A Static Power Model for Architects

J. Adam Butts and Gurindar S. Sohi
Computer Science Department
University of Wisconsin-Madison
{butts,sohi}@cs.wisc.edu

Abstract

Static power dissipation due to transistor leakage constitutes an increasing fraction of the total power in modern semiconductor technologies. Current technology trends indicate that the contribution will increase rapidly, reaching one half of total power dissipation within three process generations. Developing power efficient products will require consideration of static power in the earliest phases of design, including architecture and microarchitecture definition. We propose a simple equation for estimating static power consumption at the architectural level: $P_{static} = V_{CC} \cdot N \cdot k_{design} \cdot \hat{I}_{leak}$, *where V_{CC} is the supply voltage, N is the number of transistors, k_{design} is a design dependent parameter, and $\hat{I}_{leak}$ is a technology dependent parameter. This model enables high-level reasoning about the likely static power demands of alternative microarchitectures. Reasonably accurate values for the factors within the equation may be obtained directly from the high-level designs or by straightforward scaling arguments. The factors within the equation also suggest opportunities for static power optimization, including reducing the total number of devices, partitioning the design to allow for lower supply voltages or slower, less leaky transistors, turning off unused devices, favoring certain design styles, and favoring high bandwidth over low latency. Speculation is also examined as a means to employ slower transistors without a significant performance penalty.*

1. Introduction

Power consumption has become an important consideration in modern microprocessor design. The problem is exacerbated in multiprocessor systems such as servers in which multiple processors are in close proximity. Increasing the power dissipation much beyond current levels will result in disproportionate increases in cost as current power delivery and heat removal systems reach limits. Mobile and embedded microprocessors are also power constrained. While maximization of battery life is an obvious goal, heat removal is an important problem as well. The increasing role of power dissipation as a performance limiter has led to the consideration of power in the early stages of the design process. Traditionally the responsibility of circuit designers, power dissipation has become more important to architects as the ability of circuit techniques to control it have been rendered insufficient. The availability of simple estimation methods and the spread of simulators which provide power dissipation data have enabled power dissipation to influence high level design decisions.

Architectural efforts to control power dissipation have been directed primarily at the dynamic component of power dissipation. Dynamic power is the result of switching and is ideally the only mode of power dissipation in CMOS circuitry. It constitutes the major component of total power dissipation in today's technologies. Dynamic power dissipation is described by the familiar $P_{dyn} = CV_{CC}^2 f$ where C is the capacitance of switching nodes (roughly proportional to the number of switching devices), V_{CC} is the supply voltage, and *f* is the effective operating frequency (frequency times activity factor). In order to limit dynamic power dissipation, techniques such as clock gating [12, 31, 32], cache sub-banking [28], and eliminating needless computation [5, 19] have been employed. The goal of each of these techniques is to reduce the number or frequency of switching devices (attacking C or *f*, respectively). Optimization of the supply voltage to minimize the power/performance ratio is also performed, but this process is seldom influenced by architects.

As transistors become smaller and faster, another mode of power dissipation has become important. This is static power dissipation, or the power due to leakage current in the absence of any switching activity. Technology scaling is increasing both the absolute and relative contribution of static power dissipation. Static power dissipation is equal to the product of the supply voltage and the leakage current. While the rate of reduction of supply voltage is decreasing, leakage current is increasing exponentially.

The increasing contribution of static power is clearly evident even in today's designs. Consider two implementations of Intel's Pentium III processor manufactured on Intel's 0.18 µm process, the Pentium III 1.0 GHz B and the Pentium III 1.13 GHz [13]. The Intel datasheet lists the maximum core power dissipation of the 1.0 GHz part at 33.0 watts and the deep sleep (i.e., static) power dissipation at 3.74 watts. The 1.13 GHz processor has a total power dissipation of 41.4 watts and a static power dissipation of 5.40 watts. While the total power has increased by only 25%, the static power has increased by 44% and comprises 13% of the total power dissipation. The active power dissipation of the processor core varies significantly depending on the workload while the static power dissipation is almost constant. The datasheet values represent peak power dissipation values; therefore, static power is even a larger percentage of the total power dissipation on average.

Figure 1 shows the increases in static and dynamic power for Intel's past few technologies [34]. Projecting these trends forward, static power dissipation will equal dynamic power dissipation within a few generations. Higher order effects unimportant today and aggressive dynamic power optimizations could cause the static and dynamic power contributions to become equal in as little as two generations. Thus, it is important for architects to be aware of how they may control static power dissipation in future technologies.

The causes of leakage current are complex and far removed from the realm of architecture. Yet as static power dissipation becomes comparable to dynamic power dissipation, architects will be called upon to consider it in making design decisions. The purpose of this paper is to provide architects with a means

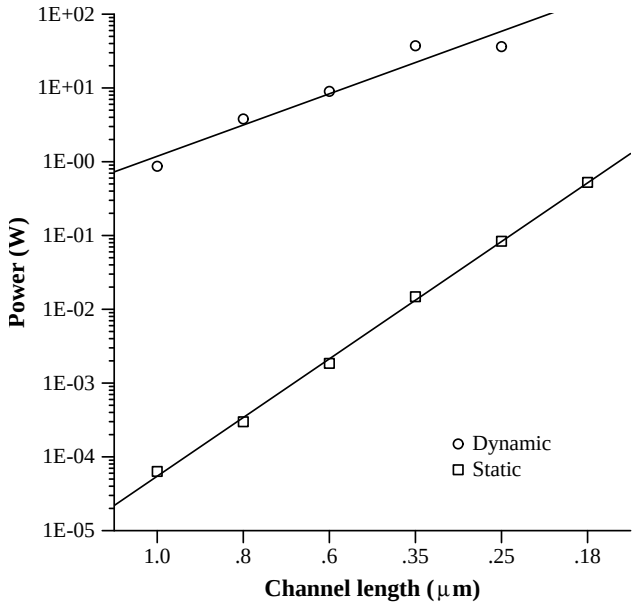

Figure 1. Trends in dynamic and static power dissipation showing increasing contribution of static power (from Thompson, *et. al.* [34])

of estimating static power and some general techniques for limiting it. We propose a simple four parameter model useful at the architectural level: $P_{static} = V_{CC} \cdot N \cdot k_{design} \cdot \hat{I}_{leak}$. The model parameters are summarized in Table 1. Overall static power consumption may be reduced by reducing any of the parameters. The table lists some general techniques applicable to reducing each parameter.

The level of abstraction in the model is appropriate for its application by architects. Each of the parameters is amenable to estimation at the architectural level (either based on the design or the expected target technology). A more detailed model would require accuracy in technology and design parameters that would not be available at an early stage in the design process. Furthermore, absolute accuracy is not as important as relative accuracy when making design tradeoffs. Finally, the model suggests different means of addressing static power early in the design process. Some may claim that architects have no control over static power because of its strong dependence on technology and circuit optimization (which does not typically involve architects). While lower level optimizations more directly affect the final static power dissipation, awareness of the issue during the architectural definition can result in an architecture better suited to later optimization.

We proceed with a brief review of semiconductor technology. Next, we motivate the increasing importance of static power with a discussion of trends in transistor scaling. The static power model above is then derived and the characteristics of each of the model parameters are discussed in detail. Finally, the model is used to motivate some general architectural-level techniques for addressing static power dissipation.

2. CMOS Technology Review

We start with a review of the basic terminology and operation of the silicon field-effect transistor. Silicon CMOS (Complementary Metal Oxide Semiconductor) has emerged as the dominant semiconductor technology for high performance microprocessors. Relative to other semiconductor technologies, silicon CMOS is cheaper, is more easily processed and scaled, and has a higher performance/power ratio. This section describes the important features of MOS transistors and introduces terminology used throughout the remainder of the paper. Readers familiar with this material are encouraged to skip to Section 3, while those desiring more detail may find it in any of several readily available texts from which this review was distilled [23, 30, 37].

A MOS transistor is a four terminal semiconductor device that can function as a switch or an amplifier (Figure 2). By convention, all terminal voltages are measured with respect to the source node. The gate voltage is symbolized by V_{gs}, the drain voltage by V_{ds} and the body voltage by V_{bs}. In digital circuit design, the transistor is usually used as a switch. Current flow between the source and drain terminals is controlled by the voltage at the gate terminal. The gate is electrically isolated from the rest of the device by a thin insulating layer (silicon dioxide for silicon devices). The gate influences the device via the elec-

Table 1. Summary of static power model parameters

Parameter	Description	Scaling behavior	Reducing
V_{CC}	Power supply voltage	Decreases by 30 % per process generation	• Multiple supply voltage domains • Increase IPC to allow lower clock frequency (allowing V_{CC} reduction) at same performance
N	Number of transistors in design	Increases by 100 % per process generation	• Reduce functionality (e.g., removing special purpose circuitry) • Use circuit style requiring fewer transistors for same functionality
k_{design}	Empirically determined parameter representing the characteristics of an average device	Approximately constant	• Use efficient circuit style • Reduce clock frequency to allow more complex (high fan-in) logic
$\hat{I}_{leak}$	Technology parameter describing the per device subthreshold leakage	Highly dependent on aggressiveness of V_T (threshold voltage) scaling	• Partition design into frequency domains allowing use of less aggressive (lower leakage) devices in some domains

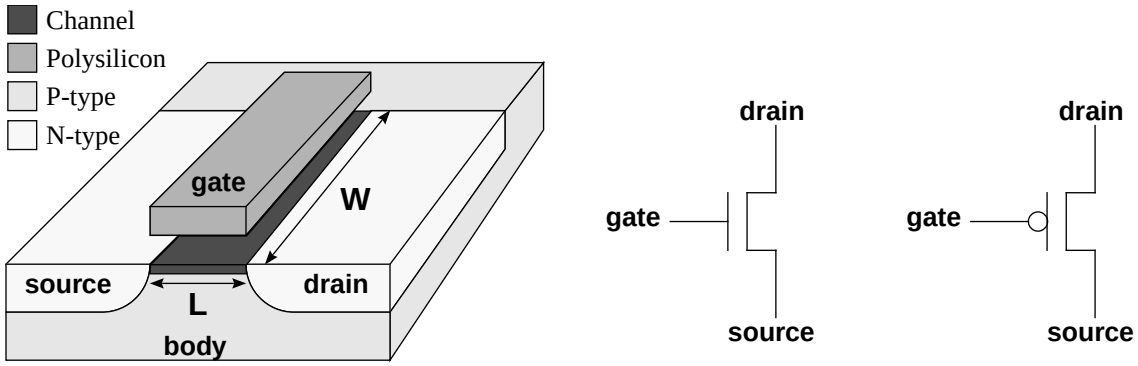

Figure 2. MOS transistor cross-section (N-type) and schematic symbols (N-type and P-type)

tric field resulting from different gate biases. Thus, the transistor is designated a Field Effect Transistor or FET.

The primary function of the body terminal is to ensure isolation of the source and drain. Impurities are added (a process called doping) to the source, drain, and body regions. The source and drain regions are doped to the opposite type as the body (N- or P-type), creating junctions through which current (ideally) can not flow. Under the influence of the gate, the type of the region at the surface of the silicon between the source and drain (called the channel) can be reversed, forming a current path between the source and drain. Since the gate is electrically insulated from the rest of the device, a transistor gate appears as a capacitor to its driving circuitry. Ideally, once the gate capacitor is charged (or discharged) to its desired state, no current is required to maintain that state; therefore, no power is consumed. The threshold voltage of the transistor (symbolized by V_T) is the voltage required at the gate (relative to the source) to turn on the transistor. It is a complicated function of the device dimensions and exact doping profiles of the transistor. N- and P- type transistors differ in the doping of the source, drain, and body regions (the Complementary in CMOS).

Most device parameters (e.g., doping profiles and oxide thickness) are fixed by the particular technology to which a design is targeted. In most cases circuit designers are limited to specifying the device dimensions (W and L) to specify the relative strengths of the devices. Some technologies provide devices with different threshold voltages as well. These technologies are referred to as MTCMOS (multi-threshold CMOS). Alternatively, the threshold voltage may be controlled by applying different voltages to the body terminal. Thus, the design parameters include the lateral device dimensions and sometimes the threshold voltage.

Power consumption in CMOS circuitry is classified as either dynamic or static (Figure 3). Dynamic power dissipation occurs during state changes (i.e., when devices are switching). It is primarily due to the charging of the capacitative load associated with the output wiring and the gates of subsequent transistors (C dV/dt). A smaller component of dynamic power arises from the short-circuit current that flows momentarily while the complementary devices in a gate are simultaneously conducting during an output state change. Static power dissipation is a result of the various leakage modes of the MOS transistor. While there are many different leakage modes, the most important leakage mechanism in modern submicron channel length technologies is subthreshold leakage [15]. Subthreshold leakage is current that flows between the source and drain even when the transistor is off (i.e., the voltage at the gate is below the threshold voltage).

3. Technology Scaling

To allow for higher clock frequencies and more devices on a chip, technologies are scaled every few years [27]. Device engineers performing the scaling must develop transistors years in advance of when they will be manufacturable. Using Moore's law as a guide, they target a 30% decrease in linear dimensions resulting in a 50% area reduction versus the prior generation. Simultaneously, the smaller dimensions allow for a speed increase of 25-30%. The primary constraint on device scaling is the process technology (e.g., lithography). Another important constraint is reliability. Many reliability parameters are functions of the electric fields that exist within the device. Permanent damage to the transistor may result if certain electric fields are exceeded. This has led to a scaling methodology known as constant field (sometimes called ideal) scaling [9].

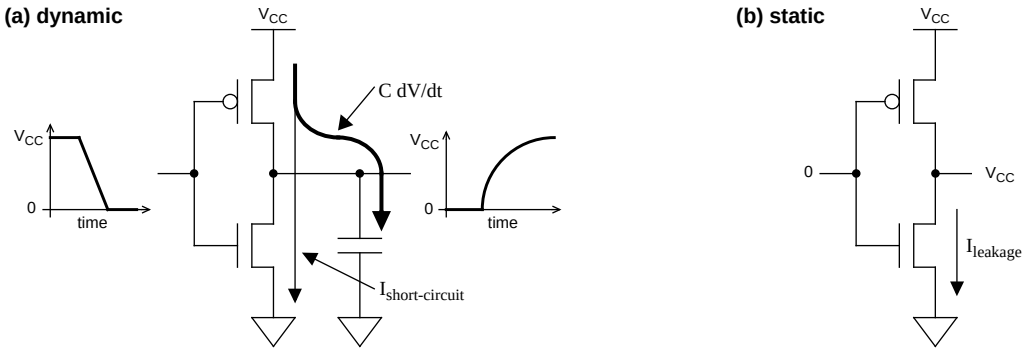

Figure 3. (a) Dynamic and (b) static power dissipation mechanisms in CMOS technologies

Constant field scaling reduces the supply voltage by the same factor as device dimensions in order to keep the electric fields the same across technology generations. This has the added benefit of addressing dynamic power dissipation (which is proportional to the square of the supply voltage). With the physical dimensions and supply voltage determined, device designers adjust other parameters (e.g., doping profiles) to maximize the performance of the device within the specified constraints. While actual technologies have not adhered strictly to constant-field scaling [7], it is illustrative of the general trends and problems associated with scaling.

Due to the complexities of device simulation, it is not practical to simulate even small circuits at the level of detail required by device engineers. Therefore, device engineers attempt to optimize simple delay metrics to arrive at a device design. These metrics may be calculated from the detailed simulation of a single transistor. After confirming the performance with actual fabricated test devices, parameters are derived for a device model that can be used in subsequent circuit-level simulations. One common delay metric used is shown in Equation 1. C_{gate} is the gate capacitance of a transistor per unit width (at a specified channel length), V_{CC} is the supply voltage, and I_{Dsat} is the maximum (saturation) drain current that can flow through a transistor (per unit width). Derived from the differential equation describing the charging of a capacitor, this metric measures the approximate time required to charge the gate capacitance of one transistor by another transistor.

$$t = \frac{C_{gate} \cdot V_{CC}}{I_{Dsat}} \qquad \text{(Eq. 1)}$$

Consider the behavior of the delay metric of Equation 1 under constant field scaling. The supply voltage (V_{CC}) is reduced by some factor S. Therefore to reduce delay by the same factor, it is sufficient to keep the ratio C_{gate} / I_{Dsat} constant. C_{gate} is proportional to the channel length and inversely proportional to the oxide thickness. Since both of these dimensions are reduced by S, C_{gate} stays constant. Thus, to achieve the expected performance improvement (delay reduction), the drive current I_{Dsat} must remain constant under scaling. In modern technologies, I_{Dsat} is a complicated function of many parameters including $V_{CC} - V_T$, C_{gate}, and L (the channel length).

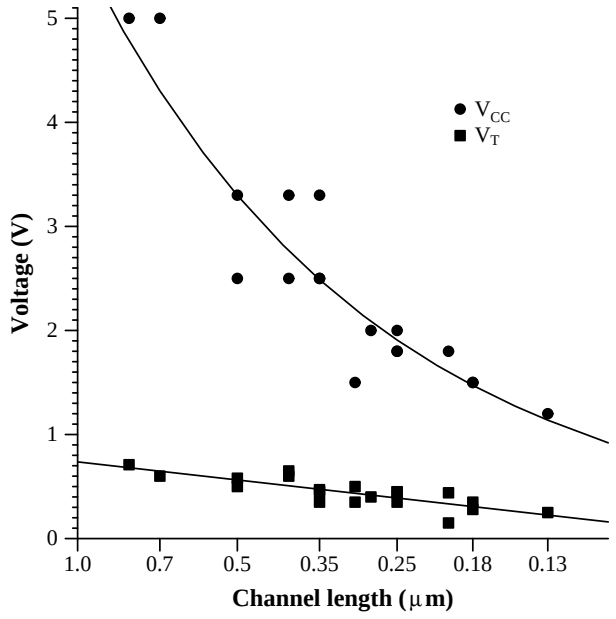

Figure 4. V_{CC} and V_T scaling showing reduction in gate overdrive ($V_{CC} - V_T$) (from data published in IEDM and ISSCC from 1990-2000)

The quantity $V_{CC} - V_T$ is referred to as the gate overdrive; it is the maximum voltage that may be applied to a transistor's gate beyond that required to turn on the transistor. I_{Dsat} is proportional to a small power (between 1 and 2) of $V_{CC} - V_T$ [26]. Recalling that V_{CC} is being decreased by S, the reduction in gate overdrive reduces I_{Dsat} by a factor larger than S. While other factors increase the drive current as devices are scaled (primarily L), these are insufficient to obtain the expected delay reduction at a constant V_T in deep submicron CMOS technologies. Therefore, V_T has also been reduced (see Figure 4). Performance goals and a desire to decrease V_{CC} further (to address dynamic power) have also driven the reduction in threshold voltage.

It is this continuing reduction of V_T that is causing static power to become increasingly important. Subthreshold leakage current increases exponentially as threshold voltage decreases [12]:

$$I_{Dsub} = k \cdot e^{\frac{-q \cdot V_T}{a \cdot k_B \cdot T}} \qquad \text{(Eq. 2)}$$

where q and k_B are physical constants, a and k are device parameters, and T is the absolute temperature. The above relationship is depicted in Figure 5 (V_T is taken to be the gate voltage at 1 μA/μm drain current). Note that the leakage current at a fixed threshold voltage also increases exponentially with temperature.

Static power is equal to the product of the supply voltage and I_{Dsub}. The exponential increase in I_{Dsub} causes the static power to increase rapidly despite supply voltage scaling. The relative contribution of static power is also growing. Dynamic power increases linearly with the capacitance being switched (increas-

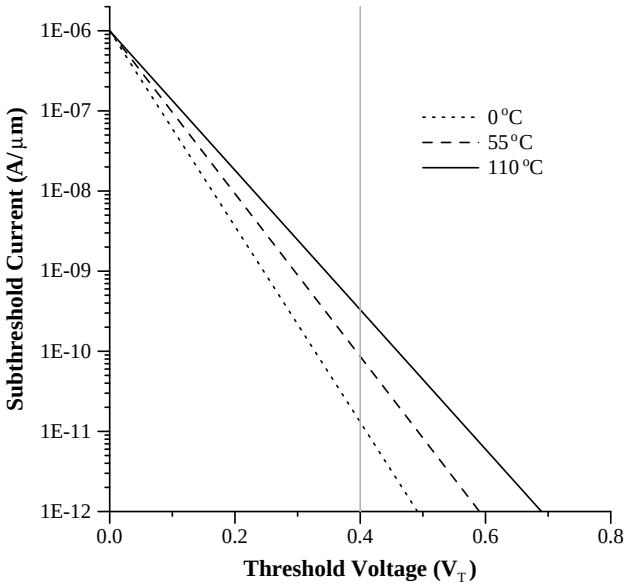

Figure 5. Effect of threshold voltage and temperature on subthreshold current

ing as the number of devices is increased) and the switching frequency (increasing as delay is reduced), but decreases with the square of the supply voltage. Thus, it is increasing much more slowly than static power (refer to Figure 1). As the primary component of power consumption today, dynamic power is being aggressively attacked in all phases of the design process to ensure that it does not restrict performance. Focusing on limiting dynamic power further increases the relative importance of static power.

4. A Static Power Model

While accurate power models are important for simulation, it is desirable to have a simple formula to allow for high-level consideration of the power characteristics of alternative designs. The absolute accuracy of such a formula is not nearly as important as the relative accuracy since the architect will generally be uninterested in determining the exact number of watts used by a particular design. In this section, we will present a formula that is a useful high-level model of static power consumption. Each of the model parameters discussed in detail with emphasis on how it scales and how it may be estimated.

4.1. Model Derivation

In this section, we derive the static power model presented in the introduction. The dearth of publicly available data on leading-edge microprocessors makes it difficult to compare the model's results with actual data. Thus, a top-down, intuitive derivation would be almost impossible to validate. Therefore, we chose a bottom-up derivation based on a widely accepted single-device model. It should be noted that successful application of the model does not depend on the material in this section. Instead, the derivation is presented to make explicit the simplifying assumptions necessary to arrive at a high-level model from the detailed device-level equation.

We begin with the BSIM3v3.2 MOSFET transistor model equation for subthreshold drain current I_{Dsub} [17]:

$$I_{Dsub} = I_{s0} \cdot \left(1 - e^{\frac{-V_{ds}}{v_t}}\right) \cdot e^{\frac{V_{gs} - V_T - V_{off}}{n \cdot v_t}} \qquad \text{(Eq. 3)}$$

V_{off} is an empirically determined model parameter, v_t is a physical parameter proportional to temperature, and n is derived from a host of other model and device parameters. I_{s0} is dependent on the transistor geometry and may be written as $I_{s0}' \cdot W / L$. For single devices in the normal "off" state, $V_{ds} = V_{CC}$ and $V_{gs} = 0$. Substituting these biases into Equation 3, the factor in parenthesis becomes 1 (since $V_{ds} = V_{CC} >> v_t$), and the last factor may be split into a product of exponents:

$$\begin{aligned} I_{Dsub} &= \left(\frac{W}{L}\right) \cdot I_{s0}' \cdot e^{\frac{-V_{off}}{n \cdot v_t}} \cdot e^{\frac{-V_T}{n \cdot v_t}} \qquad \text{(Eq. 4)} \\ &= \left(\frac{W}{L}\right) \cdot k_{tech} \cdot e^{\frac{-V_T}{n \cdot v_t}} \\ &= \left(\frac{W}{L}\right) \cdot k_{tech} \cdot 10^{\frac{-V_T}{S_t}} \end{aligned}$$

where $k_{tech} = I_{s0}' \cdot exp(-V_{off} / (n \cdot v_t))$ and $S_t = 2.303 \cdot n \cdot v_t$. S_t is referred to as the subthreshold swing parameter. It is a measure of how effectively a transistor shuts off and is equal to the inverse slope of $\log(I_D)$ vs. V_{gs} (in mV/decade) for $V_{gs} < V_T$. Although the channel length (L) appears explicitly in the equation, it should be noted that k_{tech} and S_t still have a complicated dependence on channel length. W is actually the dimension of interest since nearly every device is drawn at the minimum allowed L. Since L may be considered fixed, k_{tech} and S_t will be invariant for almost all of the devices in a given technology. The ratio of the two dimensions (the aspect ratio) was not included in k_{tech} since it depends on the design in which the transistor is used and not the technology.

Equation 4 applies to an isolated off transistor. This level of detail is inappropriate for reasoning at the architectural level. Therefore, we assume certain statistical properties about large numbers of devices to generalize the equation. Specifically, we assume that the distribution of transistor geometries (described by the aspect ratio) is the same across large groups of transistors employed in the same type of circuitry. The latter qualification is very important. Consider the transistors used in a cache array versus those employed in datapath logic: the cache transistors will be the minimum possible size to achieve high density, while the datapath transistors will be sized to operate at the best possible speed.

The circuit type also influences the proportion of the transistors which are switched off (f_{off}). In the absence of DC current paths (chains of on transistors between V_{CC} and ground), it is the off transistors which will determine the leakage current. In full static CMOS, half of the transistors should be off at any given time. However, other types of logic (e.g., domino, pass gate, or memory array) will have different leakage characteristics.

In addition to device geometries, the stacking factor of transistors is also dependent on the circuit type. Stacked transistors are those that are connected in series drain to source (Figure 6). The leakage current through each transistor in a stack must be equal; furthermore, the voltage drop across the entire stack can not exceed V_{CC}. Provided more than one transistor in the stack is off, the V_{ds} for the off transistors will be $< V_{CC}$. Thus, the leakage current is reduced by the $1 - e^{\frac{-V_{ds}}{v_t}}$ term in Equation 3. For a stack of four transistors, the reduction in leakage can be up to a factor of 20 [14]. Stacked transistors also have a non-zero body bias (potential difference between the source and body nodes) which affects I_{Dsub} through the variables n and V_T. We define a design dependent parameter k_{stack} that is the average leakage due to different stacking factors weighted by the portion of devices in the circuit with each stacking factor relative to the leakage of a single device. It is always less than one and will be lower in circuit types with higher average stacking factors (e.g., circuits with high fan-in gates).

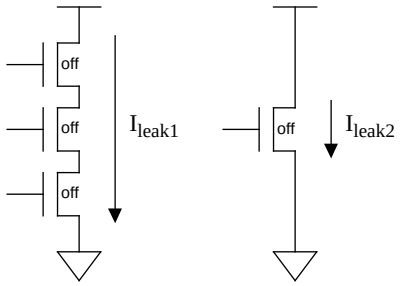

Figure 6. Leakage in stacked transistors ($I_{leak1} << I_{leak2}$)

While we have introduced the attributes of the design that affect leakage individually, they are not actually separable. Stacked transistors, for example, are generally drawn with a larger aspect ratio to make up for the reduced drive capability of stacked devices over a single device. Also, stacking factor only reduces leakage when more than one device in the stack is off. Thus, f_{off} and k_{stack} are not independent either. Because these factors are not separable, we combine them into a single circuit-dependent constant k_{design} as follows. Summing the subthreshold current given by Equation 4 for a group of N transistors, we derive:

$$I_{leakage} = \sum_{i}^{N} I_{Dsub_i} = N \cdot \overline{I_{Dsub}} \qquad \text{(Eq. 5)}$$

$$= N \cdot \boldsymbol{f}\left[\overline{\left(\frac{W}{L}\right)}, f_{off}, k_{stack}\right] \cdot k_{tech} \cdot 10^{\frac{-V_T}{S_t}}$$

$$= N \cdot k_{design} \cdot k_{tech} \cdot 10^{\frac{-V_T}{S_t}}$$

for a group of transistors with the same technology parameters. Barred parameters represent average values over all of the transistors. At this point, we note that the difference in leakage characteristics (quantified in Equation 5 by k_{tech}, V_T, and S_t) between N- and P-type MOSFET's is highly dependent on the specific technology. Provided they are similar for the two types of transistors, both types may be modeled simultaneously. In this case, k_{design} also incorporates the ratio between the two types of devices. If the devices differ significantly in the magnitude of k_{tech}, V_T or S_t, the model must be applied separately to the two groups of devices as shown in Equation 6 (where f_N is the fraction of N-type MOSFET's and the technology parameters are subscripted with the device type to which they apply). For the remainder of the paper, we assume the first case applies.

$$I_{leakage} = N \cdot \left[f_N \cdot k_{designN} \cdot k_{techN} \cdot 10^{\frac{-V_{TN}}{S_{tN}}} + \right. \qquad \text{(Eq. 6)}$$

$$\left. (1 - f_N) \cdot k_{designP} \cdot k_{techP} \cdot 10^{\frac{-V_{TP}}{S_{tP}}} \right]$$

Given that power dissipation is the product of the potential difference (voltage) and the current flowing through that difference, the total static power is given by:

$$P_{static} = V_{CC} \cdot N \cdot k_{design} \cdot k_{tech} \cdot 10^{\frac{-V_T}{S_t}} \qquad \text{(Eq. 7)}$$

Equation 7 specifies three technology dependent parameters (k_{tech}, S_t, and V_T) that may be combined into a single technology constant $\hat{I}_{leak}$:

$$P_{static} = V_{CC} \cdot N \cdot k_{design} \cdot \hat{I}_{leak} \qquad \text{(Eq. 8)}$$

where $\hat{I}_{leak}$ is the normalized leakage current (the right hand side of Equation 4 without W / L). Because of its simplicity, this variation is likely to be applied for high-level reasoning. Also, the interdependence of the technology parameters makes this model more appropriate than one where the technology parameters are seemingly independent. For MTCMOS technologies, for example, using different values of $\hat{I}_{leak}$, rather than different values of V_T for fixed k_{tech} and S_t, will be more accurate. We choose to emphasize the more detailed model of Equation 7 in the next section to underscore the nature and magnitude of the impact of the technology parameters (especially the threshold voltage) on static power.

While formulas similar to Equation 7 appear in the device literature [21, 25], they fail to differentiate the design and technology contributions to the leakage power; instead, an average per device leakage is a parameter. Such a broad parameter is impossible to estimate at any level in the design process: architects can not be expected to reason with actual leakage values during design studies, and device and process engineers can not guess about the high-level applications of various groups of devices. By separating the contributions of architectural application (design) and device physics (technology) the individual parameters can be better estimated.

4.2. Model Parameters

The parameters of the static power model of Equation 7 may be divided into two groups. The technology parameters are derived from measurements or simulations of individual devices. These parameters all appear in Equation 4 for the subthreshold leakage of a single device and are bundled into $\hat{I}_{leak}$ in Equation 8. They are all dependent on a host of lower-level process parameters (e.g., oxide thickness and doping profiles) in complex ways. The design dependent parameters (V_{CC}, N, and k_{design}) apply to groups of devices interconnected in a specific design style. Within certain constraints, they are independent of the process technology and may be varied independently. In this section, we examine each parameter in detail, focusing on relevant constraints and the determination and scaling of parameter values.

k_{tech} and S_t are relatively unimportant for high level applications of the model. Both parameters are likely to be bundled into $\hat{I}_{leak}$ along with V_T for practical applications of the model. For relative comparisons between designs targeting the same technology, the value of k_{tech} is immaterial; however, the value of k_{tech} will differ for the different threshold devices in MTCMOS technologies. The difference is easily predictable and can be estimated accurately when the threshold voltages themselves are known. S_t can potentially have a large impact on leakage current via the exponential relationship between the two. The two primary determinants of S_t are oxide thickness and temperature. Temperature control is a function of system-level design and can not be used to differentiate designs. Technologies providing multiple oxide thicknesses are not common; therefore, S_t is nearly the same for the alternate devices available in MTCMOS technologies. The scaling of oxide thickness has been slowly decreasing the magnitude of S_t over time. The minimum S_t is set by thermodynamic considerations and is about 60 mV/decade at room temperature [30]. Historical data shows that S_t is between about 80 and 100 mV/decade; SOI (silicon on insulator) technologies can more closely approach the ideal value [38].

The most important of the technology parameters is the threshold voltage V_T. It is the scaling of the threshold voltage (Figure 4) that is causing static power to become a concern. The tremendous (exponential) impact of a higher threshold voltage on static power has motivated the spread of MTCMOS technologies. At the cost of additional design and process complexity, these technologies provide devices differing in speed and leakage characteristics. Today's MTCMOS technologies provide only two options. The low-threshold voltage device provides a

small speed benefit (~10%) for a large increase in subthreshold leakage (~4×) [34]. Although V_T is a technology parameter, MTCMOS enables (crude) tuning of device characteristics to the requirements of a particular circuit.

Although V_{CC} is categorized as a design parameter, it is heavily constrained by the technology. The electric fields that occur in the transistors are directly proportional to V_{CC}; therefore, reliability limits often provide an upper bound on the supply voltage. Also, certain analog circuitry found within microprocessors (e.g., cache array sense amplifiers) requires a minimum V_{CC} to operate correctly. The reason that V_{CC} is classified as a design parameter is that it is adjusted late in the design cycle (after working chips are available) to achieve the maximum performance. Its value is made as high as possible while maintaining acceptable reliability parameters and power consumption. V_{CC} partitioning (using different supply voltages for different circuits within the chip) is also a design technique that influences this parameter. It is currently used to allow for a higher voltage for off-chip communications than used in the core. This allows the power consumption to be lowered, but complicates the design due to the required voltage translation circuitry. For this reason, finer granularity voltage partitioning is not suitable to further lower power consumption.

Under constant field scaling, V_{CC} should be reduced approximately 30% per generation. While this trend was followed in the initial reductions of supply voltage from 5 V, the emphasis on high performance has resulted in V_{CC} scaling more slowly recently than the scaling model would suggest (Figure 4) [7, 33]. The latest technology projections from the SIA forecast a continuation of this trend for the performance market [27]. In the mobile and embedded markets, the increasing pressure to limit power consumption will cause V_{CC} scaling to return to the constant-field scenario. Although V_{CC} projections for a target technology are available early in the design process, the exact value of V_{CC} is unimportant since (like k_{tech}) its value is not needed to compare alternative designs in a given technology.

The number of transistors (represented by N) is the simplest of the design variables. At the architectural level it must often be estimated since circuit designs are not yet available. Presuming a circuit with known functionality has been designed in the past, a reasonably accurate estimate may be obtained with little effort. Estimation methods are especially useful for comparison of architectural alternatives that may not reach the circuit design phase. N is only constrained by the functionality required of the circuit and the available area in which to implement it. For a given functionality, the number of transistors should be constant across generations. With more transistors available, however, overhead is likely to increase as testability and performance monitoring features are added to more circuits. Increasing clock frequency also can impact device overhead as fewer gates may be placed between latches.

The remaining design parameter k_{design} encompasses the distribution of device types (N- and P-type), geometries (W and L), states (on vs. off), and stacking factors that are characteristic of a certain circuit type (see Section 4.1). Identifying more circuit types leads to better accuracy (as the aggregate properties of circuits in a more precise class are more similar), but requires additional effort both in determining k_{design} values and in applying the model. Example circuit types appropriate for architecture-level applications include logic (e.g., datapath circuitry), static RAM array, and associative array. Derivation of k_{design} for a particular circuit design style is performed by devising a small, representative circuit for each style. Circuit simulation is then performed to obtain total leakage current (an average over several states should be used). k_{design} is then calculated using the static power model (Equation 7) with the technology parameters used during the simulation. Figure 7 presents k_{design} values for the three example design styles derived from simulation of several different technologies.

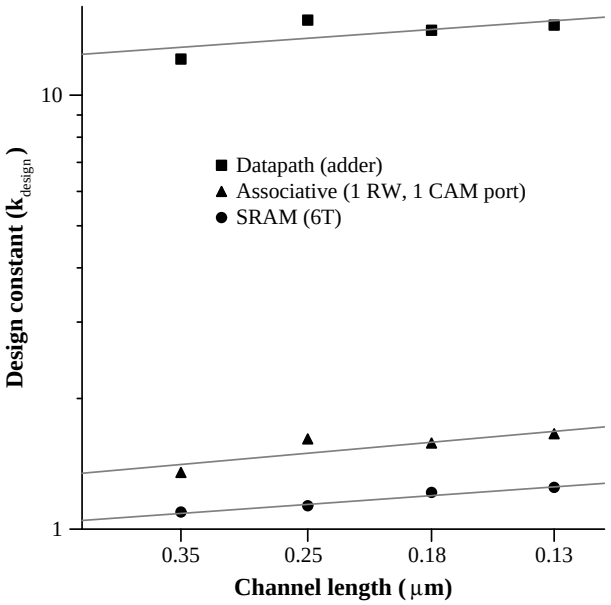

Figure 7. Technology impact on k_{design} parameters for different circuit styles

The data in Figure 7 were derived using actual transistor models and process parameters from Intel. Cells representing each sample design style were selected from the Pentium III design database and simulated together with two reference transistors (N- and P-type). All transistor dimensions were scaled appropriately for each technology prior to simulation. The leakage current of the reference transistors was averaged and divided by the aspect ratio to obtain a normalized leakage parameter $\hat{I}_{leak}$ for each technology. Each circuit's leakage current was divided by $\hat{I}_{leak} \cdot N$ to obtain the k_{design} values. The resulting values show only a slight increase over four technology generations. The values for the 0.35 μm process are systematically lower than the other values; this is the result of a different transistor model required for simulation of that technology.

Table 2 contains k_{design} values for the circuit types in Figure 7 as well as those for two additional circuit types (obtained by hand analysis of the corresponding circuits). The table also lists the number of transistors (N) used in the reference circuit for calculating the k_{design} values and notes about the specific circuits and adjustments to k_{design}. For example, an 8-bit, 4-input multiplexor would have 32 transistors (2 / bit / input * 8 bits * 4 inputs) and a k_{design} of 4.3 (1.9 + 1.2 for the third input + 1.2 for the fourth input). Static CMOS logic has two complementary (N- and P-type) transistors for each gate input. The k_{design} value varies depending on the speed and fan-out of the particular logic. Note that the median value for static logic in Table 2 is lower than that for the adder in Figure 7. The value in the table is more representative of average logic than the value for the aggressive adder used for the scaling study.

Table 2. k_{design} values

Circuit	N	k_{design}	Notes
D Flip-flop	22 / bit	1.4	Edge-triggered FF
D Latch	10 / bit	2.0	Transparent latch
2-input mux	2 / bit / input	1.9	+1.2 / input over 2
6T RAM cell	6 / bit	1.2	1 RW port
CAM cell	13 / bit	1.7	1 RW, 1 CAM
Static logic	2 / gate input	11	Depends on speed, load (± 3)

Recall that the average device geometry was incorporated into k_{design} in the form of the aspect ratio W / L. Being the ratio of two dimensions, device aspect ratios ideally do not change under scaling. The value of including these parameters as a ratio into the design constant (instead of the technology constant) is now apparent. Because the aspect ratio is independent of technology, k_{design} values (once derived) are valid for projecting static power requirements in other technologies.

5. Reducing Static Power

The model for static power presented in the previous section suggests different ways in which static power may be controlled: reducing any factor in the equation will reduce the power requirement. Thus, the static power may be lowered by reducing the supply voltage (lower V_{CC}), using fewer devices (lower N), using a more power efficient design style (lower k_{design}), or using slower devices (higher V_T, lower $\hat{I}_{leak}$). Depending on the method employed, any of these options may require performance to be sacrificed to realize power savings. We will discuss architectural applications of each of these options in this section. We conclude the section with a discussion of likely applications of speculation to power-efficient architectures.

5.1. Reducing the Supply Voltage

The supply voltage is not typically thought of as an architecturally controllable parameter. However, the nature of the architecture influences the supply voltage optimization which occurs at the end of the design cycle. Architects can enable lower supply voltages by making performance less sensitive to latency. Circuits with less strict latency requirements can operate at a lower clock frequency and supply voltage. By partitioning the circuit into several domains operating at different supply voltages, both static and dynamic power savings are possible. Modern microprocessors already use this technique to allow for a higher voltage for off-chip communication than is used in the core. Level shifter circuits are required for communications between voltage domains. The partitioning should take into account the extra delay incurred in crossing domain boundaries.

To reduce the supply voltage for the entire chip without partitioning, the global clock frequency must be reduced. Architectures which emphasize high IPC over high clock frequencies to achieve performance are superior in power characteristics provided the added complexity does not erase the gains through increased device count. The point at which an architecture falls on the frequency-IPC scale directly influences the domain in which the supply voltage may be adjusted.

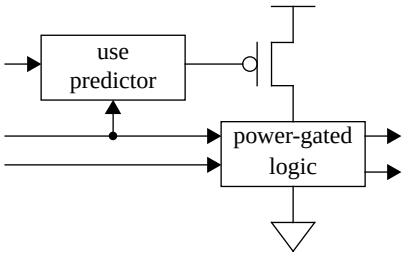

Figure 8. Power gating: gated logic receives power only when PMOS switching device is active

5.2. Reducing the Number of Devices

One obvious technique that may be employed to reduce static power is to reduce the total number of devices. Finding opportunities to reduce the device count enough to impact power dissipation without decreasing performance or functionality is difficult, however. Normal design practices eliminate obvious redundancy. Furthermore, a large number of devices must be removed to have a noticeable impact. Thus, units with replication make obvious targets. Cache size, number of functional units, and issue/retire bandwidth may all be reduced with varying degrees of difficulty and performance impact. If power optimization is a goal from the beginning, effort spent balancing the processor's resources reduces unnecessary replication by allocating fewer overall devices only where they are most needed. Another beneficial task for architects would be to equalize utilization: bursty operation requires a high maximum throughput to attain a given performance level. Equalizing resource requirements over time results in a lower total resource requirement for a given performance. Each of these approaches is appropriate for study at the architectural level.

Another method to reduce N without actually removing devices is to turn them off when they are unused. Power gating is analogous to clock gating: the supply voltage (rather than the clock) of some functional unit is switched on only when the unit is required. Additional circuitry is added to determine the need for the unit. This circuitry may monitor inputs to the switched unit or use other available signals (Figure 8). The gated circuitry will not dissipate any power when turned off. However, this must be balanced against the power dissipated by the gating circuitry and the power switching device itself. The power switching device must be large enough (W) to handle the average supply current of the circuit while in operation. If the device has a high enough threshold voltage, its leakage power can be lower than that of the gated circuit (which may use lower thresholds to be fast during operation). However, the addition of a gating device can result in reduced performance and noise margins [24, 36].

The major problem with power gating is the latency between when the signal to turn a unit on arrives and when the unit is ready to operate. Due to the huge capacitance on the power supply nodes in a unit, several clock cycles will be needed to allow the power supply to reach its operating level†. There are two alternatives which may apply regarding this latency. If the functional unit is required very rarely or is not on the critical computation path, it may not significantly impact performance to stall until the unit is ready. Alternatively, the requirement for a unit may be predicted far enough in advance for the unit to be ready when it is required.

Predicting the need for a functional unit raises the question of what kinds of microarchitectural events can be predicted accurately in advance. One obvious choice is the use of floating point functionality. Some operating systems already track the use of floating point hardware by applications to avoid saving the floating point registers on context switches when unnecessary [20]. Thus, the floating point hardware may be switched at the same granularity as context switches. Portions of the cache may also be turned off provided the working set of the application fits in a subset of the cache [22]. Other opportunities include decode logic for rare or privileged instructions, interrupt logic (a timer interrupt, usually the most frequent interrupt, at 100Hz occurs only every 10 million clock cycles at 1GHz), or logic to handle certain rare exceptions. Architectural study is ideal for determining the impact of increased startup latencies and the feasibility of prediction.

5.3. Using More Efficient Circuits

The design factors comprising k_{design} offer few opportunities for static power reduction directly. Architects may not think directly about the distribution of device geometries or stacking factors; however, the requirements of the microarchitecture ultimately determine the type of circuitry which can be used for its implementation. For example, targeting higher IPC at a lower clock frequency allows for more logic between pipeline latches; power savings are realized by allowing the use of more complex gates with larger average stacking factors.

The k_{design} values in Table 2 suggest some additional ways of employing power-efficient circuits. Wide multiplexors should be avoided as they have a cost which grows super-linearly with the number of inputs. A tri-state bus with multiple drivers can accomplish the same function with lower total leakage (tri-state drivers have stacked devices where pass-gate multiplexors do not). Associative arrays are approximately three times leakier (including the larger number of transistors) than simple random-access memories. Implementing pseudo-associativity using hashing may be appropriate depending on the exact requirements of the microarchitecture.

5.4. Using Multiple Threshold Voltages

Technologies which provide multiple threshold voltages allow for an even better tradeoff between static power and performance. By using slower transistors, the leakage current may be reduced significantly. Note that it is not sufficient to simply clock a regular device more slowly, since this does not affect the subthreshold leakage. The transistor must actually be *slower*.

Different transistor speeds may be used in different ways. One method would be to employ the fast devices only along critical timing paths. Although algorithms have been proposed to automatically perform this task [29, 36], a concern is that automated modification of path delays could result in races. A second technique involves determining which functional units require the lowest latencies and allocating the budget of fast, leaky devices to these units only. To reduce dynamic power consumption, at least one announced product divides core logic into clock domains of different frequencies [18]. Limited partitioning has occurred ever since core frequencies exceeded bus frequencies. Partitioning enables one to use a device speed appropriate to the particular clock domain in which the device is to be located. Architects are best suited to determine which functionality belongs in which clock domain and what particular method of interdomain communication should be used. This partitioning allows for optimization of both static and dynamic power consumption.

Threshold voltage may also be adjusted by applying a voltage to the body node of a transistor to reverse bias the source-body junction. By raising the threshold voltage, this technique also results in slower devices. The ideal use of such a technique would be to apply the body bias only when the circuitry is unused and return to normal conditions when the circuit is required. The very high resistance of transistor body nodes results in a similar problem as in power gating, but of a much higher magnitude: establishing or removing a body bias will require a long time due to the high resistance of the body nodes of MOSFET's. Therefore, functional units that have long idle periods and startups that can be accurately predicted with architectural state are most appropriate for these techniques.

5.5. Power Reduction with Speculation

Speculation can be an important tool for architects when designing power-efficient architectures. Specifically, it provides a means of using slower devices without proportionally impacting performance. The performance critical speculation circuitry employs fast devices, while the slower devices are used to verify the speculative results. The additional latency is incurred only when the speculation is incorrect. In some cases, the circuitry to perform the speculation is simple and very few of the power-hungry fast devices are required. The verification circuitry may use higher-threshold devices, use a lower supply voltage, run at a lower clock frequency, or some combination resulting in both static and dynamic power savings over a fast, non-speculative solution at little performance cost. An architecture such as DIVA [2] in which a slow checker augments a fast, highly speculative core could directly benefit from intelligent partitioning based on device speed requirements.

As a more specific example, consider data speculation on L1 cache accesses. Such speculation is already implemented on Intel's Willamette for performance reasons [10]. L1 cache accesses are on the critical execution path for load instructions. Recognizing that the majority of such accesses hit in the cache, it is reasonable to speculatively assume that any data retrieved from a direct-mapped cache is correct prior to checking the tags. The cache tags and tag match logic may then be implemented with slower, more efficient circuitry. Mis-speculation detection suffers from an increased latency implied by the slower circuitry. Performance is only impacted in the event of an L1 cache miss. Without speculation, the tags and matching logic would have to be fast to avoid a significant performance penalty. The potential power savings depends on the exact cache behavior, the amount of logic that was moved off of the critical path, and the amount of additional logic required to recover from mis-speculation.

Another application of speculation was referred to briefly in Section 5.2 in the context of predicting when certain circuitry will be needed. It may be hard to determine when certain functional units are required and when they may be shut off to save power. Instead of choosing to leave these units on constantly, it may be more appropriate to speculatively power-down such functional units. Provided the speculation accuracy is reasonable, a large decrease in power consumption would incur only a small performance penalty. Mis-speculation would be visible as increased latency of the functional unit. In architectures which are power-limited (the peak performance is limited by power

† The switching device must supply current corresponding to the average power dissipation. Consider a circuit representing 1% of a chip that dissipates 150 W at 1.5 V. The device must conduct 1 A of average current. Assuming a decoupling capacitance of 500 nF for the entire chip, the supply node capacitance of the switched unit will be approximately 5 nF. Charging 5 nF to 1.5 V with 1 A takes approximately (Equation 1): (5 nF)(1.5 V) / (1 A) = 7.5 ns or 7.5 cycles at 1 GHz.

considerations), such techniques could actually allow for higher performance.

6. Related Work

Prior work on power modeling of power dissipation at the architectural level has been focused almost entirely on dynamic power. The oft quoted $P_{dyn} = CV_{CC}^2 f$ is easily derived by consideration of a loaded inverter (see for example [37]). This metric is often used to compare the dynamic power requirements of alternative designs. A survey of more detailed power modeling tools was compiled by Blaauw, *et. al*. [4]. Several researchers have reported modifying performance simulators to provide power estimates as well [6, 35].

Reducing power consumption in microprocessors is the subject of active research. These works tend to focus on caches because of the large potential gains and ease of modeling [1, 3, 16, 28]. Dynamic power reduction in more irregular structures is demonstrated by efficiency based arguments wherein the amount of switching or needless work is reduced [5, 11, 19, 32]. Static power has been addressed in recent work by Powell, *et. al*. [22] which combines circuit and architectural techniques to reduce the power consumption in a processor's cache. The cache miss rate is used to determine the working set size of the application relative to that of the cache. Power is then removed from the unused portions of the cache via a gating transistor.

The device and circuits communities have been concerned with increasing static power for several generations. Besides numerous publications of specific technologies with improved leakage characteristics (e.g., MTCMOS), several reviews have focussed on leakage current as an important concern in future technologies. Keshavarzi, *et. al*. present the various leakage modes of the MOS transistor and identify subthreshold leakage as the dominant one [15]. De and Borkar project leakage power growing 5× per generation and conclude that power dissipation and delivery will be the main barrier to future scaling [8].

7. Conclusion

Static power dissipation due primarily to subthreshold leakage will become an important component of overall power dissipation. Technology trends are reducing the transistor threshold voltage to achieve performance target. While dynamic power is partially offset by the reduction in supply voltage that occurs during scaling, static power is increasing exponentially as the threshold voltage is decreased. Static power will likely contribute as much to total power as dynamic power in as little as two technology generations unless architects consider it as important as dynamic power when making design tradeoffs.

Modeling static power consumption at the architectural level is possible using a relatively simple equation (Equation 7). The equation combines technology-based factors (k_{tech}, V_T, and S_t) with design-dependent parameters (V_{CC}, N, and k_{design}). A simpler version of the model combines the technology parameters into a single constant $\hat{I}_{leak}$. Each of the parameters is readily obtainable by projecting technology trends or performing simple simulations. The model provides a useful level of abstraction for application at an early stage in the design process. Low-level detail is sacrificed for ease of application. Secondly, the relative accuracy of model predictions does not require precise values of technology parameters which may not be available. Finally, the model illuminates various approaches for reducing the static power dissipation.

Reducing the number of devices used is a straightforward approach when the performance loss may be controlled or mitigated by other factors. Turning off unused devices is another way to control power consumption although the long restart latency must be considered. It may be possible to predict some events far enough in advance to hide this latency. Partitioning the design into blocks based on the latency requirements can enable per-block supply voltage tuning or the selective use of high threshold devices. High threshold (i.e., slower) devices are inherently less leaky and reduce power requirements. Technologies that provide multiple threshold devices are already available and will become commonplace.

One useful application of slower devices is to the logic used to check the correctness of speculation. This decouples the increased latency of the slower logic from overall performance since the slower logic is on the critical execution path only during mis-speculation recovery. By using fewer fast devices to generate the speculative result than would be required to generate the actual result, static power savings are achieved.

Many of the techniques described to limit static power dissipation have the side effect of controlling dynamic power dissipation as well. Reducing the number of devices (N) directly reduces the switching capacitance (C) which affects dynamic power. In the absence of clock gating, power gating has a similar effect since only powered devices contribute to the switching capacitance. Using lower supply voltages in less critical logic blocks also reduces dynamic power. Finally, because the switching frequency *f* is limited by the device performance (V_T), reducing the frequency wherever possible also benefits dynamic power. In contrast, techniques for reducing dynamic power dissipation (e.g., clock gating) do not generally improve static power dissipation. Considering only dynamic power dissipation can actually lead to choosing a microarchitecture with higher total power dissipation (e.g., one that uses fast, leaky devices to achieve high-throughput when latency is not critical).

Architects are in a position to affect the power requirements of their designs. Given the ability to reason about power, the architect can factor that information in when making trade-offs between alternative designs. Due to the long design cycle, architects must be considering power dissipation now to deliver products which are not unduly constrained by power.

Acknowledgements

The authors would like to thank Intel Corp. for allowing use of their simulation infrastructure. Richard Green and Jeff Smith of Intel provided comments and advice which were extremely helpful. Finally, the authors thank the referees for their reviews.

This work was supported in part by National Science Foundation grants MIP-9505853 and CCR-9900584, donations from Intel Corp. and Sun Microsystems, and the University of Wisconsin Graduate School. Adam Butts is supported by a fellowship from the Fannie and John Hertz Foundation.

References

[1] D. Albonesi. Selective Cache Ways: On-Demand Cache Resource Allocation. In *Proceedings of the 32nd International Symposium on Microarchitecture*, November 1999. pp. 248-259.

[2] T. Austin. DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design. In *Proceedings of the 32nd International Symposium on Microarchitecture*, November 1999. pp. 196-207.

[3] R. Bahar, G. Albera, and S. Manne. Power and Performance Tradeoffs Using Various Caching Strategies. In *Proceedings of the International Symposium on Low Power Electronics and Design*, 1998. pp. 64-69.

[4] D. Blaauw, A. Dharchoudhury, R. Panda, S. Sirichotiyakul, C. Oh, and T. Edwards. Emerging Power Management Tools for Processor Design. In *Proceedings of the International Symposium on Low Power Electronics and Design*, 1998. pp. 143-148.

[5] D. Brooks and M. Martonosi. Dynamically Exploiting Narrow Width Operands to Improve Processor Power and Performance. In *Proceedings of the 5th International Symposium on High-Performance Computer Architecture*, January 1999. pp. 13-22.

[6] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *Proceedings of the 27th International Symposium on Computer Architecture*, June 2000. pp. 83-94.

[7] B. Davari. CMOS Technology Scaling, 0.1 μm and Beyond. In *Proceedings of the International Electron Devices Meeting*, 1996. pp. 555-558.

[8] V. De and S. Borkar. Technology and Design Challenges for Low Power and High Performance. In *Proceedings of the International Symposium on Low Power Electronics and Design*, 1999. pp. 163-168.

[9] R. Dennard, F. Gaensslen, H. Yu, V. Rideout, E. Bassous, and A. LeBlanc. Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions. *IEEE Journal of Solid-State Circuits*, 9(5), October 1974. pp. 256-268.

[10] P. Glaskowsky. Pentium 4 (Partially) Previewed. *Microprocessor Report*, August 28, 2000.

[11] R. Gonzalez and M. Horowitz. Energy Dissipation in General Purpose Microprocessors. *IEEE Journal of Solid-State Circuits*, 31(9), September 1996. pp. 1277-1284.

[12] M. Horowitz, T. Indermaur, and R. Gonzalez. Low-Power Digital Design. In *Proceedings of the 1994 IEEE Symposium on Low Power Electronics*. pp. 8-11.

[13] Intel Corporation. *Pentium III Processor for the SC242 at 450 MHz to 1.13 GHz Datasheet*. pp. 26-30.

[14] M. Johnson, D. Somasekhar, and K. Roy. Models and Algorithms for Bounds on Leakage in CMOS Circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 18(6), June 1999. pp. 714-725.

[15] A. Keshavarzi, K. Roy, and C. Hawkins. Intrinsic Leakage in Low Power Deep Submicron CMOS ICs. In *Proceedings of the IEEE International Test Conference*, 1997. pp. 146-155.

[16] J. Kin, M. Gupta, and W. Mangione-Smith. The Filter Cache: An Energy Efficient Memory Structure. In *Proceedings of the 30th International Symposium on Microarchitecture*, December 1997. pp. 184-193.

[17] P. Ko, J. Huang, Z. Liu, and C. Hu. BSIM3 for Analog and Digital Circuit Simulation. In *Proceedings of the IEEE Symposium on VLSI Technology CAD*, January 1993. pp. 400-429.

[18] K. Krewell. Quicktake: Willamette Revealed. *Microprocessor Report*, February 2000. p. 19.

[19] S. Manne, A. Klauser, and D. Grunwald. Pipeline Gating: Speculation Control for Energy Reduction. In *Proceedings of the 25th International Symposium on Computer Architecture*, June 1998. pp. 132-141.

[20] H. Massalin and C. Pu. Threads and Input/Output in the Synthesis Kernel. In *Proceedings of the 12th Symposium on Operating Systems Principles*, December 1989. pp. 191-201.

[21] E. Nowak. Ultimate CMOS ULSI Performance. In *Proceedings of the International Electron Devices Meeting*, 1993. pp. 115-118.

[22] M. Powell, S. Yang, B. Falsafi, K. Roy, T. Vijaykumar. Gated-V_{DD}: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories. In the *Proceedings of the International Symposium on Low Power Electronics and Design*, July 2000. pp. 90-95.

[23] J. Rabaey. *Digital Integrated Circuits: A Design Perspective*. Prentice-Hall, 1995.

[24] K. Roy. Leakage Power Reduction in Low-Voltage CMOS Design. In *Proceedings of the IEEE International Conference on Circuits and Systems*, 1998. pp. 167-173.

[25] T. Sakurai, H. Kawaguchi, and T. Kuroda. Low-Power CMOS Design through V_{TH} Control and Low-Swing Circuits. In *Proceedings of the International Symposium on Low Power Electronics and Design*, 1997. pp. 1-6.

[26] T. Sakurai and A. Newton. Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas. *IEEE Journal of Solid-State Circuits*, 25(4), April 1990. pp. 584-594.

[27] Semiconductor Industry Association. *International Technology Roadmap for Semiconductors*, 1999 edition. Austin, TX: SEMATECH, 1999.

[28] C. Su and A. Despain. Cache Designs for Energy Efficiency. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, 1995. pp. 306-315.

[29] V. Sundararajan and K. Parhi. Low Power Synthesis of Dual Threshold Voltage CMOS VLSI Circuits. In *Proceedings of the International Symposium on Low Power Electronics and Design*, 1999. pp. 139-144.

[30] S. Sze. *Physics of Semiconductor Devices, 2nd. Ed*. John Wiley and Sons, 1981.

[31] V. Tiwari, R. Donnelly, S. Malik, and R. Gonzalez. Dynamic Power Management for Microprocessors: A Case Study. In *Proceedings of the 10th International Conference on VLSI Design*, 1997. pp. 185-192.

[32] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, F. Baez. Reducing Power in High-Performance Microprocessors. In *Proceedings of the Design Automation Conference*, 1998. pp. 732-737.

[33] Y. Taur and E. Nowak. CMOS Devices Below 0.1 μm: How High Will Performance Go? In *Proceedings of the International Electron Devices Meeting*, 1997. pp. 215-218.

[34] S. Thompson, P. Packan, and M. Bohr. MOS Scaling: Transistor Challenges for the 21st Century. *Intel Technology Journal*, Q3 1998.

[35] N. Vijaykrishnan, M. Kandemir, M. Irwin, H. Kim, and W. Ye. Energy-Driven Integrated Hardware-Software Optimizations Using SimplePower. In *Proceedings of the 27th International Symposium on Computer Architecture*, June 2000. pp. 95-106.

[36] Q. Wang and S. Vrudhula. Static Power Optimization of Deep Submicron CMOS Circuits for Dual V_T Technology. In *Proceedings of the International Conference on Computer-Aided Design*, 1998. pp. 490-496.

[37] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design: A Systems Perspective, 2nd. Ed*. Addison-Wesley, 1993.

[38] D. Wouters, J. Colinge, H. Maes. Subthreshold Slope in Thin-Film SOI MOSFET's. *IEEE Transactions on Electron Devices*, 37(9), September 1990. pp. 2022-2033.