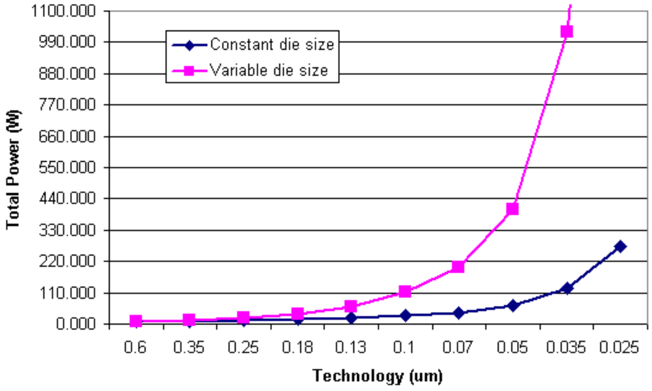


Figure 5–8: Total power variation for the two die size cases.

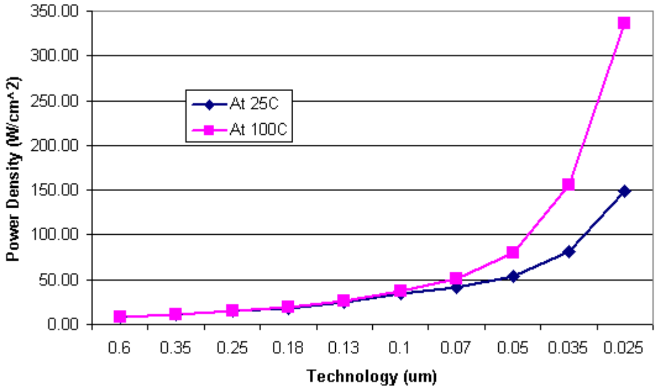


Figure 5–9: Power densities for non-scaling die size.

The default case assumes the presence of DIBL effects and an operating temperature of 100^0C, although temperatures of 150^0C are already observed in some current designs and may become standard in the near future, if cooling mechanisms are not significantly improved. *Figure 5–8* shows the total power and *Figure 5–9* shows the power density variation across technologies, which provides a first possible solution to keep power down to reasonable levels: limit the die size. Efficient temperature-control mechanisms, if obtainable, can compensate for the heat incurred with larger die sizes that may be required in highly integrated systems.

In addition to the above mentioned design constraints (die size and operating temperature), the process technology will also provide means for limiting power consumption. In particular, there are two technology enhancements that have a strong impact in both power and performance and are expected to become standard in mainstream CMOS products within the next five years [71].

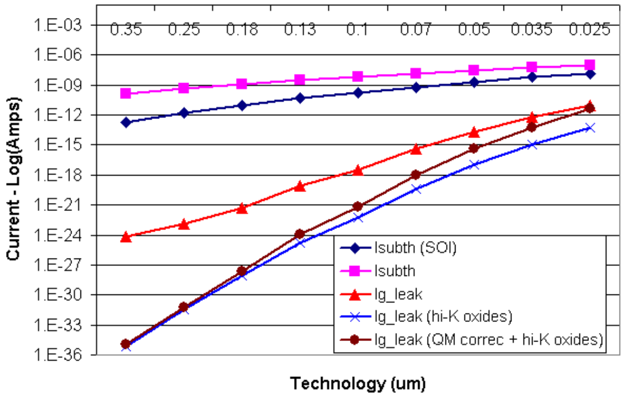


Figure 5–10: Leakage mechanisms comparison after incorporating technology improvements.

The first improvement requires replacing the material commonly used to build the gate oxide (SiO_2), with materials with a larger dielectric constant (high-K materials). For the small dimensions of future oxide layers, the thickness of the inversion layer beneath the oxide makes the apparent electrical thickness (measured by finding C_{ox}) significantly larger than the actual physical thickness. It has been shown that the electrical thickness can be from 0.5nm up to 1.0nm greater than the physical thickness [72]. Thus, it seems very likely that after the 0.13um generation, gate oxides will be fabricated with high-K materials, which will allow maintaining the physical thickness at an approximately constant value, while the electrical thickness is reduced. These materials are also expected to dramatically reduce gate leakage for the same physical oxide thickness, due to a higher oxide energy barrier (ϕ_B), as shown in *Figure 5–10**.***

Quantum-mechanical models, that consider the effects of the inversion layer, have been formulated recently [73, 74] and can accurately estimate the electrical thickness from the C_{ox} measurements (C-V curves) with reported values within 0.1nm of the real physical thicknesses. These models have also estimated the discrepancy between the electrical and physical thicknesses to be about 0.3nm, for thin oxides (below 4nm). Although the model given in [74] cannot be easily tuned to obtain the leakage currents for the technologies studied, it is important to perform an estimation of the effect on the apparent thickness of the oxide as technology scales. Thus, based on a maximum deviation of 0.3nm for a 0.025um technology, rough but small adjustments were performed on the used t_{ox} values (see Table 5.1) to reflect an increase in the gate-tunneling currents for technologies that require ultra-thin oxides (0.18um and below). *Figure 5–10* shows how the corrected estimates of gate leakage will still be negligible if high-K oxides are used, which as mentioned, are expected to become standard in the near future.

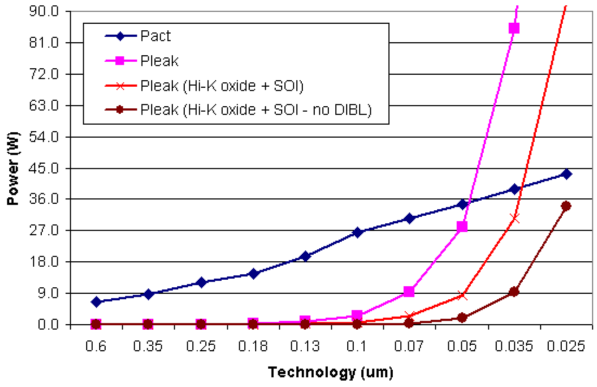


Figure 5–11: Impact of SOI and high-K dielectrics in the leakage system power (constant die size)

The second improvement is the replacing of bulk CMOS by SOI (Silicon On Insulator). It seems likely that a majority of the industry will be using SOI for 0.1um processes and beyond. SOI has a significant impact on power by virtually eliminating the diffusion capacitance and allowing for steeper subthreshold slopes. In particular, the subthreshold slope is given by $S_T = \ln(10)\eta.\phi_T$ [3]. For bulk CMOS, η is around 1.4 which, at 100^{O}C, leads to a subthreshold slope of about 100mV/decade. In SOI, η is close to 1, such that S_T becomes 75mV/decade. *Figure 5–10* shows how subthreshold leakage is effectively reduced with the use of SOI. *Figure 5–11* captures the impact of the mentioned improvements on the total system power. It is assumed that dynamic power remains the same as in the bulk CMOS case, although the operating frequency is higher. In the optimum case (when DIBL effects are completely minimized by SOI), leakage is always less than active power for the technologies considered. But as process variations continue to influence the device parameters, the actual effect is not as ideal but translates into delaying the surge of leakage power by one technology generation (i.e., for this study, from 0.035um to 0.025um, as shown in *Figure 5–11)*.

The above discussion of technology impact on leakage is not expected to be comprehensive but merely informative, so that a more realistic view of the problem can be taken. The assumption of gate leakage being negligible compared to subthreshold conduction remains and the impact of SOI will be captured later, when the role and extent of leakage reduction techniques is discussed.

5.2 Effects of scaling in clock power consumption

Now that models that effectively capture the influence of scaling on power consumption are available, the proposed clock model can be used to predict how the contribution of clock power will change in future technologies. The base design used for this evaluation assumes that the die size does not scale with technology following observations made earlier. In addition, this eliminates the need to consider more complicated clocking mechanisms as it has been formulated that if the die size remains constant, tree-based clock distribution networks can be used well into the deep-submicron regime [23]. Since the total clock power depends on the presence or not of various structures as well as the architectural parameters associated with them, further assumptions need to be made in order to allow the model to be applied. The architectural trends for structures like on-chip memory, number of pipelines, etc., are formulated such that the numbers obtained closely match the characteristics of commercial medium to high end products fabricated in current (i.e., 0.18 and 0.13um) and older technologies (0.6, 0.35 and 0.25um). In particular, the L_D and *Act* factors described earlier are used as indicators of how the architectures are expected to evolve so that estimates of all required parameters can be determined. The basic assumptions made were:

- The total chip area was divided according to the type of circuitry implemented on it - either logic or memory. The amount of area required by the logic was estimated from the amount of area required in the preceding generation as given by Eq. 5.14, where S_L is the scaling parameter as defined in Appendix E and the subscripts *_p* and *_c* refer to the previous and current technologies, respectively.

$$A_{\log ic\_c} = A_{\log ic\_p} \cdot S_L^2 \cdot \left(\frac{L_{D\_p}}{L_{D\_c}} \right) \cdot \left(\frac{Act_c}{Act_p} \right) \tag{5.14}$$

- The area used for memory structures was assumed to be the remaining available area ($A_{mem} = A_{die} - A_{logic}$). The total possible on-chip memory storage capacity was estimated using Eq. 5.15, where K_{array} is a factor that accounts for additional control circuitry. Note that this factor considers all on-chip user controlled memory (i.e., L1, L2 and L3 caches).

$$Size_{mem\_c} = Size_{mem\_c} \left(\frac{A_{mem\_c}}{A_{mem\_p}} \right) \cdot K_{array} \tag{5.15}$$

- The data width was doubled every three generations while the total addressable memory space was scaled proportional to the scaling of the on-chip memory. In this way, the address bus width can be estimated. The size of the register file increases following the increase in *Act*, such that doubling of the register file occurs every generation. Once pipelining is exhausted, the number of ports also increases with *Act*. Similarly, the TLB size increases with the estimated increment of the L1 caches.
- The datapath width increased with the data width. The number of pipeline stages increased proportionally to the decrease of the logic depth (L_D), while the number of pipelines (execution paths) increased proportionally to the change on *Act*.
- Additional memory structures such as branch history tables, issue windows, scratch pad memory, etc, were assumed to be proportional to the on-chip memory estimated earlier, but considerable smaller. Similarly, in order to consider dynamic logic, Eq. 5.16 provides an estimate of the fraction of transistors that are actually a load to the clock network. In the equation, K_{dyn} captures the fraction of the logic that is actually implementing functional units (not pipeline registers or embedded memory structures), K_{gates} represent the number of those devices that are implemented with dynamic gates while $K_{devices}$ gives the fraction of clocked transistors per gate. This estimate is rough but it is used to provide a more complete picture. For now, it will be assumed that both high-performance dynamic CMOS and an all-static CMOS implementations are possible, with the latter incurring no dynamic circuitry load in the clock network.

$$N_{Tdyn\_c} = N_T \left(\frac{A_{\mathrm{log}ic}}{A_{die}} \right) K_{dyn} K_{gates\_p} \left(\frac{L_{D\_p}}{L_{D\_n}} \right) K_{devices} \qquad (5.16)$$

For all the loads described so far, the size of the required buffers was calculated as described in Chapter 2 and then used to estimate the total buffer load. For the clock distribution, an H-tree was assumed for all cases with the number of levels in the tree increasing if the number of gates within the area of a terminal point doubled. *Table 5–2* shows how key architectural parameters change according to the assumptions presented.

Table 5–2: Behavior of various architectural parameters.

Feature \ Tech.	0.6	0.35	0.25	0.18	0.13	0.1	0.07	0.05	0.035	0.025
$\text{Area}_{\text{logic}}$ (%)	80	77	61	48	38	33	23	15	15	16
Memory (kB)	32	64	96	192	384	640	1280	2304	4096	6656
Data Width	32	32	32	64	64	64	64	128	128	128
Execution Paths	4	5	6	7	8	9	10	13	17	23
Dyn. Logic (%)	0.05	0.07	0.1	0.14	0.2	0.25	0.3	0.36	0.43	0.52
H-tree levels	6	7	7	8	8	9	9	10	10	11

As mentioned earlier, a high-performance system with aggressive use of dynamic logic and an embedded-system-oriented processor that uses only static CMOS are considered. The first task is to determine the role of the clock in the power budget of future designs. *Figure 5–12* captures the percentages of clock power for the two cases with respect to both the total chip power and the dynamic component. It can be observed that in both design cases, clock power will remain a significant portion of the chip power as long as dynamic power dominates. When leakage power becomes significant (beyond the 0.1um feature size), the contribution of the clock to the total power begins to decrease because the clock leakage contribution is less than 1% of the total leakage power.

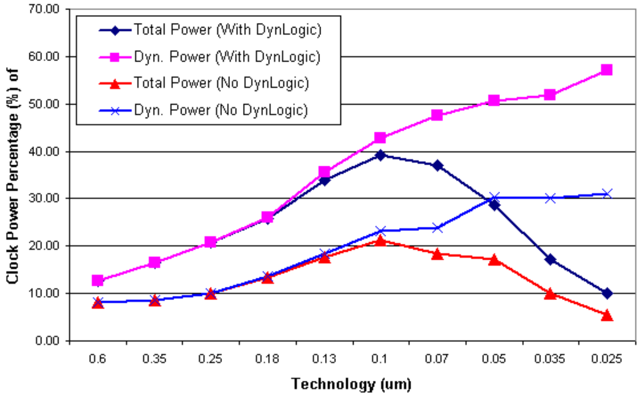


Figure 5–12: Expected clock power contribution to the total dynamic and on-chip power without utilization the leakage reduction techniques.

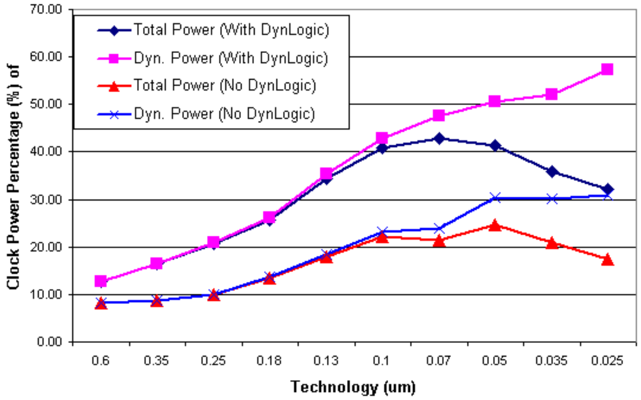


Figure 5–13: Expected clock power contribution to the total dynamic and on-chip power with the utilization the leakage reduction techniques.

Figure 5–13 shows that with the application of effective leakage reduction schemes (both during run-time and at design-time) that result in a projected 25% decrease on leakage power from one technology to the next, the clock power contribution remains significant. It should be noted that proposed practices like dual-edge-triggered logic will certainly help maintain the contribution of clock power in the range of 30% to 50% of the total power.

Figure 5–14 shows the relative contribution of the various loads to the total clock load for the embedded case, while the absolute distribution for the high performance case is shown in *Figure 5–15*. In general, the relative contributions remain basically constant across all technologies considered (for the high-performance case, the dynamic logic maintained a very stable contribution of about 25%). This is reasonable given that, as the architecture evolves, all the system resources are expected to grow in a way such that they balance each other. If the architecture is not updated from one technology to the next, the relative contributions change as the interconnect capacitance increases while the parasitic capacitance associated with active devices decreases with the feature size. But in the case portrayed here, since complexity is captured, the effects tend to compensate for each other, resulting in the distribution shown in *Figure 5–14*.

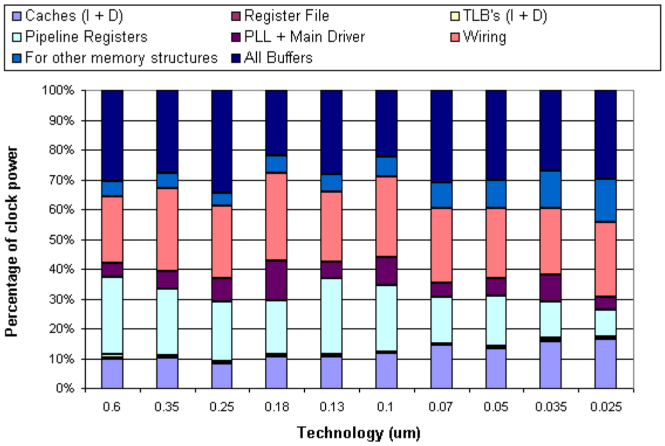


Figure 5–14: Relative clock effective capacitance breakdown for the all-static CMOS logic case.

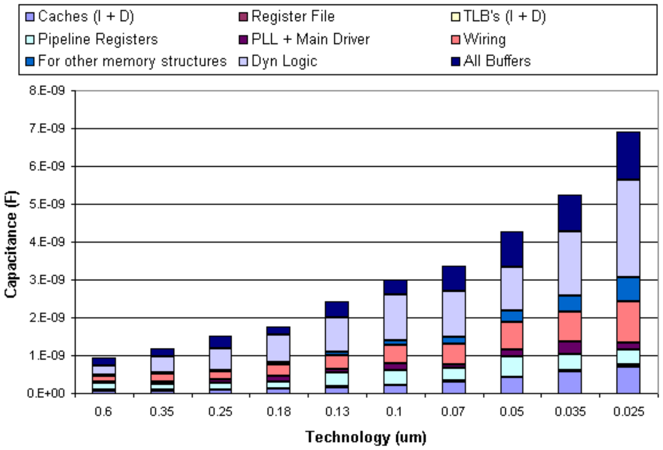


Figure 5–15: Absolute clock effective capacitance breakdown for the dynamic CMOS logic case.

Figure 5–15 shows that the total clock load increases with technology, providing a clear picture that the increase in the number of loads overwhelms the potential reduction achievable by the scaling of the parasitic capacitances. In this way, it is clear that an almost constant clock load distribution does not mean a constant total clock load. In general, the impact of interconnect is captured as structures with lots of wires increase their contribution with respect to those that are dominated by the capacitance of active devices (i.e., memory structures vs. pipeline registers). Although the buffer capacitance is basically due to transistors, its contribution remains basically constant since the increase in the driven loads compensated for the scaling of the feature size.

5.2.1 Scaling of interconnect capacitance

As mentioned earlier, the wire capacitance is expected to increase as technology scales due to the increase of the interwire capacitance as a consequence of the reduction of the minimum possible wire separation.

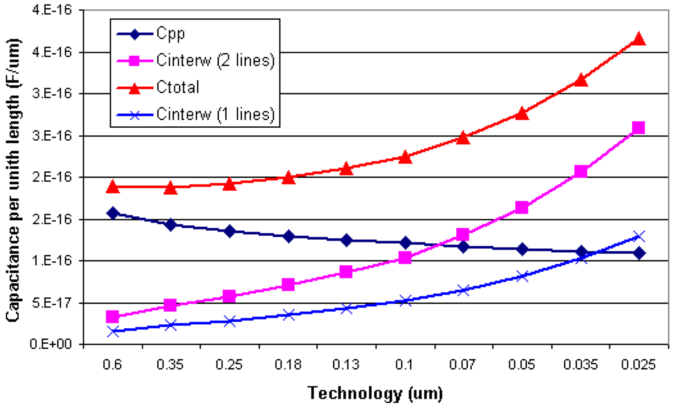


Figure 5–16: Trends of the interconnect capacitance components.

Figure 5–16 shows how the parallel plate (which includes the both the area and fringe contributions) and interwire components of the interconnect capacitance vary as technology scales. It can be seen that although the parallel plate component decreases, the influence of the interwire component increases the total capacitance. In the worst possible case, the wire under consideration is surrounded by the two other lines, which are separated by the minimum possible distance. The figure also shows the coupling capacitance if only one line is present, which is also considerable.

Figure 5–17 shows the expected relative behavior of the various parasitic capacitances. The wire capacitance is estimated for both local (length is proportional to the minimum feature size - λ) and unit-level interconnection (length is set to 1um for fixed die size). It can be seen that the local interconnect scales with the other parasitics but becomes more important than the diffusion capacitance after the 0.1um technology. On the other hand, for the same technology, the 1um-long wire capacitance becomes larger than the gate capacitance. This clearly illustrates the problems incurred when very large interconnect networks are required and emphasizes again the importance of using global interconnects sparingly.

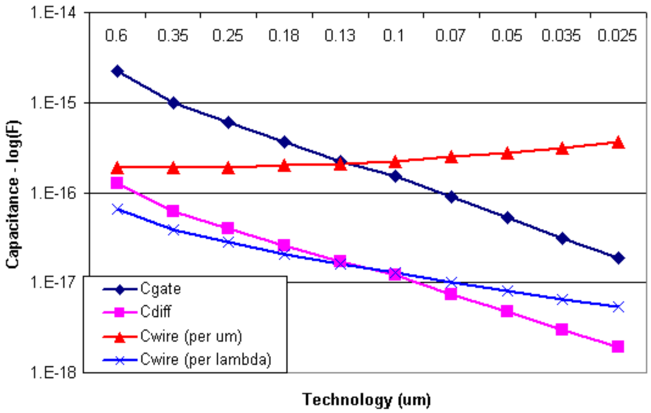


Figure 5–17: Trends of the technology-defined parasitic capacitances.

Now that the impact of the various contributors of the wire capacitance has been analyzed, *Figure 5–18* shows the expected increase of both the interconnect and transistor generated clock loads, including the case where the design has only static gates. It can be seen that for the high-performance design, both components increase in a similar fashion while maintaining their relative contributions at an almost constant level. In contrast, for the embedded design, the interconnect component dominates. The estimates consider the nature of the capacitance that determines the load size rather than how the actual load is implemented. For instance, in the memory structures, the load due to precharge transistors is captured as both interconnect and active capacitance because the transistors' sizing is determined by wires and diffusion capacitances in the bitline.

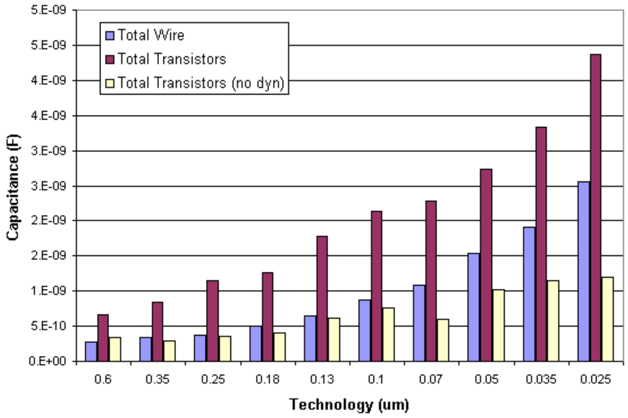


Figure 5–18: Sharing of total clock capacitance among interconnect and active device generate loads.

The only structure whose load is entirely determined by interconnect is the top-level distribution network. Given the two topologies considered so far (trees and grids), it is of interest to study whether either of these should be favored when systems are implemented in the deep submicron regime. It was found that the interwire capacitance is, for all designs in the various technologies, at least two orders of magnitude less than the parallel plate component. This is expected as the separation between the wires at this level can be measured in tenths of µms, distance enough to yield the interwire capacitance negligible. *Figure 5–19* shows the total distribution network capacitance for grids and trees. Both topologies behave similarly and trees are only marginally better than grids for future technologies while the opposite happens in earlier ones.

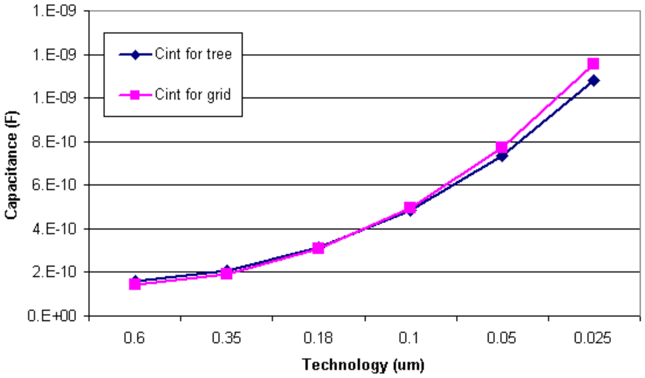


Figure 5–19: Total network capacitance of two possible topologies.

5.2.2 Scaling of leakage power

Although, leakage will become a major power consumption mechanism in future designs, very little can be done in the clock network to reduce its total leakage contribution. The only sources of leakage in the network are the buffers and the clock generation circuitry. While the drivers are very large devices, it was found that for all technologies considered, the share of the clock power that is due to leakage is at most 2.5%, which occurs for the 0.025um technology (see *Figure 5–20*). Furthermore, the contribution of the clock network to the total leakage power is never larger than 0.4%. Even in the case when leakage reduction schemes are applied, this contribution is still less than 2%. Thus, it is clear that as important as leakage will become in the future, it will have little impact in the way that the power is consumed on the clock network. Even though techniques like *stack forcing* [68] can be used to reduce leakage in buffers by giving away speed, the focus in reducing leakage should be maintained in the bulk of the logic and the memory structures. Appendix F presents a brief analysis of the impact of *stack forcing* on the design of clock buffers.

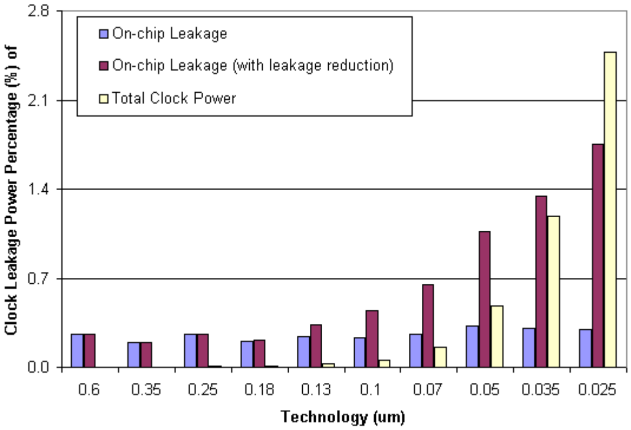


Figure 5–20: Relative clock leakage power contribution to the total clock power and on-chip leakage power.

Chapter 6

CONCLUDING REMARKS

6.1 Summary of contributions

In this work, a complete energy model that covers both the clock distribution and clock generation circuitry was derived. It includes parametric equations for the most common types of loads present in the clock network, including memory structures, registers, buffers and the clock generation device (i.e., a Phase-Locked Loop). The average error (across all components) with respect to circuit-level simulations was less than 10%.

Understanding the importance of a more detailed PLL power model for evaluation of DVS systems and clocking mechanisms on SOC designs, additional research provided accurate equations for PLL power estimation during both the locked and acquisition states. The validation was required not only for the models but also for the circuit parameters like the VCO gain, the filter resistance, among others. The average error deviation of the calculated values with respect to SPICE simulations was always below 5% for both the parameters and the actual models.

By evaluating the impact on clock energy of design-time choices, the effectiveness of the model was corroborated. It was shown that when the clock timing requirements are relaxed (i.e., a slightly smaller memory or a slower buffer are allowed) significant savings in clock power could be obtained. Additionally, for the evaluation of the influence of architectural and topology-related design alternatives on clock energy, the model was incorporated on a cycle-accurate energy simulator such that valuable observations could be obtained from dynamic (runtime) experiments.

A detailed framework that predicts the ‘scaling’ of architectural features was formulated and used in conjunction with already proposed technology scaling parameters. In this way, the impact of technology scaling on the clock network could be evaluated after accounting for the changes in architecture design and optimizations in the process technology.

This work also provides contributions for the development of techniques for leakage power reduction, which is now a big concern, as discussed in Chapter 5. In particular, the feasibility of using PLLs in power supply gating strategies is briefly discussed in Appendix D. Power supply gating schemes can be considered an extension of clock gating and could be applied in parallel, such that dynamic power savings are accompanied with leakage power savings. The feasibility of such a scheme remains an issue of future research.

6.2 Summary of results

The most important results obtained are listed but they could be much more if further research is performed using this work as basis.

- The memory subsystem was found to heavily affect clock energy due to two conflicting aspects caused by variations in the cache size and/or associativity: the clock load increases directly with size and associativity while the number of clock cycles decreases due to improved cache hit rates and a consequent reduction in stall cycles. Optimum cache configurations (2-way), that minimize both clock and on-chip energy, were found for variable associativity, even when the contributions of other units (i.e., caches) were not at their minimum. In contrast, when varying cache size, even though clock and cache energy could not be optimized simultaneously, the total on-chip savings are determined by the cache energy behavior.

- Low power cache configuration schemes like block buffering and cache sub-banking were modified to reduce clock energy in addition to the primary goal of reducing cache energy. The pre-charge circuits of the modified caches were disabled at no-performance cost in the sub-banking case as opposed to a single cycle penalty on a block buffer miss in the block-buffering optimization. These optimizations result in average clock energy savings of 14% and 24% for the block buffering and cache sub-banking schemes, respectively. However, the block-buffering scheme had lower energy per cycle than the sub-banking scheme. This illustrates the need to apply the optimizations based on the need of the system such as prolonging battery life or reducing cooling requirements.

- Clock gating at a finer granularity was found to yield clock energy savings as significant as those obtained with higher levels of gating. In particular, for the base configuration of the architecture, low-level gating energy reductions were 18% on average, while mid-level and high-level gating clock energy savings were 18% and 5%, respectively. The trade-off relates load size and activity, such that the savings depend on the size of the load and how often it is gated.

- The clock generation and the main distribution network of the clock were found to consume a smaller portion of the clock power budget, which explains the results obtained above. This is consistent with data provided by recent publications in the field [23], providing another proof of the accuracy of the model.

- Memory-oriented compiler optimizations influenced clock power in two conflicting ways: the number of executed instructions increased after optimization due to more complex loop and array indexing functions while the number of memory stall cycles decreased after optimization due to better cache performance. When gating is applied during stall cycles, the optimized code usually increased the clock energy as the increased number of execution cycles dominates the energy consumed during stall cycles. The overall on-chip energy increased after optimizations, while off-chip and overall system power reduced.

- PLL power during lock acquisition is strongly dependent on the PLL implementation and the magnitude of the frequency step applied (or caused by changing *N*), but is within ±5% of the power consumed by the PLL once it reaches lock at the target frequency. This overhead corresponds basically to the energy delivered to the loop filter by the charge pump.

- Although, leakage power will play a major role at some point in the future, it will not be the dominant contributor to the total clock power. The relative contributions of the various clock loads will remain constant while the absolute total clock load will increase with every technology generation.

- As leakage begins to dominate, the clock role in the total power budget begins to weaken. But as aggressive leakage reduction techniques and technology improvements will be incorporated in the near future, the clock power will remain a big contributor to the total system power.

Appendix A

ABBREVIATIONS

For the sake of convenience, the abbreviations used in this thesis are listed here although they have been defined throughout the manuscript.

Abbreviation	Meaning
CMOS	Complementary Metal-Oxide Semiconductor
DIBL	Drain Induced Barrier Lowering
DVS	Dynamic Voltage Scaling
DSP	Digital Signal Processor/Processing
EPC	Energy Per Cycle
FDIV	Frequency Divider
FP/FPR	Floating Point / Floating Point Registers
GIDL	Gate Induced Drain Leakage
GP/GPR	General Purpose / General Purpose Registers
ILP	Instruction Level Parallelism
L1/L2/L3	Level 1 / Level 2 / Level 3 Cache
NMOS	Negative-channel Metal-Oxide Semiconductor
NP	Network Processor
PFD	Phase-Frequency Detector
PLL	Phase-Locked Loop

Abbreviation	Meaning
PMOS	Positive-channel Metal-Oxide Semiconductor
RISC	Reduced Instruction Set Computer
RT/RTL	Registrar Transfer / Registrar Transfer Level
SCE	Short-Channel Effects
SOC	System On a Chip
SOI	Silicon on Insulator
SPICE	Simulation Program for Integrated Circuits Emphasis
SRAM	Static Random Access Memory
TLB	Translation Look-aside Buffer
TSPC	True Single Phase Clocking
VCO	Voltage Controlled Oscillator
VLIW	Very Long Instruction World

Appendix B

SCALING OF THE SHORT CIRCUIT POWER

The inverter's short-circuit power was given as Eq 2.10, but is repeated as Eq B.1. Experiments without considering supply voltage scaling were performed, which, according to Eq. B.1, should result in an increase in short-circuit power. This equation was used to estimate short-circuit power for the PFD and FDIV under the condition that the power supply is always 3.3V for all technologies. The obtained results are shown in *Table B–1*. The percentage variation was calculated with respect to the SPICE power measurements, which include the short-circuit contribution. It appears that the cubic factor increment on the (V_{dd}-2V_{th}) difference dominates over the reduction on τ while variations on *k* also contribute to the estimates obtained. This analysis was performed in response to relatively large error margins found between the model and SPICE power values and provided an effective error reduction of 8%. To avoid this unnecessary additional correction, supply voltage scaling must accompany technology scaling.

$$P_{short-circuit/inverter} = \frac{k}{12}\left(V_{dd} - 2V_{th}\right)^3 \frac{\tau}{T_{clk}} \qquad \text{(B.1)}$$

Table B–1: Short-circuit power estimates for a constant supply scenario.

	PFD		FDIV	
Tech. (um)	P $_{short\text{-}circuit}$ (W)	% of total Power	P $_{short\text{-}circuit}$ (W)	% of total Power
0.8	2.02E-05	12	2.03E-05	11
0.5	1.33E-05	11	1.33E-05	11
0.35	1.38E-05	14	1.38E-05	15
0.25	2.18E-05	22	2.19E-05	23

Appendix C

PLL DESIGN EXAMPLE

In order to better understand the trade-offs involved, the steps followed during the definition of the base PLL design are presented. First, it is crucial to guarantee the stability of the loop, which is achieved by guaranteeing the condition given in Eq. C.1.

$$\omega_n \le \frac{\omega_{lowest-operating-frequency}}{10} \tag{C.1}$$

Since the design is intended to be a frequency multiplier, the lowest operating frequency is clearly the input reference, which does not change and is very stable. Any possible changes occur in the feedback path of the loop when the division factor (N) is varied to provide a certain output frequency. Changes in N can be expressed as changes in the input (f_{ext}) as given below.

$$\Delta f_{ext} = f_{vco-old}\left(\frac{1}{N_{old}} - \frac{1}{N_{new}}\right) \tag{C.2}$$

The input reference frequency is 40MHz and a counter (divider) gives multiplication factors of 2, 4, 8 and 16. A non-differential 5-stage VCO is used (K_v = 390MHz/V), which provides the desired frequency range. Some of these requirements are similar to those in the clock design of the DEC SA-110 processor [42]. After replacing ω_n in the stability condition, Eq. C.3 is obtained. Note the dependence with N, which changes with the desired operating frequency. It is also of interest that C and I_{CH} have a linear relationship, which means that a smaller *C* (less area) will require a smaller I_{CH} (less power consumption).

$$C \ge \frac{100 K_V I_{CH}}{(2\pi \cdot 40MHz)^2 N} = 6.18\times10^{-7}\frac{I_{CH}}{N} \tag{C.3}$$

In order to minimize the overshoot, ξ is set to the optimum value (0.707), which, after some manipulation and using Eq. C.3, results in Eq. C.4.

$$RI_{CH} = \sqrt{\frac{4N^2\xi^2}{(6.2\times10^{-7})K_V}} = (0.09089)N \tag{C.4}$$

Where, again, the dependence on *N* is found. Using *N* = 2 for Eq. C.3 (worst case condition) and *N* = 4 for Eq. C.4, the following vales are obtained (I_{CH} = 50uA).

$$C = 15.5\,pF \qquad ; \qquad \mathrm{R} = 7271.20\,\Omega \tag{C.5}$$

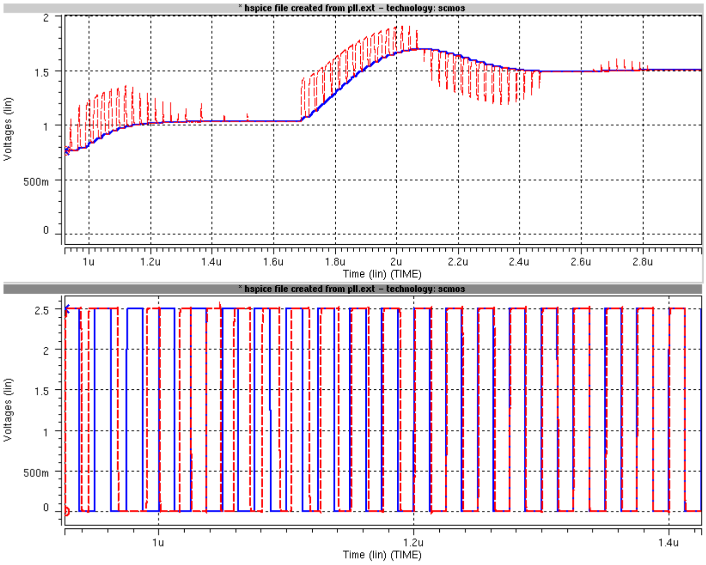


Figure C–1: Snapshot of the lock acquisition process.

To implement these elements, an area of approximately 3483.1um^2 is required for *C* while the resistor needs the following dimensions for the devices used: for the N transistor W = 6, L = 5 and for the P transistor W = 24, L = 5. The simulation confirms the correctness of the design, as shown in *Figure C–1*. The top plot shows the variation in the capacitor voltage as the divider factor *N* is changed first from 2 to 4 and then to 8. The control voltage is superimposed. It is clear how the damping factor (ξ) decreases with increasing *N*. The bottom plot shows details of the acquisition process.

Appendix D

PLL DESIGN ISSUES AND APPLICATIONS

In Chapter 3, it was found that ω_n and ξ depend on $(1/N)$ while K_V changes slightly with frequency and R might also do so. Then, it will be of interest to find a way to try to compensate for these variations so that the above parameters remain unaffected. One simple way to do this is by adjusting the charge pump current by an amount directly proportional to the change in the division factor. So, when N is doubled from 2 to 4, I_{CH} is also doubled to $2I_{CH}$ and some additional fine-tuning can also be included to compensate for some of the other non-linearities mentioned. The same control signals that determine the value of N can be used to change the charge pump bias reference voltage.

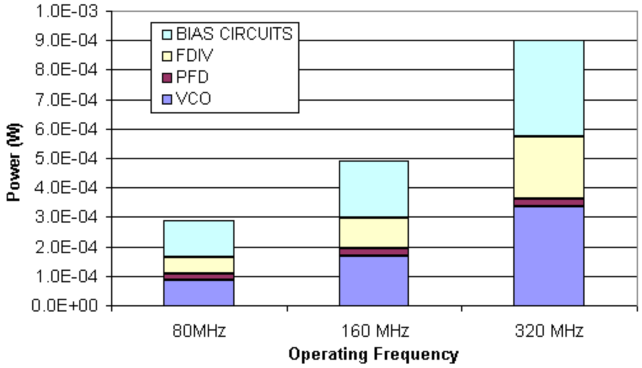


Figure D–1: Power consumption with adaptable I_{CH}.

Figure D–1 presents the power consumption breakdown if the above mentioned technique is used with the base design and I_{CH} takes a value of 12.5uA, 25uA or 50uA, following the frequency increase from 80, to 160 to 320MHz. Again, the price to pay for a constant acquisition time is a larger power consumption and this technique is only useful when the frequency change is determined by a change in *N*.

In addition to the use of the PLL as a frequency synthesizer for DVS systems, it could also be used for power supply regulation [40] and for leakage reduction by power supply gating. In this way, one single device can provide all the basic support required to implement a DVS scheme. The basic idea is to use the capacitor voltage (V_c) as a reference, as shown in *Figure D–2*, while a voltage follower provides the current required by the target load unit for normal operation. Since the VCO frequency tracks the variations in V_c, it is possible to couple these two variables in a way that, for the frequency currently being generated, the reference voltage equals the desired supply voltage for all the units that operate at the mentioned frequency.

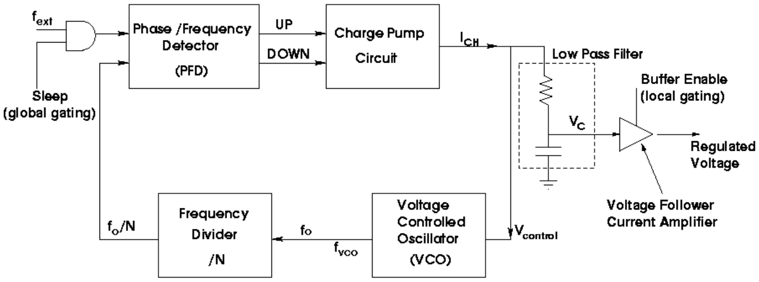


Figure D–2: The PLL as a voltage regulator.

Using such a scheme, leakage reduction is possible at various levels of granularity, as presented by *Figure D–3*. Consider a System-on-a-Chip (SOC) design in which many functional units are present. At a higher level, the supply voltage of each core can be shut down whenever the processing unit is not being used so that leakage can be minimized. At a lower level of granularity, the technique can also be applied inside a processing core so that the PLL remains active, while the individual buffers that provide power to functional units inside the core are disabled. In this way, additional leakage power savings are achieved by taking advantage of the idleness of certain components in the core. No significant power overhead is incurred since the PLL is shared among all the units and its overhead is calculated at the processing unit level. For further details, the reader is referred to [50].

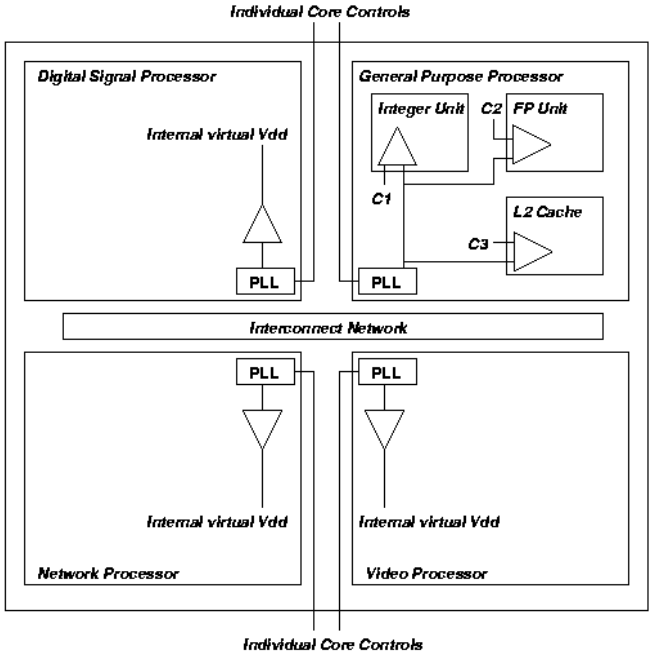


Figure D–3: A multi-level power supply gating scheme.

Appendix E

SCALING PARAMETERS

Table E–1 provides a brief summary of the scaling factors used in the experiments presented in Chapter 5, while *Table E–2* lists the key technology parameters calculated using the scaling factors mentioned. For further details, please refer to [53].

Table E–1: Summary of scaling parameters.

Scaling Parameter	Calculated as
S_L	$= L_{eff\_p}/L_{eff\_c}$
S_{tox}	$= S_L^{0.71}$
S_{Vdd}	$= S_L^{0.78}$
S_{sub}	$= S_L^{1.27}$
S_{vto}	$= S_L^{0.44}$
S_{vtdibl}	$= S_L^{0.52}$
S_{eta}	$= (S_{Vdd}* S_{tox})/(S_{vtdibl}*S_L^3)$

Table *E–2*: Scaling of technology parameters.

Technology Feature	Calculated as
Oxide Thickness	$t_{ox\_c} = \frac{t_{ox\_p}}{S_{tox}}$
Threshold voltage at V_{ds} = 0	$V_{th\_c} = \frac{V_{th\_p}}{S_{vto}}$
Bottom junction capacitance	$C_{j\_c} = C_{j\_p}\sqrt{S_{sub}}$
Sidewall junction capacitance	$C_{jsw\_c} = C_{jsw\_p}\frac{\sqrt{S_{sub}}}{S_L}$
DIBL factor	$\eta_{\_c} = \eta_{j\_p} S_{eta}$
Equivalent ΔV_{th} due to DIBL effects	$\Delta V_{th\_c} = \Delta V_{th\_p} S_{vtdibl}$
Power supply scaling	$V_{dd\_c} = \frac{V_{dd\_p}}{S_{Vdd}}$

Appendix F

LEAKAGE-AWARE CLOCK BUFFER DESIGN

Techniques like stack forcing [68] can be used to reduce leakage in buffers that drive clock signals (buffers that drive I/O signals can also be optimized). The idea is to prevent the input capacitive load from changing while the devices are stacked as shown in *Figure F–1*.

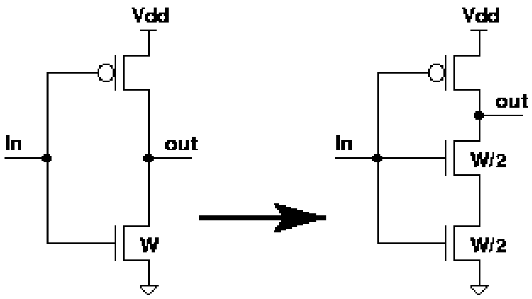


Figure F–1: The principle of stack forcing

If symmetric rising and falling times are needed, both P and N devices should be stacked. This obviously has a negative impact on the timing characteristics of the buffer and the design equations need to be modified to account for this. By halving the transistor width, the current driving capability is also halved, which means that the effective increasing factor per stage reduces from u to (u/2). Being that x is the ratio of the output to the input capacitance in the chain (i.e., $x = C_{out}/C_{in}$), the required number of inverter stages changes to Eq. F.1.

$$N_{new} = \frac{\ln(x)}{\ln(u/2)} = \frac{\ln(x)}{\ln(u) - \ln(2)} = N_{old}\,\frac{\ln(u)}{\ln(u) - \ln(2)} \qquad \text{(F.1)}$$

Since u will normally be larger than 2, this implies an increase in the number of stages N_{new}, from the case where stack forcing is not used (N_{old}). For the optimum case of u=3, stack forcing translates into a 171% increase in N (almost a tripling). This clearly emphasizes the importance of balancing the possible savings in leakage power with the incurred overheads in dynamic power.

Appendix G

COPYRIGHT INFORMATION

Some of the material in this thesis has already been partially published, although in no case, the presented material exactly resembles any of the manuscripts listed below. In order to comply with the requirements of the publishing entity, a copy of the IEEE copyright form is included, from which it is clear that the author can freely reuse all or portions of the published work, but third parties are required to obtain clearance from the IEEE Copyrights Permissions Office for reprinting, republishing, or other types of re-use.

Publications (or manuscripts expected to be published soon) from which substantial portions were extracted for use in this thesis are listed next:

- Duarte D., Vijaykrishnan, N. and Irwin, M.J., "*Impact of Technology Scaling in the Clock System Power*", To appear in the proceedings IEEE Annual Symposium on VLSI (ISVLSI), April 2002.
- Duarte D., Vijaykrishnan, N. and Irwin, M.J., "*A Complete PLL Power Consumption Model"*, To appear in the proceedings of the Conference on Design, Automation and Test in Europe (DATE), March 2002.
- Duarte D., Vijaykrishnan, N. and Irwin, M.J., "*Power, Energy and Timing Characterization of VLSI Charge-Pump Phase-Locked Loops"*, Submitted for publication in the IEEE Transactions on Circuits and Systems (CAS)– Part I.
- Duarte D., Tsai, Y, Vijaykrishnan, N. and Irwin, M.J., "*Evaluating Run-time Techniques for Leakage Power Reduction"*, Proceedings of the 7th ASP-DAC / 15th International Conference on VLSI Design, pp. 31-38, Jan. 2002.

- Duarte D., Vijaykrishnan, N. and Irwin, M.J., "*A Clock Power Model to Evaluate Impact of Architectural and Technology Optimizations"*, Submitted for publication in the IEEE Transactions on Very Large Scale Integration (VLSI) Systems.
- Duarte, D., Vijaykrishnan, N., Irwin, M.J. and Kandemir M., "*Evaluating the Impact of Architectural-level Optimizations on Clock Power*", Proceedings of the 14th IEEE ASIC/SOC Conference, pp.447-451, Sept. 2001.
- Vijaykrishnan, N., Kandemir, M., Irwin, M.J., Kim, H.S., Ye, W. and Duarte D., "*Evaluating Integrated Hardware-Software Optimizations using a Unified Energy Estimation Framework"*, To appear in the IEEE Transactions on Computers.
- Duarte, D., Vijaykrishnan, N., Irwin, M.J. and Kandemir M., "*Formulation and Validation of an Energy Dissipation Model for the Clock Generation Circuitry and Distribution Networks*", Proceedings of the 14th International Conference on VLSI Design, pp. 248-253, January 2001.
- Duarte, D., Irwin, M.J. and Vijaykrishnan, N., "*Modeling Energy of the Clock Generation and Distribution Circuitry*", Proceedings of the 13th Annual IEEE ASIC/SOC Conference, pp. 261-265, Sept. 2000.

Other published work that is only marginally used and related to this work, includes:

- Zhang W., Vijaykrishnan, N., Kandemir, M., Irwin, M.J., Duarte, D. and Tsai, Y., "*Exploiting VLIW Schedule Slacks for Dynamic and Leakage Energy Reduction*", Proceedings of the 34th Annual International Symposium on Microarchitecture (MICRO'01), December 2001.
- Hezavei, J., Vijaykrishnan, N., Irwin, M.J., Kandemir, M. and Duarte, D., "*Input Sensitive High-level Power Analysis*", Proceedings of the 2001 IEEE Workshop on Signal Processing Systems (SiPS), pp. 149-156, Sept. 2001.
- Duarte D., Hezavei, J. and Irwin, M.J., "*Power Consumption & Performance Comparative Study Of Logarithmic-time CMOS Adders*", Proceedings of the 2000 IEEE Workshop on Signal Processing Systems (SiPS), pp. 467-476, October 2000.

IEEE COPYRIGHT FORM

The IEEE has developed this form with great care and with the best interests of its members and contributing authors in mind. Therefore, in order to maintain uniform treatment among all contributors, other forms may not be substituted for this form, nor may any wording of this form be changed. This form is intended for original, previously unpublished material submitted to IEEE publications. This form, when completed, must accompany any such material in order to be published by IEEE. Please read it carefully and keep a copy of it for your files.

TITLE OF PAPER (hereinafter, "the work"):

AUTHOR(S):

PUBLICATION TITLE/DATE:

PART A—COPYRIGHT TRANSFER FORM

(U.S. Government employees whose work is not subject to U.S. copyright should so certify by signing Part B below. Authors of works subject to Crown Copyright should sign Part C below)

The undersigned hereby assigns all copyright rights in and to the above work to The Institute of Electrical and Electronics Engineers, Inc. (the "IEEE"). The undersigned hereby represents and warrants that the work is original and that he/she is the author of the work, except possibly for material such as text passages, figures, and data that clearly identify the original source, with permission notices from the copyright owners where required. The undersigned represents that he/she has the power and authority to make and execute this assignment.

In return for these rights, the IEEE recognizes the retained rights noted in items 1 and 4 below, and grants to the above authors and employers for whom the work may have been performed a royalty-free license to use the material as noted in items 2 and 3. Item 5 stipulates that authors and employers must seek permission to republish in cases not covered by items 2, 3, and 4.

1. Employers (or authors) retain all proprietary rights in any process, procedure, or article of manufacture described in the work.
2. Authors/employers may reproduce or authorize others to reproduce the above work, material extracted verbatim from the above work, or derivative works for the author's personal use or for company use provided that the source and the IEEE copyright notice are indicated, that the copies are not used in any way that implies IEEE endorsement of a product or service of an employer, and that the copies themselves are not offered for sale. (See "Author/Employer Rights" below.)
3. Authors/employers may make limited distribution of all or portions of the above work prior to publication if they inform the IEEE of the nature and extent of such limited distribution prior thereto.
4. In the case of work performed under a U.S. Government contract or grant, IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the above work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires. (Appropriate documentation may be attached, but IEEE's Copyright Form MUST BE SIGNED. See "U.S. Government Employees/U.S. Government Contract Work" below.)
5. For all circumstances not covered by items 2, 3, and 4, authors/employers must request permission from the IEEE Copyrights Office to reproduce or authorize the reproduction of the work or material extracted verbatim from the work, including figures and tables.

Please see notes on "IEEE Obligations" as copyright holder.

In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, this agreement becomes null and void.

AUTHORIZED SIGNATURE (or if joint work, as AGENT for all authors)	TITLE
EMPLOYER FOR WHOM WORK WAS PERFORMED	DATE

JOINT AUTHORSHIP

For jointly authored works, all the joint authors should sign, or one of the authors should sign as an authorized agent for the others. In the case of multiple authorship where one or more authors are Government employees but at least one author is not, that non-Government author should sign Part A of this copyright transfer form.

PLEASE DIRECT ALL QUESTIONS ABOUT IEEE COPYRIGHT POLICY OR THIS FORM TO: Manager, Copyrights and Trademarks, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. Telephone: (908) 562-3966.

PART B—U.S. GOVERNMENT EMPLOYEE CERTIFICATION

Authors who are U.S. Government employees are not required to sign Part A of the IEEE Copyright Form, but any coauthors outside the U.S. Government are required to sign Part A (see JOINT AUTHORSHIP overleaf). Authors whose work was performed under a U.S. Government contract or grant, but who are not Government employees, are required to sign Part A of this form (Note: If your work was performed under Government contract but you are not a Government employee, sign transfer form (Part A) and see Item 4.).

This will certify that all authors of the above work are employees of the U.S. Government and performed this work as part of their official duties and that the work is therefore not subject to U.S. copyright protection.

AUTHORIZED SIGNATURE

TITLE

NAME OF GOVERNMENT ORGANIZATION

DATE

PART C—CROWN COPYRIGHT

Authors who are employees of the British Government (or a British Commonwealth Government) and whose works are subject to Crown Copyright may sign Part C (below). IEEE recognizes and will honor Crown Copyright as it does U.S. Copyright. It is understood that, in asserting Crown Copyright, IEEE in no way diminishes its rights as publisher. Sign only if ALL authors are subject to Crown Copyright.

Signing Part C will certify that all authors of the (above) work are subject to Crown Copyright. (Appropriate documentation and instructions regarding wording of Crown Copyright notice may be attached.)

AUTHORIZED SIGNATURE

TITLE

NAME OF GOVERNMENT ORGANIZATION

DATE

Notes and Information for Authors and Their Employers

IEEE POLICY

In connection with its publishing activities, it is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums and similar publications, etc When an article is submitted to the IEEE for publication, the IEEE understands that its acceptance of the article implies that IEEE has the rights to do all of the things it normally does with such an article

IEEE Policy 6.17—CLEARANCE OF PAPERS—applies to all material submitted to IEEE: "The IEEE must of necessity assume that material presented at its meetings or submitted to its publications is properly available for general dissemination to the audiences these activities are organized to serve It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it "

Furthermore, if an author uses within his/her article previously published and/or copyrighted material, the IEEE must assume that permission has been obtained for such use and that any required credit lines, copyright notices, etc are duly noted

IEEE OBLIGATIONS

In exercising its rights under copyright, the IEEE will make all reasonable efforts to act in the interests of the authors and employers as well as in its own interest In handling third party reprint/republication requests for an IEEE work, the IEEE requires that 1) the consent of the first-named author be sought as a condition in granting republication (of a full paper) to others 2) The consent of the employer be obtained as a condition in granting permission to others to re-use all or portions of a paper for promotion or marketing purposes

AUTHOR/COMPANY RIGHTS

If you are employed and you prepared your paper as a part of your job, the rights to your paper rest initially with your employer In that case, when you sign the copyright transfer form, we assume you are authorized to do so by your employer and that your employer has consented to all the terms and conditions of this form If not, it should be signed by someone so authorized (See also Policy 6 17 above)

SPECIAL NOTE TO EMPLOYERS: Just as the IEEE requires a signed copyright transfer form (for copyrightable material) in order to do "business as usual," it is the intent of the transfer portion of the form to return rights to the author and employer so that they, too, may do "business as usual"

Please note that, although authors are permitted to re-use all or portions of their IEEE-copyrighted material in other works, this does not include granting third party requests for reprinting, republishing, or other types of re-use All third party requests must be handled by the IEEE Copyrights Permissions Office

BIBLIOGRAPHY

1. Stanford University's VLSI Research Group, Microprocessors Through the Ages. http://mos.stanford.edu/group/chips_micropro.html, 2001.

2. Chandrakasan, A. and Brodersen R., Low Power Digital CMOS Design, pp. 56-103, Kluwer Academic Publishers, Boston, MA, 1995.

3. Keshavarzi, A., Roy, K. and Hawkins, C.F., Intrinsic Leakage in Low Power Deep Submicron CMOS IC, Proceedings of the 1997 International Test Conference, pp. 146-155.

4. Rabaey, J. and Pedram, M., Low Power Design Methodologies, Kluwer Academic Publishers, Boston, MA, 1996.

5. Gronowski, P.E., et al., High-Performance Microprocessor Design, IEEE Journal of Solid State Circuits, Vol. 33, No. 5, pp. 676-686, May 1998.

6. Tiwari, V., et al., Reducing Power in High-Performance Microprocessors, Proceedings of the 35th DAC, June 1998.

7. Friedman, E., Clock Distribution Design in VLSI Circuits: an Overview, Proceedings of the IEEE ISCAS, May 1994, pp. 1475-1478.

8. Restle, P.J. and Deutsch, A., Designing the Best Clock Distribution Network, IEEE Symposium on VLSI Circuits – Digest of Technical Papers, 1998, pp. 2-5.

9. Kourtev, I.S. and Friedman, E.G., Simultaneous Clock Scheduling and Buffered Clock Tree Synthesis, Proceedings of the 1997 IEEE ISCAS'97, June 1997, pp. 1812-1815.

10. Maaz, M.B. and Bayoumi, M.A., A non-zero clock skew scheduling algorithm for high speed clock distribution network, Proceedings of the 1999 ISCAS'99, June 1999, pp. 382-385.

11. Neves, J.L. and Friedman, E.G., "Design methodology for synthesizing clock distribution networks exploiting nonzero localized clock skew", IEEE Transactions on VLSI, Vol. 42, June 1996, pp. 286 –291.

12. Ting-Hai Chao, Yu-Chin Hsu, Jan-Ming Ho and Kahng, A.B, Zero skew clock routing with minimum wirelength, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 39 11, Nov. 1992, pp. 799 –814.

13. Pullela, S., Menezes, N. and Pillage, L.T., Low power IC clock tree design, Proceedings of the IEEE 1995 Custom Integrated Circuits Conference, May 1995, pp. 263 –266.

14. Tellez, E., Farrah, A. and Sarrafzadeh, M., Activity-driven Clock Design for Low Power Circuits, Proceedings of the ICCAD, Nov. 1995, pp. 62-65.

15. Kahng, A.B. and Tsao, C.W.A., More Practical Bounded-Skew Clock Routing, Proceedings of the 34th DAC, June 1997, pp. 594-599.

16. Ellis, G., Pileggi, L.T. and Rutenbar, R.A., A hierarchical decomposition methodology for single-stage clock circuits, Proceedings of the IEEE 1997 Custom Integrated Circuits Conference, May 1997, pp. 115 –118.

17. Mizuno, H. and Ishibashi, K., A Noise-Immune GHz-Clock Distribution Scheme using Synchronous Distributed Oscillators, The 1998 IEEE International Solid-State Circuits Conference, 1998, pp. 405-406.

18. Chen, R.Y., Vijaykrishnan, N. and Irwin, M.J., Clock Power Issues in SOC Designs, Proceedings of the IEEE Computer Society Workshop on VLSI, April 1999, pp. 48-53.

19. Rabaey, J., Digital Integrated Circuits: A Design Perspective, Prentice-Hall International, NJ, 1996.

20. Wang X. and Porod, W., A low power charge-recycling CMOS clock buffer, Proceedings of the Ninth Great Lakes Symposium on VLSI, March 1999, pp 238 –239.

21. Khoo K.Y., and Willson, A.N. Jr., Low power CMOS clock buffer, Proceedings of the 1994 IEEE International Symposium on Computer Architecture – ISCAS'94, June 1994, pp. 355-358.

22. Cho, J. and Sarrafzadeh, M., A buffer distribution algorithm for high-performance clock net optimization, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 31, March 1995, pp. 84 –98.

23. Restle, P., et al., A Clock Distribution Network for Microprocessors, IEEE Journal on Solid-State Circuits, Vol.36, No. 5, May 2001, pp. 792-799.

24. Liu D. and Svensson C., Power Consumption in CMOS VLSI Chips, ", IEEE Journal of Solid State Circuits, Vol. 29, No. 6, pp. 663-670, June 1994.

25. Ye, W, Vijaykrishnan, N., Kandemir, M., and Irwin, M. J., The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool, Proceedings of the 37th DAC, June 2000.

26. Brooks, D., Tiwari, V., and Martonosi, M., Wattch: A Framework for Architectural Level Power Analysis and Optimization, Proceedings of the 27th IEEE International Symposium on Computer Architecture, June 2000, pp. 83-94.

27. Kamble, M., and Ghose, K., Energy-Efficiency of VLSI Caches: A Comparative Study, Proceedings of the 10th International Conference on VLSI Design, January 1997, pp. 261-267.

28. Zyuban, V., and Kogge, P., The Energy Complexity of Register Files, Proceedings of International Symposium on Low Power Electronics and Design, (ISLPED), 1998, pp. 305-310.

29. Juan, T., Lang., T., and Navarro, J., Reducing the TLB Power Requirements, Proceedings of the 1997 International Symposium on Low Power Electronics and Design, August 1997, pp.196-201.

30. Jeong, D.Y., Borriello, G., Hodges, D.A. and Katz, R.H., Design of PLL-Based Clock Generation Circuits, IEEE Journal on Solid-State Circuits, Vol. 22, No. 2, April 1987, pp. 255-261.

31. Tourette, B., A 200 MHz, 600uW CMOS PLL for Mobile Communication Asics, Proceedings of the Twelfth Annual IEEE International ASIC/SOC Conference, September 1999, pp. 366-369.

32. Weste, N., and Eshraghian, K., Principles of CMOS VLSI Design, Addison-Wesley Publishing Company, MA, 1994, pp. 180-191.

33. Duarte, D., Irwin, M.J. and Narayanan, V., Modeling Energy of the Clock Generation and Distribution Circuitry, Proceedings of the 13th Annual IEEE International ASIC/SOC Conference, September 2000, pp. 261-265.

34. Duarte, D., Narayanan, V., Irwin, M.J. and Kandemir M., Formulation and Validation of an Energy Dissipation Model for the Clock Generation Circuitry and Distribution Network, Proceedings of the 14th International Conference on VLSI Design, January 2001, pp. 248-253.

35. Burger, D. and Austin, T., The SimpleScalar Tool Set, Version 2.0, University of Wisconsin-Madison, Computer Sciences Department Technical Report #1342, June 1997.

36. Ye, W. and Irwin, M.J., Power Analysis of Gated Pipeline Registers, Proceedings of the 12th IEEE International ASIC/SOC Conference, September 1999, pp. 281-285

37. Halfhill, T., Top PC Vendors Adopt Crusoe, Microprocessor Report, Vol. 14, Archive 7, July 2000.

38. Leibson, S., Xscale (StrongARM-2) Muscles In, Microprocessor Report, Vol. 14, Archive 9, Sept. 2000.

39. Burd, T., et al., A Dynamic Voltage Scaled Microprocessor System, IEEE Journal on Solid-State Circuits, Vol. 35, No. 11, November 2000, pp. 1571-1580.

40. Von Kaenel, V., et al., A Voltage Reduction Technique for Battery-Operated Systems, IEEE Journal on Solid-State Circuits, Vol. 25, No. 5, October 1990, pp. 1136-1140.

41. Young, I., et al., "A PLL Clock Generator with 5 to 110MHz of lock range for Microprocessors, IEEE Journal on Solid-State Circuits, Vol. 27, No. 11, November 1992, pp. 1599-1607.

42. Von Kaenel, V., et al., A 320MHz, 1.5mW @ 1.35V CMOS PLL for Microprocessor Clock Generation, IEEE Journal on Solid-State Circuits, Vol. 31, No. 11, November 1996, pp. 1715-1722.

43. Razavi, B., Monolithic Phase-Locked Loops and Clock Recovery Circuits, 1st Ed. IEEE Press, 1995.

44. Maneatis, J., Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques, IEEE Journal on Solid-State Circuits, Vol. 31, No. 11, November 1996, pp. 1723-1732.

45. Gardner, F., Charge-Pump Phase-Lock Loops, IEEE Transactions on Communications, Vol. 28, November 1980, pp. 1849-1858.

46. Van Paemel, M., Analysis of a Charge-Pump PLL: A New Model, IEEE Transactions on Communications, Vol. 42, No. 7, July 1994, pp. 2490-2498.

47. Best, R., Phase-Locked Loops: Design, Simulation and Applications, 4th Ed., Mc Graw-Hill, 1999.

48. Duarte, D., et al., A Clock Power Model to Evaluate Impact of Architectural and Technology Optimizations, CSE Technical Report (available on the web).

49. Rhee, W., Design of High-Performance CMOS Charge Pumps in Phase-Locked Loops, Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, pp. 545-548.

50. Duarte, D., et al, Evaluating Run-Time Techniques for Leakage Power Reduction, Proceedings of the 7th ASP-DAC / 15th International Conference on VLSI Design, pp. 31-38, Jan. 2002.

51. Scott, J., et al., Designing the Low-Power MCORE Architecture, at http://www.mot.com/SPS/MCORE/info_documentation.htm, Motorola's Online Documentation Library.

52. Wolfe, M., High Performance Compilers for Parallel Computing. 1st Ed., Addison-Wesley, 1996.

53. Mc. Farlan, G., CMOS Technology Scaling and Its Impact on Cache Delay, PhD. Thesis, Stanford University, June1997, http://umunhum.stanford.edu/~farland/.

54. Butts, J. and Sohi, G., A Static Power Model for Architects, Proceedings of the 33rd Annual IEEE International Symposium on Microarchitecture – MICRO 2000, pp. 223-234.

55. Streetman, G., Solid State Electronics, 4th Ed., Prentice-Hall Editors, 1995..

56. Chandrakasan, A., Bowhill, W. and Fox, F., Design of High Performance Microprocessor Circuits. 1st Ed., IEEE Press. 2000.

57. Gupta, S. and Najm, F.N, Power Modeling for High-level Power Estimation, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 8, No. 1, Feb. 2000, pp. 18 –29.

58. Landman, P.E. and Rabaey, J.M., Architectural Power Analysis: The Dual Bit Type Method, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 3, No. 2, June 1995, pp. 173 –187.

59. Nemani, M. and Najm, F.N., Towards a High-level Power Estimation Capability [Digital ICs], IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 15, No. 6, June 1996, pp. 588 –598.

60. Mehta, H., Owens, R.M. and Irwin, M.J., Energy Characterization Based on Clustering, Proceedings of the 33rd Design Automation Conference, June 1996, pp. 702 – 707.

61. Lin, J., Shen, W. and Jou, J., A Power Modeling and Characterization Method for Macrocells Using Structure Information, IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, 1997, pp. 502 –506.

62. Landman, P.E., High-level Power Estimation, Proceedings of the International Symposium on Low Power Electronics and Design, (ISLPED), 1996, pp. 29.

63. Frenkil, J., Issues and Directions in Low Power Design Tools: An Industry Perspective, Proceedings of the International Symposium on Low Power Electronics and Design, (ISLPED), 1997, pp. 152.

64. Katkoori, S. and Vemuri R., Architectural Power Estimation Based on Behavioral Profiling, Journal on VLSI Design, Special Issue on Low Power Design, 1996.

65. Sakurai, T., Simple Formulas for Two and Three Dimensional Capacitances, IEEE Transactions on Electron Devices, 1983, p. 183.

66. Chen, Z., Diaz, C., et al., 0.18um Dual Vt MOSFET Process and Energy-delay Measurement, International Electronic Devices Meeting, 1996, pp.851-854.

67. Borkar, S., Design Challenges of Technology Scaling, IEEE Micro, July-August 1999, pp. 23-29.

68. Narendra, S., et al., Scaling of Stack Effect and Its Application for Leakage Reduction, Proceedings of the 2001 International Symposium on Low Power Electronics and Design.

69. Grotjohn, T. and Hoefflinger, B., A Parametric Short-channel CMOS Transistor Model for Subthreshold and Strong Inversion Currents, IEEE Transactions on Electron Devices, 1984, pp. 234-246.

70. Yoshida, T., Quantitative Analysis of Tunneling Current Through Ultra-thin Gate Oxides, Japanese Journal of Applied Physics, July 1995, pp. L903-L906.

71. Intel Corporation, Intel Announces Breakthrough In Chip Transistor Design, http://www.intel.com/pressroom/archive/releases/20011126tech.htm.

72. Hu, C., Gate Oxide Scaling Limits and Projection, IEDM, 1996, pp. 319-322.

73. Lo, S-H., et al., Quantum-Mechanical Modeling of Electron Tunneling Current from the Inversion Layer of Ultra-Thin-Oxide nMOSFETs , IEEE Electron Device Letters, Vol. 18, No. 5, May 1997, pp. 209-211.

74. Lo, S-H., et al., Modeling and Characterization of Quantization, Polysilicon Depletion and Direct Tunneling Effects in MOSFETs with Ultrathin Oxides, IBM Journal of Research and Development, Vol. 43, No. 3, May 1999, pp. 327-337.

VITA

David Enrique Duarte

David E. Duarte received the title of Electronics Engineer from the Pontificia Universidad Javeriana, Bogota, Colombia in October 1996 after completing five years of coursework and the writing of a thesis. In May 1999, he received an MS in Electrical Engineering from The Pennsylvania State University, University Park, PA. His MS research dealt with illumination and rotation invariant face recognition techniques.

During his undergraduate studies, Mr. Duarte held various teaching assistant appointments in the Departments of Mathematics, Physics and Electronics Engineering. In January 1996, while working on his undergraduate thesis, he was awarded a part-time instructor appointment for two courses at the Pontificia Universidad Javeriana, which he held until May 1997. In April 1996 he began working for the largest Colombian telecommunications company (ITEC-TELECOM) as a research engineer. Since January 1998, he has been a graduate assistant at Penn State University, with duties as both a teaching and research assistant. He was a Graduate Technical Intern at the Circuit Research Laboratory, Intel Corporation, Hillsboro, OR during the summer of 2000, working on leakage estimation techniques and power delivery issues.

At the undergraduate commencement, Mr. Duarte was honored as the student with the best GPA and also as the best student of the second class of Electronic Engineers of 1996. The focus of his PhD work has been digital (and analog) circuit and system design, including, but not restricted to, low power VLSI design and processor architecture. He is a member of the IEEE. For a complete list of publications please refer to his website at http://www.cse.psu.edu/~duarte.