Architectural Power Models for SRAM and CAM Structures Based on Hybrid Analytical/Empirical Techniques

Xiaoyao Liang, Kerem Turgay, David Brooks
School of Engineering and Applied Sciences, Harvard University
33 Oxford Street, Cambridge MA 02138 USA

Abstract— **The need to perform power analysis in the early stages of the design process has become critical as power has become a major design constraint. Embedded and high-performance microprocessors incorporate large on-chip cache and similar SRAM-based or CAM-based structures, and these components can consume a significant fraction of the total chip power. Thus an accurate power modeling method for such structures is important in early architecture design studies. We present a unified architecture-level power modeling methodology for array structures which is highly-accurate, parameterizable, and technology scalable. We demonstrate the applicability of the model to different memory structures (SRAMs and CAMs) and include leakage-variability in advanced technologies. The power modeling approach is validated against HSPICE power simulation results, and we show power estimation accuracy within 5% of detailed circuit simulations.**

I. Introduction

To satisfy the demand for power estimation during early-stage design, several architectural power models and simulators have been developed around architectural performance simulators [2], [3], [8]. These tools can be used to provide insight into high-level architectural power/performance tradeoffs in embedded and high-performance chips. Given the importance of these decisions, flexible and accurate models for architectural constructs must be developed.

There are two predominant approaches for developing architectural power models: analytical and empirical. Analytical models use library device parameters to calculate power consumption using analytical formulas for the capacitance of key internal nodes. They are flexible and easy to implement, but generally have poor accuracy, sometimes with more than 20% error for deep submicron designs. Another problem is that the parameters, fitting curves, and formulas usually need to be modified for each new technology node. This is especially problematic when modeling complex sources of static power and interactions with process variations. In contrast, empirical models rely on circuit simulation of the entire structure to be modeled, often from previous generation designs. These models are very accurate, but lack flexibility, and because they rely on previous designs, may also not scale well to new technology nodes. Detailed circuit-level design and verification for large hardware structures takes significant time which may not meet the needs of high-level architecture studies.

This paper discusses a new methodology for high-level power modeling that retains the flexibility advantages of analytical models with the accuracy advantages of empirical models. The approach involves building very small hardware blocks and extracting power information from detailed circuit simulation using HSPICE. Detailed simulations on very small blocks can capture relevant technology information about new process technology nodes. Given this empirical information, we form analytical formulas that combine these building blocks together to accurately model larger structures. Unlike analytical models, this method encapsulates the details of the circuit implementation and device formulas, relieving architects of this burden.

The modeling methodology presented in this paper presents several explicit advantages compared with previous efforts:

- Accuracy comparable to full-circuit HSPICE simulation, but model complexity comparable to analytical models.
- Parameterizability from the architectural viewpoint and applicable to RAM/CAM memories.
- Scalability across technology nodes, temperatures, and includes the impact of process variation.

II. Background and Related Work

RAM/CAM-based array memories (including caches, TLBs, queues, and buffers) consume a significant fraction of the power in a high-performance microprocessor. Thus, power models for SRAM and CAM structures have become an essential tool for system designers and computer architects. These tools have two distinct styles: analytical models based on capacitance estimates and empirical models derived from detailed circuit designs.

A. Analytical Models

Analytical models for dynamic power dissipation can be constructed by identifying the key internal nodes of a block and deriving analytical formulas for the node capacitance. This technique has been widely applied to RAM/CAM structures such as caches, register files, buffers, and queues [3], [7].

For example, most popular analytical models use Equations 1 and 2 to calculate SRAM array bitline energy, where $C_{bitline}$ is the total stacked bitline capacitance and V_{swing} is the voltage swing of the bitline. The total bitline capacitance includes N stacked access transistor diffusion capacitance, pre-charge circuit capacitance, bitline wire capacitance, and column select circuit capacitance.

$$E = C_{bitline} \times V_{DD} \times V_{swing} \quad (1)$$

$$\begin{aligned} C_{bitline} =& C_{diff\_access} \times N + C_{diff\_precharge} \\ &+ C_{diff\_column} + C_{wire} \end{aligned} \quad (2)$$

Analytical power models provide a fast, flexible method of estimating dynamic power consumption. However, the accuracy of these models suffers for several reasons:

- **Inaccurate Capacitance Estimates:** Node capacitance depends heavily on the actual circuit design (including overlap and node-to-node coupling capacitances). Node capacitance also changes with the supply voltage during read and writes, and analytical models generally ignore these effects. Capacitance can be difficult to accurately estimate using simple analytical formulas.

- **Direct-Path Current:** Analytical models usually neglect direct-path current, because this is difficult to estimate as it depends heavily on the circuit design. However, direct-path current can be appreciable in memories.
- **Memory Cell Toggle Power:** Many analytical models neglect the power to toggle memory cell bits when writing to the memory. Although this power is usually small, it can be a fairly significant error source when there are periods of intensive writes and toggles to the memory bits.

Analytical power models for leakage power are also available, based on simple analytical formulas for subthreshold leakage and gate leakage [4], [10]. For example leakage current is commonly estimated as follows:

$$I_{leak} = \beta \cdot e^{b(v_{dd} - V_{dd0})} \cdot V_t^2 \cdot (1 - e^{\frac{-V_{dd}}{V_t}}) \cdot e^{\frac{-|V_{th}| - V_{off}}{nV_t}} \tag{3}$$

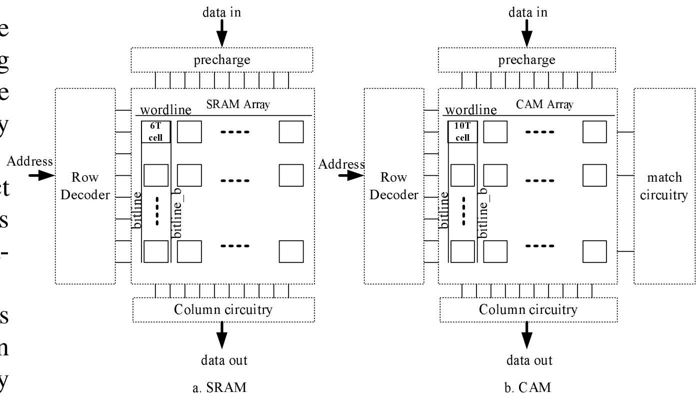
This leakage model is an approximation and several of the parameters are highly dependent on circuit topology and circuit state, e.g., in Equation 3, parameters β and n are empirically derived. In [5], the authors employ a technique that uses analytical formulas for dynamic power augmented with circuit simulations to calibrate leakage estimates.

B. Empirical Approaches

Empirical modeling approaches [2], [8] typically utilize detailed, full circuit power simulation data that is available from prior microprocessor designs and develop energy models for next-generation processors using traditional scaling theory. Memory compilers provided by some foundries provide an alternative method to automatically generate empirical models for SRAM structures.

A major drawback to empirical modeling approaches is that they require time-consuming detailed simulation of large array structures. This is especially problematic for tools like CACTI which perform power estimation within an inner-loop of a optimization flow designed to choose an optimal cache configuration for area, power, and latency. Full-array simulation is even less attractive for integrated modeling toolkits that tightly couple cycle-level architectural simulators with power models to understand the impact of architectural activity on leakage-dependent temperature variations (temporal and spatial thermal hotspots) and power-induced voltage variations (inductive noise).

Empirical modeling approaches such as memory compilers inherently lack flexibility because they often place limitations on the type and size of the generated memory and do not allow designers to guide the memory layout and circuit design choices (e.g. varying degrees of bitline folding). In contrast, analytical models can be extended for new circuit design styles [6]. Memory compilers are also generally limited to simple single and dual-ported SRAMs and cannot generate other structures that are common in high-performance microprocessor design such as heavily multi-ported register files or integrated RAM/CAM memories that are part of instruction issue logic. Finally, the tools are specific to a single foundry process and cannot be used with predictive technology models.



a. SRAM

b. CAM

Fig. 1. Basic SRAM and CAM structures.

III. Block-based Power Models

The power modeling approach that we propose is a hybrid between the previously developed analytical and empirical models. This approach seeks to address the accuracy limitations of analytical models while still providing flexibility to model a diverse range of hardware structures across a range of technology nodes. We propose to use simple HSPICE circuit simulation to extract power information for small blocks that form the basis of analytical models for larger components. Since these HSPICE simulations are only performed on very small circuit blocks, the simulation time is short in contrast to the very long simulation time of empirical models. Also, since we base our power estimation on real circuit power information, the accuracy will be significantly improved compared with pure analytical models.

The rest of the paper describes this approach for SRAM and CAM memories: two commonly used structures in embedded and high-performance microprocessors. Fig. 1 illustrates the basic SRAM and CAM structures that we use. These structures are composed of address decoders, core arrays with bit cells, column mux logic, and sense amplifiers.

A. Cell Array Power Models

In this subsection, we discuss dynamic and leakage power models for the core array structure which demonstrates the benefits of this power modeling methodology.

1) Dynamic Power Model For SRAM Array: The dynamic power dissipation in the array consists of wordline capacitance dissipation, bitline capacitance dissipation and short circuit power consumption. For wordline dissipation, the output driver of the row decoder charges the wordline each time the row is being read or written, resulting in wordline power consumption. The bitline dissipation is divided into the read cycle and the write cycle. In the read cycle, the power dissipation results

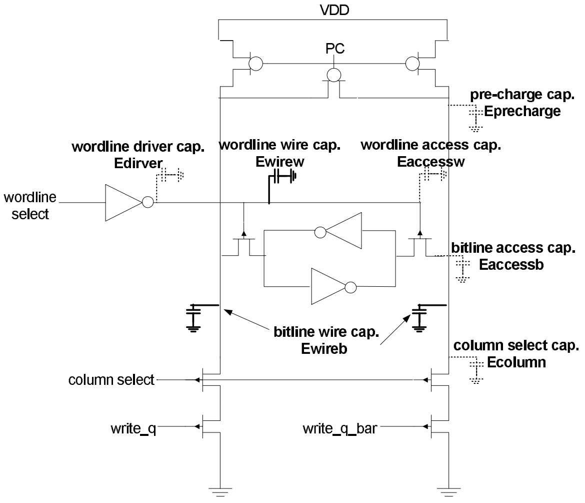


Fig. 2. Single cell model.

from charging and discharging of the bitline capacitance through the bitline pre-charge circuit. Each time the bit is read out, read energy (E_{rd}) is consumed. For the write cycle, one of the bitlines will still consume energy for each write for the same reason as reads. However, writes have an additional source of power dissipation related to the bit toggle rate. If the data value written into the cell bit is different from the previously stored value, there will be energy consumption to toggle the cell bit. Also during the toggling, there is a small period when both the PMOS and NMOS of the cell inverter are conducting which causes short circuit power. The write energy (E_{wr}) is defined as the summation of all of these components.

The single cell model shown in Fig. 2 is the first small block model that we consider. It consists of a single-bit memory cell (6T) with pre-charge circuits, wordline driver, and column select logic. The components drawn with solid lines are the real components in the circuit used for HSPICE simulation (e.g. the simulation decks include explicit wire capacitance), while components drawn with dashed lines are abstract components shown only for illustrative purposes. The transistor geometry size and wire capacitance are extracted from a single-cell layout. Fig. 3 shows a two-cell bitline model. It is very similar to the single-cell model with the addition of another memory bit cell stacked on the bitlines. Fig. 4 shows a two-cell wordline model. Compared with the single-cell model, it includes two horizontally nested bit cells with two pre-charge and bitline circuits. Because the two cells belong to the same wordline, only one wordline driver is needed.

Once we have designed these blocks, we can develop our power models. Capacitance plays an important role for dynamic power. Each time the circuit charges or discharges the capacitance, a certain amount of energy is consumed. Unfortunately, most of these capacitance values are supply voltage related. For simple power estimation, it is quite natural to use greatly simplified formulas to model the capacitance, but the result tends to be very inaccurate.

However, the basic structures are quite regular and it is possible to perform circuit simulation on the small blocks and calculate the power consumption for the whole structure without deriving the exact capacitance value. We will first consider the memory array energy consumption for a read cycle. As shown in Fig. 2, we define the dynamic energy consumed due to the existence of capacitance of a pre-charge transistor as $E_{precharge}$. Similarly, we define the dynamic energy consumed by capacitance of a column select transistor and a wordline driver as E_{column} and E_{driver}. We also define the dynamic bitline energy consumed due to the capacitance of an access transistor as $E_{accessb}$, the bitline wire capacitance of a single-cell as E_{wireb}, the dynamic wordline energy consumed due to the capacitance of an access transistor as $E_{accessw}$, and the wordline wire capacitance of a single-cell as E_{wirew}. With these definitions, we can easily express the energy consumption for a single read to the memory array for the single cell model as in Eq. (4). The energy consumption for a single read for the two-cell bitline model is shown in Eq. (5). Eq. (6) gives the energy consumption for a single read for the two-cell wordline model. Subtracting Eq. (4) from Eq. (5), gives the dynamic energy consumption for a single access transistor capacitance and wire capacitance on the bitline, as shown in Eq. (7). By multiplying Eq. (4) by 2 and then subtracting Eq. (6), we obtain the dynamic energy consumption of the wordline driver transistor capacitance, as in Eq. (8). Subtracting Eq. (5) from Eq. (6), we obtain the dynamic energy consumption of pre-charge capacitance, column select capacitance and wordline access transistors capacitance with wordline wire capacitance, as shown in Eq. (9). From Eq. (7), Eq. (8) and Eq. (9), we can easily calculate the total memory array energy consumption for a single read as in Eq. (10), assuming M columns and N rows. By applying a test with a single read to the three small models in HSPICE simulation, we can extract circuit-level accurate energy values for E_1, E_2 and E_3. Then using Eq. (10), it is easy to estimate the array energy consumption for a single read.

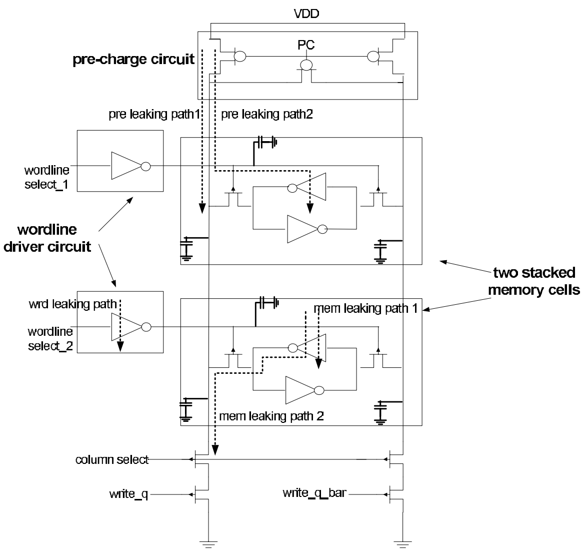


Fig. 3. Two-cell bitline model.

$$\begin{aligned} E_1 = E_{precharge} + E_{accessb} + E_{wireb} + E_{column} \\ + E_{driver} + 2E_{accessw} + E_{wirew} \end{aligned} \qquad (4)$$

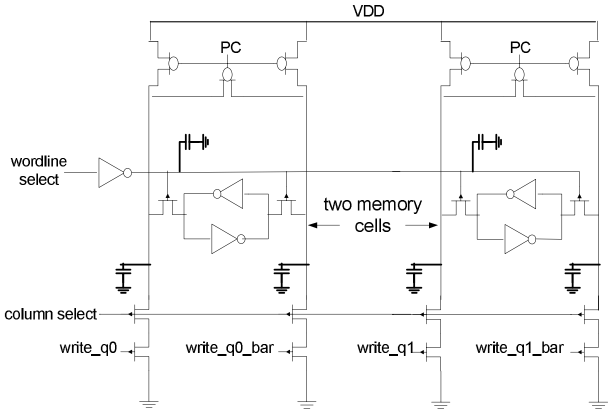


Fig. 4. Two-cell wordline model.

$$E_2 = E_{precharge} + 2E_{accessb} + 2E_{wireb} + E_{column} + E_{driver} + 2E_{accessw} + E_{wirew} \tag{5}$$

$$E_3 = 2E_{precharge} + 2E_{accessb} + 2E_{wireb} + 2E_{column} + E_{driver} + 4E_{accessw} + 2E_{wirew} \tag{6}$$

$$E_{accessb} + E_{wireb} = E_2 - E_1 \tag{7}$$

$$E_{driver} = 2E_1 - E_3 \tag{8}$$

$$E_b = E_{precharge} + E_{column} + 2E_{accessw} + E_{wirew} = E_3 - E_2 \tag{9}$$

$$\begin{aligned} E_{rd} &= (E_{accessb} + E_{wireb})MN + E_bM + E_{driver} \\ &= E_2M(N-1) - E_1(MN-2) + E_3(M-1) \end{aligned} \tag{10}$$

Write energy calculation follows in a similar manner to the above energy calculation, and we call the energy E'_{wr}. In addition to the bitline and wordline energy consumption for each write, there is additional energy dissipation if the bit stored in the cell is toggled. In order to get this energy information, we need to perform additional HSPICE simulations. Two tests of different write patterns are applied to the single-cell model. One test is the write pattern 1-0-1 and the other is the write pattern 1-1-1. From these two tests, we obtain the energy consumption E_{101} and E_{111}, and Eq. (11) gives the energy consumption for a single bit toggle. This includes the cell internal dynamic power for toggling the bit as well as the short circuit power. By using Eq. (11) and assuming TG_b bits toggle in a single write, single write energy can be calculated as Eq. (12).

$$E_{toggle} = (E_{101} - E_{111})/2 \tag{11}$$

$$E_{wr} = E'_{wr} + E_{toggle} \times TG_b \tag{12}$$

The models are simple to develop and the total HSPICE simulation time to run the three models is much less than simulation time for the whole large memory array. There is no need to obtain the technology details for the calculation of transistor capacitance, because our power models do not rely on the value of node capacitance, but extract the power directly from circuit simulation for the whole charging or discharging period. Furthermore, it is simple to obtain short circuit power in write toggles.

2) Leakage Power Model For SRAM Array: One of the main difficulties for analytical models is lack of accurate leakage power models. It is especially important to model leakage power for SRAM arrays designed below 0.13um, because large SRAM structures can consume substantial leakage power simply because of the large number of leaking transistors.

The leakage power usually depends on supply voltage, number of transistors, circuit states, temperature, and other process conditions. In Fig. 3, we first define 5 states the circuit can be in: pre-charge, read selected, read not selected, write selected and write not selected, denoted as (P, Rs, Rn, Ws, Wn). We then break the circuit into three major components: wordline drivers, memory cells, and pre-charge circuits, denoted as (wrd, mem, pre). We denote the leakage power for the wordline driver circuit in pre-charge state as $P_{leak\_wrd,P}$, and so on. Fig. 3 also shows the primary leakage paths for each component in the circuit. We first put the model circuits in different states for leakage measurements and extract the leakage power for each of the three components in each of the states from HSPICE simulation. With the information extracted, the total array leakage power can be calculated in Eq. (13), where x%, y%, z% denote the percentage time the array is in pre-charge (P), read (R, i.e. Rs and Rn) and write (W, i.e. Ws and Wn) states which can be obtained from architecture simulators.

$$\begin{aligned} P_{leak\_array} = {} & (P_{leak\_wrd,P}N + P_{leak\_mem,P}MN \\ & + P_{leak\_pre,P}M)x\% + (P_{leak\_wrd,Rs} \\ & + P_{leak\_wrd,Rn}(N-1) + P_{leak\_mem,Rs}M \\ & + P_{leak\_mem,Rn}M(N-1) + P_{leak\_pre,R}M)y\% \\ & + (P_{leak\_wrd,Ws} + P_{leak\_wrd,Wn}(N-1) \\ & + P_{leak\_mem,Ws}M + P_{leak\_mem,Wn}M(N-1) \\ & + P_{leak\_pre,W}M)z\% \end{aligned} \tag{13}$$

3) Leakage Power Model For Process Variations: Gate length (L) and threshold voltage (V_{th}) have been identified as the two major sources of process variations. Both sources of variation impact leakage in different ways. Furthermore, in many cases, the two variation sources are coupled (e.g. short and narrow-channel effects) making it very difficult to derive closed-form analytical models for the calculation of leakage distributions. As devices continue to scale to smaller feature sizes, analytical models will need to be updated constantly to take additional effects into consideration. Since our block-based model uses circuit simulation, we can easily incorporate the new power characteristics of a technology by simply replacing the device library with no change to the modeling approach. We can simulate leakage power under a particular variation range by sweeping different gate length and threshold voltages. Since our technique only requires the two-cell bitline model for leakage simulation, the simulation time is relatively short and is a one-time cost for each new technology library.

We model the threshold voltage as a random variable, but we consider the correlation of gate length variations. To capture this effect, we use the method introduced in [1]. The SRAM area is divided into a multi-level quad-tree partitioning as

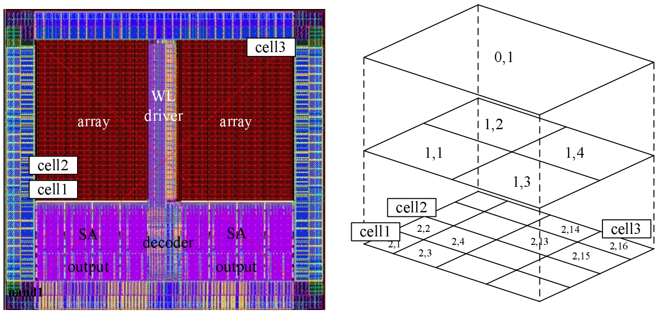


Fig. 5. Quad-tree for WID gate length variations.

Parameter	Testing Model	Operation	Notes
R1	Single-cell model	Read	Read 1 bit
R3	Two-cell bitline model	Read	Read 1 bit
W1	Single-cell model	Write	100% bit toggling
W3	Two-cell bitline model	Write	100% bit toggling
W2	Two-cell wordline model	Write	0% bit toggling
H1	Single-cell model	Match (Hit)	Hit a single bit
H2	Two-cell wordline model	Match (Hit)	Hit a single bit
H3	Two-cell bitline model	Match (Hit)	Hit a single bit
M2	Two-cell wordline model	Match (Miss)	Miss all bits
L1	Single-cell model	Idle	Idle state for leakage power
L2	Two-cell wordline model	Idle	Idle state for leakage power
L3	Two-cell bitline model	Idle	Idle state for leakage power

TABLE I
CAM Power Model Parameters Extracted from Circuit Sims.

shown in Figure 5. We show the layout of a sample SRAM design in the figure to illustrate the method. The area is covered with several layers of quadrants. For each quadrant we generate a random variable according to a normal distribution. Individual gate length variation can be obtained by summing all the random variables of the quadrants it belongs to. Therefore, $cell1$ and $cell2$ have strong correlation because they share the variable $rand_{0,1}$ and $rand_{1,1}$, while $cell1$ has less correlation with $cell3$, because they only share the variable $rand_{0,1}$. The exact location of the three cells is marked on the layout. We can easily change the amount of variation and the number of quad-tree levels for different process technologies. Once we generate the gate length and threshold voltage of each cell in the SRAM, we perform a table-lookup on our pre-simulated leakage power values and use the same formula from Section 3.1.2 to calculate the leakage power for all chips.

4) Power Model For CAM Array: We can apply the same modeling approach to estimate the power for CAM arrays. We build three basic models just as we did for the SRAM, and the primary difference is that we use 10T CAM cells in the models and include match circuitry (e.g. the match line precharge circuit). In addition to read and write power, we need to estimate match power when the CAM is searched. Once a match operation is performed, the match power will be consumed in the array as a function of the array size and the input pattern (number of matches). To account for such power, additional HSPICE simulations are performed on the three small blocks. Tab. I records all the HSPICE tests needed for power calculation.

Models for read and write energy are very similar to those derived for the SRAM model. However, although we can use the same method to derive the models, match operation energy models are more complex, because the power is a function of both the bitline and matchline energy and the number of hit operations. We also need to consider the case when there is no match.

Assuming M columns and N rows with H hits in the array, Eq. (14) and Eq. (15) show the formulas for calculating match energy (E_{mc}) and idle state leakage power (P_{leak_array}) for the entire CAM array.

$$\begin{aligned} E_{mc} = H_1(5N + 2M + 3HM - 3NM - 6H) \\ +H_2(2MN + 5H - 2HM - M - 4N) \\ +H_3(2H + MN - HM - M - N) \\ +M_2(2N + M + HM - 3H - MN) \\ +W_1(2H - N - HM) + W_3(HM + N - 2H) \end{aligned} \tag{14}$$

$$\begin{aligned} P_{leak\_array} = MN(L_2 + L_3 - 3L_1) + M(2L_1 \\ -L_3) + N(2L_1 - L_2) \end{aligned} \tag{15}$$

B. Decoder

The decoder model is also based on small circuit block simulation. We model a three-stage static decoder with two-input predecoders. The first decoder stage is an inverter driving two NOR gates for the predecoder with interconnect wire capacitance. By properly applying tests to this block that force the interconnecting node to change from 0 to 1, we obtain the average dynamic energy consumption for this stage including both the capacitance power and short circuit power. Similarly, we can extract energy consumption for the second and third stages. For the switching of the internal nodes in the decoder, we need to assume some statistical activity. We assume that there are a total of A-bits of decoder address input. For each address change, N_{st1} stands for the number of first stage inverters whose outputs undergo a 0 to 1 change. N_{st2} is the number of NOR gates whose outputs undergo a 0 to 1 change. For simple estimation, N_{st1} and N_{st2} can be estimated as 50% of the gate number. There is always one NAND gate switching from 0 to 1 for each address change. With these numbers, we can estimate the dynamic energy consumption for the whole decoder as shown in Eq. (16).

$$E_{decoder} = E_{d1} \times N_{st1} + E_{d2} \times N_{st2} + E_{d3} \tag{16}$$

To calculate the leakage power, we apply tests to the first and third stage to measure the leakage. Since the states of the circuit affect the leakage, we toggle all possible states of each circuit and extract an average leakage power number for the first and third stages. Eq. (17) defines the total leakage power.

$$P_{leak\_decoder} = P_{leak\_st1} \times A + P_{leak\_st3} \times 2^A \tag{17}$$

C. Sense Amplifier and Other Circuits

The method used for the decoder is also used for power estimation for all the other regular logic structures such as column multiplexers. Because of the analog nature of sense amplifiers and irregularity of control circuits, for these structures we use a purely empirical method, e.g. the sense amplifier is not decomposed into smaller building blocks.

We model a latch based voltage sense amplifier with a high input impedance differential stage providing very fast sense operation [9]. When this amplifier is enabled, the cross coupled inverters form positive feedback and pull the output to full rail swing in a very short time. The second advantage is that the high impedance differential input stage separates the input bitlines from the output so that no other separation is needed. The cross coupled inverters will automatically cut off the direct path once the output value is established. This is very important because the main power consumption for sense amplifier is direct path power.

If we define the leakage power as P_{leak_sense}, the direct path energy consumption during sensing as E_{direct_sense}, the read frequency as f_{read}, the power consumption of the sense amplifiers is $P_{sense} = (P_{leak\_sense} + E_{direct\_sense} \times f_{read})$.

IV. Simulation results

We compare the power estimation results of our modeling method with other power modeling approaches and with HSPICE simulation of post-layout structures (Fig. 5). We also show the robustness of the modeling approach across a range of memory configurations, temperatures, and process libraries.

If not explicitly mentioned, all simulation results are based on a 130nm technology node with a supply voltage of 1.3V, temperature of 100^oC, and frequency of 400MHz. We have performed all of our simulations after performing custom layout with commercial foundry design libraries. When studying variability, we assume 7% gate length and 10% threshold voltage variation. We exercise the array with 200M reads/s and 200M writes/s, with a 25% data toggle rate when writing. We compare our model with an analytical model, an empirical model, and complete designs simulated with HSPICE. To represent a typical empirical model scaling scenario, we design the empirical model for the 0.35 μm technology node with a voltage of 3.5V and frequency of 150MHz. The model is then scaled to our baseline technology node, supply voltage, and frequency using traditional scaling theory. The dynamic power model for our analytical method is similar to that used in CACTI [7]. For the analytical leakage power, we first match the leakage power with HSPICE simulation at 25^oC. We then use Eq. (3) to scale the leakage power with temperature up to 100^oC, where $V_t = KT/q$. We use the same fitting parameters as HotLeakage [10].

Fig. 6 displays the power estimation results for each component in the 64x32 SRAM design with 8 sense amplifiers. As the diagram shows, our model tracks closely with HSPICE simulation for each component in the SRAM for both leakage and dynamic power (3.6% offset), while other models tend to yield large errors (8%-14%).

Fig. 7 shows the simulation results for different SRAM array configurations and sizes. For each configuration, the power is normalized to the HSPICE results. Our modeling method tracks closely to the HSPICE circuit simulation, with small estimation error (maximum error of 4.1%) through different configurations and array sizes while the analytical models can yield large estimation error (maximum of 17% observed).

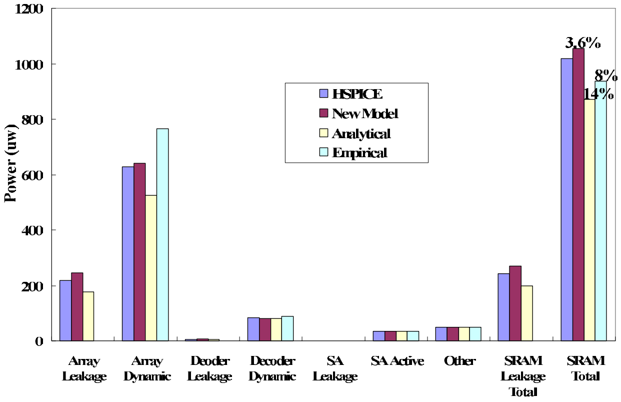


Fig. 6. Model comparison for SRAM components.

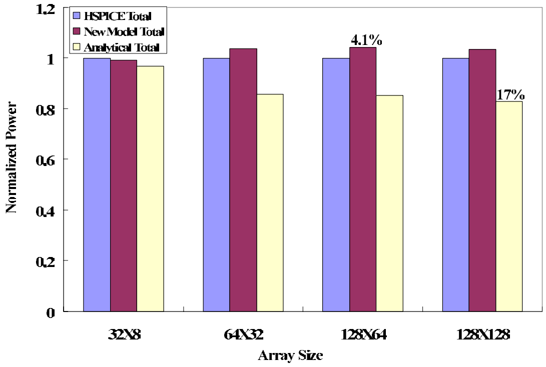


Fig. 7. Power for SRAM with different sizes.

Fig. 8 shows the relative error for a 64x32 SRAM across seven technology nodes. The power model is compared with full circuit HSPICE simulation. We use commercial technology libraries for 250nm, 180nm and 130nm, and predictive model (PTM) [11] for the remaining technology nodes. The total power is a combination of read, write, and leakage. Our model yields good results for each process node because it can easily use new technology libraries as the base of simulation. These additional models were quickly generated with a handful of small circuit simulation to extract the new energy data. The results show that our modeling approach can closely track detailed circuit simulation under different types of libraries (commercial and PTM). We also show the modeling error for the CAM model under the commercial libraries in Fig. 9.

Fig. 10 plots leakage power under different temperatures for a 64x32 SRAM and a 12x12 CAM. The figure shows that when temperature increases from 25^oC to 100^oC, the leakage power rises more than 6 times. For accurate power modeling, a model that can predict leakage power under varying temperature is critical. The analytical power model we use includes the exponential impact of temperature on power, but it still deviates by nearly 18% at 100^oC, even though we matched the leakage with HSPICE at 25^oC. Our approach captures temperature dependent effects with HSPICE simulation of the building blocks by including temperature as one of the simulation parameters.

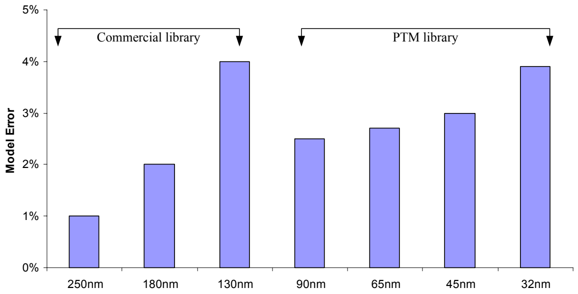


Fig. 8. SRAM relative error across technology nodes.

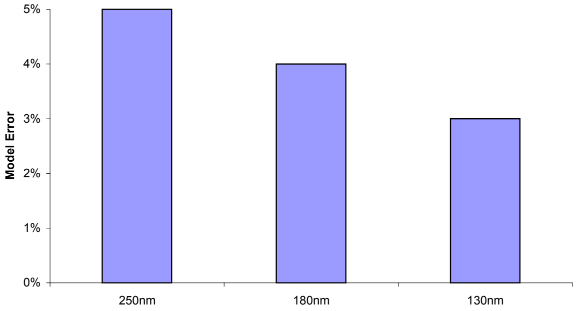


Fig. 9. CAM relative error across technology nodes under commercial library.

Fig. 11 shows the distribution of SRAM leakage power under process variations, normalized to the nominal leakage power. We see that some chips can have more than 12X larger leakage which will become a significant design issue in future technologies. Also, because of the complex relationship between the variation sources and leakage, the distribution is non-Gaussian emphasizing the difficulty in generating accurate distributions using purely analytical methods.

For each simulation, our power modeling method takes about 20 seconds to complete on a machine with a 2GHz CPU and 1GB memory. The simulation time is constant with the size of the modeled structure. In contrast, detailed HSPICE simulations for a 256x256 SRAM takes more than an hour on the same machine and increases rapidly with SRAM size.

V. Conclusion

This paper proposes a novel power modeling method and we demonstrate the effectiveness of the method for widely used SRAM and CAM structures. With the help of very simple circuit simulation on small blocks, we can achieve accurate and flexible power estimation results. We have compared this technique with HSPICE simulation and the other traditional power modeling methods. Since many modern CPU components are based on these array structures, we believe that this modeling method can be applied for accurate power estimation in early architecture design stage for microprocessors.

Acknowledgments

This work was funded in part NSF grant CCF-0048313, an IBM Faculty Partnership Award, and Intel.

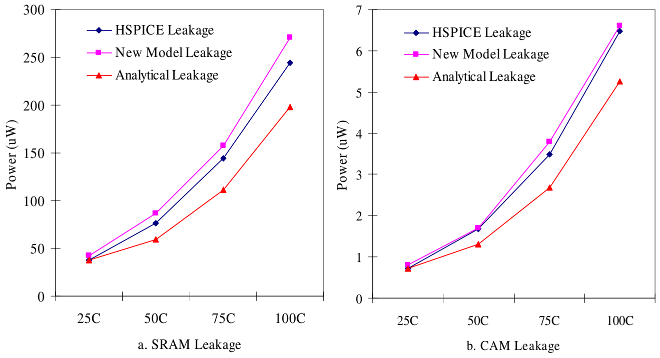


Fig. 10. Leakage power varying with temperature.

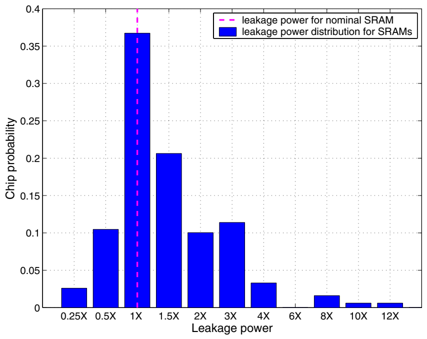


Fig. 11. Leakage power distribution under process variations.

References

[1] A. Agarwal, D. Blaauw, S. Sundareswaran, V. Zolotov, M. Zhou, K. Gala, and R. Panda. Path-based statistical timing analysis considering inter and intra-die correlations. In *Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, June 2002.

[2] D. Brooks, P. Bose, V. Srinivasan, M. Gschwind, P. G. Emma, and M. G. Rosenfield. New methodology for early-stage, microarchitecture-level power-performance analysis of microprocessors. *IBM Journal of Research and Development*, 47(5/6), 2003.

[3] D. Brooks, V. Tiwari, and M. Martonosi. Watch: A framework for architectural-level power analysis and optimizations. In *27th Annual International Symposium on Computer Architecture*, June 2000.

[4] J. A. Butts and G. S. Sohi. A static power model for architects. In *33rd IEEE/ACM International Symposium on Microarchitecture*, Dec. 2000.

[5] M. Q. Do, M. Drazdziulis, P. Larsson-Edefors, and L. Bengtsson. Parameterizable architecture-level SRAM power model using circuit-simulation backend for leakage calibration. In *International Symposium on Quality Electronic Design*, Mar. 2006.

[6] M. Mamidipaka, K. Khouri, N. Dutt, and M. Abadir. IDAP: A tool for high level power estimation of custom array structures. In *International Conference on Computer Aided Design (ICCAD)*, Nov 2003.

[7] P. Shivakumar and N. P. Jouppi. Cacti 3.0: An integrated cache timing, power and area model. Technical Report 2001/2, Compaq Western Research Laboratory, Aug. 2001.

[8] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye. Energy-driven integrated hardware-software optimizations using SimplePower. In *27th Int'l Symp. on Computer Architecture*, June 2000.

[9] B. Wicht, T. Nirschl, and D. Schmitt-Landsiedel. A yield-optimized latch-type SRAM sense amplifier. In *33rd European Solid-state Device Research Conference*, September 2003.

[10] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects. Technical Report CS-2003-05, University of Virginia Department of Computer Science, Mar. 2003.

[11] W. Zhao and Y. Cao. New generation of predictive technology model for sub-45nm design exploration. In *IEEE International Symposium on Quality Electronic Design*, 2006.