A Methodology for Accurate Modeling of Energy Dissipation in Array Structures

Mahesh N. Mamidipaka Nikil D. Dutt
Center for Embedded Computer Systems
University of California, Irvine, CA 92697
{maheshmn,dutt}@cecs.uci.edu

Kamal S. Khouri
Arch. and Systems Platforms
Motorola Inc., Austin, TX 78729
Kamal.Khouri@motorola.com

Abstract

There is an increasing need for obtaining a reasonably accurate estimate of energy dissipation in SoC designs. Array structures have a significant contribution to the total system level energy consumption. In this paper, we propose a new methodology to develop analytical models for accurately estimating energy dissipation in array structures. The methodology is based on the characterization of arrays for energy as a function of micro-architecture level inputs. The coefficients of the function are extracted using circuit level simulations. We apply the proposed methodology to develop energy models for three different array structures used in the Motorola e500[1] *processor core. The models are validated by comparing them against post-layout SPICE simulation. The energy models are seen to be highly accurate with an error margin of less than 8%. While the experiments are specific to the e500 processor core based array structures, the methodology is generic and can be used to develop energy models for array structures of any SOC design.*

1. Introduction

Energy dissipation has become critical design factor in SOC designs. Until recently, energy dissipation was considered important only to portable systems as it determines the battery lifetime. However, there is a growing concern on energy dissipation in application domains such as high-speed desktop and network processing. In these systems, decreasing feature and die size and higher clock rates significantly increase the power density, thereby affecting system reliability and packaging costs. For desktop computing, lower yields and higher cooling system costs increases the over all cost of the system. In the network processor domain, heat removal systems have a fixed capacity, limiting the number of processors that can be placed in the server/switch farm. This demands an early and reasonably accurate estimate of the energy dissipated in a processor for a given application.

Traditionally power dissipation in micro-architectural blocks is estimated using gate level simulations for combinational logic and transistor level simulations for array structures. However, because of their long simulation times, these techniques are not practical for power estimation at an application level. To overcome this, researchers have focussed on developing analytical models as a function of micro-architecture level inputs. Estimation of energy dissipation in a processor at an application level helps in: 1. Rapid design space exploration by evaluating different power-performance tradeoffs for an application. 2. Possible elimination of hot-spots by determining the power distribution across modules for a given processor configuration and a given application. 3. Optimizing the software for reduction in power dissipation [10] for a given configuration

Array structures are an important class of modules in current day micro-processors. Not only do they consume significant portion of the die area, but also have a major contribution to the overall power dissipation in processors. It can be noted that the array structures (viz. register files, branch target buffer, reservation stations, tag arrays, caches) consume up to 70% of the total chip power [3]. Also, it has been shown that caches alone consume up to 40% of total power [4]. Existing models in the literature have focussed on estimating energy dissipation in array structures based on conventional style of implementation. The accuracy of these models is limited considering the wide variety of possible implementation styles in array structures. In this paper, we propose a methodology for modeling energy dissipation in existing array structure designs with an emphasis on accuracy. We show that for different implementations of array sub-blocks, the energy dissipation can be characterized as a function of primary inputs and outputs. The coefficients of the function are then extracted using accurate circuit level simulations. The methodology yields analytical models which take micro-architecture level inputs(write data, write address, read data, read address etc.) as parameters. The methodology is generic and can used for modeling arrays belonging to any processor/system designs because of their similar inherent structure. To the best of our knowledge, this is the first work which proposes a generic methodology for modeling energy dissipation in array structures over wide variety of possible implementation styles.

The paper is organized as follows. The related work is discussed in Section 2. Section 3 presents an overview of different array structures used in current SOC designs with regard to their operations and their implementation details. The methodology used for extracting the energy models is de-

[1] e500 is the Motorola processor core that is compliant with the PowerPC Book E architecture

Proceedings of the 16th International Conference on VLSI Design (VLSI'03)
1063-9667/03 $17.00 © 2003 IEEE

scribed in Section 4. This methodology is applied to three different array structures, each corresponding to different styles of implementation. The accuracy of the models are evaluated by comparing the estimated values against the SPICE based simulation results. Finally, we conclude in Section 6 with brief summary and future work directions.

2 Related Work

Traditionally, power dissipation in array structures has been estimated using circuit level simulators such as SPICE, Powermill, and Quickpower [1, 2]. Even though highly accurate, these simulators cannot be used for power estimation at the application level because of impractical run-times. To enable power estimation at the application level, there have been research efforts to analytically model the power consumption in array structures [3, 5, 6, 17]. Kamble and Ghose [6] proposed analytical models for estimating energy dissipation in conventional caches and low power caches. In these models, only the power dissipated due to bitlines and wordlines are considered. The power consumed due to control logic, decode logic, and sense amplifiers is considered negligible in these models. While this assumption might hold for some implementations of caches, the assumption does not hold for array structures customized for performance or power. Energy models specific to caches have also been proposed by Li and Henkel[7]. The models are similar to those proposed by Kamble and Ghose except that the energy dissipation due to decoder, output drivers, and memory write is accounted based on some statistical assumptions not described in the paper. A simplified energy model for caches is proposed by Su and Despain[13] and is enhanced in [11, 12]. We think that these models are more suitable for exploration purposes where relative comparisons are more important than accuracy. For a variety of register file implementations, power dissipation is studied elaborately and analytical models are proposed by Zyuban and Kogge [17]. A comparison of different approaches for modeling energy dissipations was studied by Evans et al. [5]. However, the approaches were not meant for estimation of power at the application level. The accuracy of these analytical models is not known as there have been very few efforts to compare the model estimates with the actual power dissipation. Furthermore, the models have been proposed for specific array structures with specific implementation styles.

Brooks et al. [3] proposed parameterizable analytical models to estimate power for different sizes of generic array structures. These models are based on the capacitance values estimated using the Cacti [15] tool. While these models can be used to get an early estimate of power even before the array is designed, they do not give good estimates for a variety of implementations. This is a drawback when doing accurate power estimation since array power is highly dependent on the implementation style. For example, the power consumed due to an inverter based sense-amplifiers is significantly different from the power consumed due to a differential based sense-amplifier. In addition, the accuracy of the analytical models is further limited by the lack of accountability for power dissipation due to leakage and short circuit currents. To overcome these issues, in this paper, we propose a simulation based modeling of energy dissipation in array structures.

3 Array Structures

Array structures used in microprocessor design fall under two broad categories [9]: Random Access Memories (RAMs) and Content Addressable Memories (CAMs). RAMs are array structures that are used to store and retrieve data randomly (e.g., cache tag arrays, register files, branch predictors, instruction windows). A CAM is a hardware implementation of associative processing. Associative processing manipulates data based on matching, or associating an input value with other values stored in the array. Examples include TLBs, instruction window wake up logic, and load/store order checks. Array structures are primarily composed of row/column decoders, wordline drivers, memory arrays, bitlines, sense-amplifiers, and the control circuitry. However, there are different ways in which each of these sub-blocks can be implemented. A typical design of an array structure is shown in Figure 1.

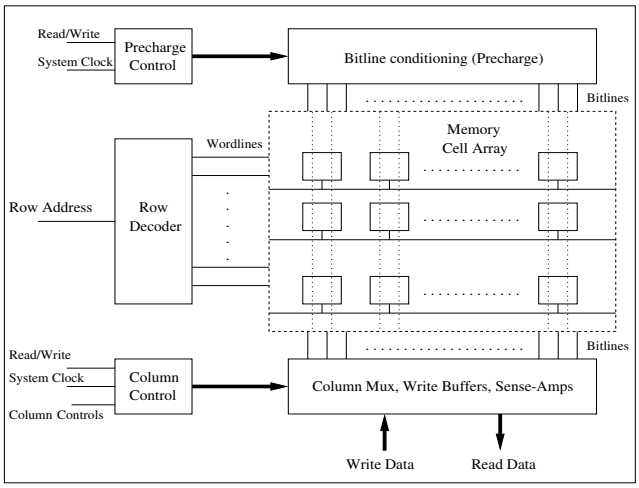

Figure 1. Typical Architecture of Array Structures

For a read/write operation, the row decoder selects the appropriate wordline which activates the row in the memory array. The bitlines are precharged by the bitline conditioning module which is controlled by the precharge control. For a read operation, the bitlines either retain charge or discharge depending on the data stored in the cells selected by the wordline. The sense-amplifier detects the changes in the voltage on the bitlines and the appropriate data is multiplexed to the data output. However, for a write operation, the sense-amplifiers and bitline conditioning are isolated by the control logic. The write buffers drive the bitlines according to the write data before being written into the memory location corresponding to the write address. CAMs differ from RAMs because of the support for an additonal *match* operation. The match operation compares an input data with the internally stored data. So the CAM cell consists of a conventional static RAM cell

with additional transistors which form an XOR gate and a distributed NOR pull-down [9].

During an array access, energy is dissipated in each of the array sub-blocks. The energy dissipated in each sub-block depends on the type of operation (read, write, or match), the implementation style, and the size of the array. The total access energy for an operation is the sum of the energy dissipated in the decode logic, E_{Decode}, memory cells, $E_{Memcells}$, bitlines, $E_{Bitlines}$, sense-amplifiers, E_{SA}, write drivers, E_{Wrdri}, and the control circuitry that drives clocks and signals to these units, $E_{control}$. For array structures where multiple read and write operations may occur simultaneously, we assume that the energy of an operation is independent of other operations occurring concurrently and that the total energy is cumulative. It can be observed that for any given operation, in some cases the energy dissipation in a specific unit can be independent of input/output signals. For example, the energy dissipation on double-ended precharge based bitlines, is constant for any data write/read operation (one bitline in each pair makes a $1 \rightarrow 0$ transition). However, on single-ended driver based bitlines, the energy dissipation for a read/write operation is proportional to the transitions that occur on the bitlines. Figure 2 summarizes the energy dissipation dependencies in the array sub-circuits over different implementation styles and for different operations.

Module	Implementation Type	Write Operation	Read Operation	Match Operation
Decode	Static logic	$\propto$ # of internal and addr. bus trans.	$\propto$ # of internal and addr. bus trans.	$\propto$ # of internal and addr. bus trans.
	Dynamic logic	Const.	Const.	Const.
Bitlines	Driver based	$\propto$ # of read/write data trans. + Const.		--
	Single-ended Precharge based	$\propto$ # of 1/0 's in read/write data + Const.		--
	Double-ended Precharge based	Const.	Const.	Const.
Memory Cells	Static SRAM	$\propto$ # of trans. in each mem-cell + Const.	Const. (Lkg)	Const.
Sense-Amps	Inverter based	Const. (Lkg)	$\propto$ # of trans. in rd data + Const.	Const. (Lkg)
	Differential based	Const. (Lkg)	Const.	Const. (Lkg)
Control	Static logic	$\propto$ # of internal trans. + Const.	$\propto$ # of internal trans. + Const.	$\propto$ # of internal trans. + Const.

Figure 2. Energy Dissipation in Array Structure Sub-modules for different operations

For a match operation in arrays, the data input to the memory cells is typically provided using double-ended bitlines. This reduces the comparison logic needed in each cell because of the availability of both data and its inversion. The energy dissipation on the match line for a match operation would be a constant amount of energy. However, this constant energy would be different for 'matched' and 'mismatched' data. For most sub-blocks in arrays, the energy dissipation is either a constant or a linear function of primary input/outputs. Because of the combinational logic involved in the design of address decoder and control circuitry, the energy dissipation in these sub-blocks is proportional to their internal transitions. Nevertheless, the internal transitions in the static CMOS based address decoder can be estimated based on the current address and the previous address. In the case of a control unit, a significant amount of energy dissipation is a constant because of the huge capacitive switching on clock and control signals for precharging bitlines and dis-engaging the sense-amplifiers for write operations.

The basis behind the methodology proposed in this paper is to evaluate these proportionality and constant coefficients that determine the energy dissipation in sub-blocks. We achieve this by performing SPICE simulations on the array for different operations and analyzing the energy dissipated in each sub-block corresponing to each operation. The energy dissipation for the whole array being the sum of the energy dissipated in the sub-blocks. Since the models are based on circuit level simulations, they include the energy dissipations due to short circuit and leakage currents along with currents due to capacitive node transitions and hence yield highly accurate estimates.

The remainder of this section describes the analytical equations used for calculating the internal transitions in an address decoder. A decoder typically consists of two levels of decoding and a set of buffers at each level for optimal delay purposes. Buffers are required after predecoding because of the high fanout at the pre-decode outputs. The second set of buffers are required for driving the high capacitive wordlines. While the energy dissipation in the pre-decode depends on the transition activity at the primary inputs, energy dissipation in the second level of decoding depends on the transition activity at the predecode outputs. Because of the symmetrical and regular structure of pre-decoder, for static CMOS based pre-decoder implementation, the transition activity at the pre-decode outputs can be expressed in terms of the primary inputs. For example, number of $0 \rightarrow 1$ transitions at the pre-decode output for a 9 bit address decoder (9-512 decoder) implemented using 3-8 decoders can be calculated as:

$$T_{int} = (Addr_i[0-2] \& Addr_{i-1}[0-2]) + (Addr_i[3-5] \& Addr_{i-1}[3-5]) + (Addr_i[6-8] \& Addr_{i-1}[6-8])$$

where $Addr_i$ represents the address in current clock cycle and $Addr_{i-1}$ representing the previous address. The '&' operation returning 1 if the operands are different and 0 if they are same. The analytical equation signifies that since the predecoder is implemented of smaller decoders, a transition at the output occurs only if there is transition at the input of the smaller decoder.

4 Methodology

In this section we describe the methodology used for extracting the energy model coefficients. Since the accuracy of the estimates depend on the accuracy of the model coefficients, we use SPICE based circuit simulations with back annotation of node capacitances from layout to extract these coefficients. The flow of our coefficient extraction methodol-

ogy is shown in Figure 3.

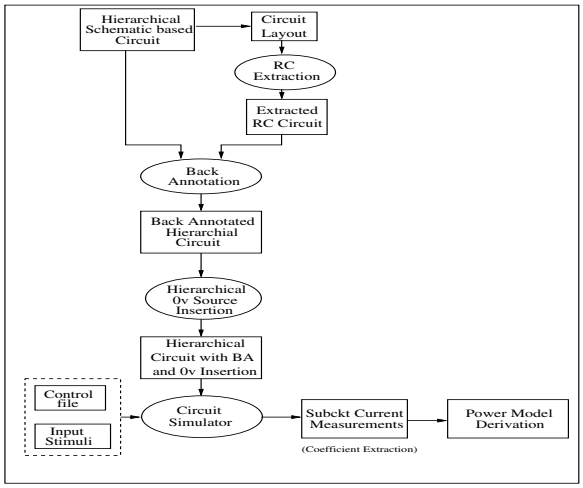

Figure 3. Methodology for Characterization of Array Structures

The flow starts with the hierarchical schematic based netlist of the circuit under characterization. The netlist is then back annotated with node capacitances obtained from the RC extraction of the corresponding circuit layout. The resulting back annotated hierarchical circuit netlist is further processed to enable the measurement of currents drawn by each sub-circuit in the netlist. This is done by inserting a zero voltage source at the voltage supply of each sub-circuit. The current drawn by the sub-circuits of interest can be determined by measuring the current through these voltage sources. The energy dissipation in the sub-block is computed as the integral of current during the operation period multiplied by the supply voltage. The circuit is then simulated for each operation supported by the circuit. The input vectors are selected so as to trigger transitions in the circuits which enable characterization of the sub-blocks. The currents drawn by each sub-circuit because of the activity in the sub-blocks are measured through the control statements indicated in the control file. Also, an additional vector is provided in the simulation to measure the steady state currents drawn by each sub-circuit. The measurements are then analyzed and analytical models are derived based on these measurements. Note that since these measurements are based on circuit level simulations, the energy dissipation due to short circuit and leakage currents also are accounted for.

In circuits where the memory arrays are very large, the RC extraction of the whole array is impractical. In such cases, only a few wordlines, bitlines, and memory cells over which the input vectors operate may be extracted. This method would still be accurate because of the symmetry of the bitlines and wordlines in the memory arrays and the variation of capacitive loads across these lines being minimal.

5 Experimental Designs and Results

In this section, we show the designs over which the proposed methodology has been applied and evaluate the accuracy of the extracted analytical models. In all the experiments, we assume the minimal capacitive loading on the output nodes. Since the actual output capacitive loading is not known until the placement and routing of the whole chip is done, we assume that there would be a separate analytical model to capture the energy dissipation due to the interconnect loading and the corresponding drivers. To cover the wide variety of implementation types in array structures we use three different array structures to show the efficiency of the methodology. The basic cell structure and the implementation types of these designs are indicated below. For write operations, the memory cell power depends on the data being currently held in that location. Since it is not possible to determine the existing data, we assume that the probability of transition in a memory cell is 1/2. The control and stimulus file necessary for the circuit simulation requires a reasonable understanding of the design since it has to capture the currents drawn by each sub-circuit under each supporting array operation. The control files existing, the SPICE circuit simulations required for model generation for each design took 2-5 hrs depending on its size. However, it is to be noted that this is only a one time effort needed for generating the analytical models. The extracted models are validated for accuracy by running simulations over traces of address and data obtained from application programs.

5.1 Simple Register File

This is a small register file design with 16 rows and 44 columns without any row decode logic. The array is directly fed with wordline selects. However, a read enable and write enable signals are fed to the array to enable/disable the reads and writes. The enable logic is implemented using dynamic CMOS logic and hence has read and write clocks for precharging. The energy dissipation on wordlines are included in the E_{Enable} since the wordline drivers are part of enable logic sub-circuit. There is no control logic in this design because of the simple precharge and the sense-amplifier mechanisms used. The structure of the basic memory cell of this register file is shown in Figure 4. The array has the following features:

- Dynamic CMOS based enable logic
- 1 Read and 1 Write port
- Single ended bitline for read using inverter based sense-amplifiers
- Double ended bitlines for write using write buffers

The energy dissipation models based on simulations for read and write operations are shown in Equations 1 and 2 respectively. E_{Con} indicates the constant energy dissipation due to the leakage currents in the array sub-circuits. E_{Pchclk} indicates the energy dissipation due to the capacitive loading on the precharge clock signal. Note that in case of a read operation, the energy dissipation due to the enable logic, read clock, precharge clock, and leakage is constant. The energy dissipation on read bitlines depends on the number of 1's in the data being read because of the discharging of precharge voltage. Coefficient E3 represents the energy dissipated in restoring

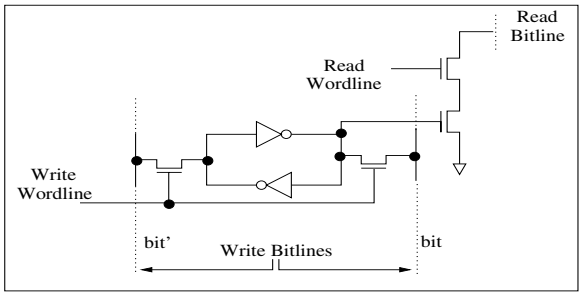

Figure 4. Schematic of the Memory Cell in Simple Register File

to the precharge state when such discharge happens on a single bitline and sense-amplifier. Similarly in case of a write operation, since the write bitlines are driven by static CMOS drivers, the dissipation depends on the number of transitions on the input of the write data. Coefficient E3 in Equation 2 represents the energy dissipation due to a single transition on input data. Table 1 shows the percentage of the total energy dissipation for each sub-circuit for reads and writes for this design and the percentage error compared to actual measurements. The coefficient values and the actual energy dissipation numbers are not indicated because they are Motorola proprietary data and cannot be published.

$$E_{rd} = E_{Enable} + E_{Rdclk} + E_{Bitline+SA} + E_{Pchclk} + E_{Con}$$
$$= E_1 + E_2 + (E_3 * (\#\ 1's\ in\ rd\ data) + E_4 + E_5$$
$$= K_1 + K_2 * N_{1's} \quad (1)$$
$$E_{wrt} = E_{Enable} + E_{Wrclk} + E_{Bitline+Drv} + E_{Memwrt} + E_{Con}$$
$$= E_1 + E_2 + E_3 * (\#\ of\ trans.\ on\ write\ data) +$$
$$E_4 * (\#\ of\ mem\ trans.\ due\ to\ write) + E_5$$
$$= K_1 + K_2 * N_{data\_trans} + K_3 * N_{mem\_trans} \quad (2)$$

	Read		Write	
Sub-circuit	%total	%error	%total	%error
Enable	31.2	-1.0	16.1	-1.0
Clocks	22.3	0.0	5.7	0.0
Bitlines+SA/DRV	45.5	-6.0	68.3	-2.8
Memory_cells	1.0	1	9.9	-3.3
Total energy	–	-3.1	–	-2.4

Table 1. Energy distribution and Accuracy for a Simple Register File

Note that the models are highly accurate and have an estimation error margin of 3.1%. As expected the bitlines consume the most energy, as much as 45.5% and 68.3% of the total energy consumption. The enable logic contribution is quite significant since it includes the wordline energy dissipation.

5.2 Multi-ported General Purpose Register File

This is a 32 row and 64 column General Purpose Register(GPR) file with multiple read and write ports. This is the most commonly used array structure in super-scalar based processor architectures. The address decoder of this GPR is implemented using dynamic CMOS logic and the memory cells are accessed for reads and writes through single-ended bitlines. The schematic of the memory cell in this general purpose register file is shown in Figure 5. The main features of this GPR are:

- 5 Read and 2 Write ports
- Dynamic CMOS logic based address decoder
- Single ended bitline for read using inverter based sense-amplifiers
- Single ended bitline for write using buffers

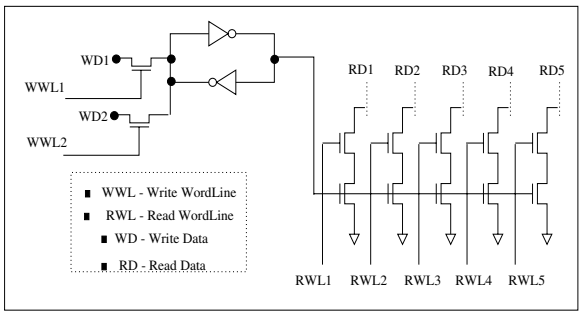

Figure 5. Schematic of the Memory Cell in Multi-ported Register File

The energy dissipation models based on simulations for read and write operations are shown in Equation 3 and 4 respectively. As can be noted, similar to the simple register file, the energy dissipation during the read operation for the address decoder, clocks are constant. However, unlike the simple register file, for the read operation, the bitline energy depends on the number of 0's in the read data because of the difference in the implementation of the memory cell. Table 2 shows the percentage of the total energy dissipation for each sub-circuit for reads and writes for this design and the percentage error compared to actual measurements. It can be noted that because of the larger size of the array compared to simple register file, the percentage contribution of energy dissipation due to bitlines is more significant. The memory array cell energy estimates has the largest error margin (18%). However, since the percentage contribution of memory cell energy to the total energy is not significant, the error incurred due to this approximation over the total power estimation is seen to be negligible.

$$E_{rd} = E_{Dec} + E_{Rdclk} + E_{Bitline+SA} + E_{Pchclk} + E_{Con}$$
$$= E_1 + E_2 + (E_3 * (\#\ 0's\ in\ rd\ data) + E_4 + E_5$$
$$= K_1 + K_2 * N_{0's} \quad (3)$$
$$E_{wrt} = E_{Dec} + E_{Wrclk} + E_{Bitline\_Drv} + E_{Memwrt} + E_{Con}$$
$$= E_1 + E_2 + E_3 * (\#\ of\ trans.\ on\ write\ data) +$$
$$E_4 * (\#\ of\ mem\ trans.\ due\ to\ write) + E_5$$
$$= K_1 + K_2 * N_{data\_trans} + K_3 * N_{mem\_trans} \quad (4)$$

	Read		Write	
Sub-circuit	%total	%error	%total	%error
Decode	5.1	-2.0	18.7	+2.8
Clocks	10.7	0.0	4.5	0.0
Bitlines+SA/DRV	81.1	-8.6	66.4	-8.3
Memory_cells	3.1	-9	13.6	+18.2
Total energy	–	-7.4	–	-2.6

Table 2. Energy Distribution and Accuracy for a Multi-ported GPR

5.3 Conventional memory array

This array structure has 64 rows and 2 sets of 40 columns. The design supports 20-bit read and write operations. The appropriate columns for read and write operations are selected through logic implemented in control circuitry. The control logic also controls the signals and clocks to column multiplexers, precharging logic, and differential sense-amplifiers. The row decoder is implemented with static CMOS logic and hence there would be energy dissipation in the decoder only when there is change in row address. The reads and writes to the memory locations are implemented through double ended bit lines with a single access port. The main features of this array structure are indicated below:

- Static CMOS based row decoder
- 1 access port (same port for both reads and writes)
- Double ended precharge based bitlines with reads using differential sense-amplifiers

The schematic of the memory cell in this conventional array structure is shown in Figure 6. The energy dissipation models based on simulations for read and write operations are shown in Equation 5 and 6 respectively. It should be noted that unlike simple register file or the GPR the energy dissipation on bitlines is constant for both reads and writes because of the double ended precharge based implementation. Also Equation 5 implies that the energy dissipation due to a read operation in a conventional array structure is constant if there is change in the row address. Table 3 shows the percentage of the total energy dissipation for each sub-circuit for reads and writes for this design and the percentage error compared to actual measurements. Note that the energy dissipation on wordline is included in the decoder energy. The simulation based estimates are highly accurate and are seen to have less than 1% error margin for both reads and writes. As was mentioned in the previous section, the dominant error in estimation of memory cell power for a write operation is because of the approximation in transition activity because of a data write.

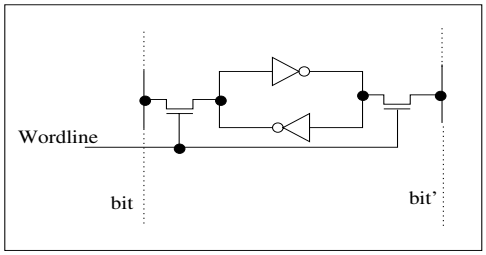

Figure 6. Schematic of the Memory Cell in Conventional Array Structure

$$
\begin{aligned}
E_{rd} &= E_{Dec} + E_{Bitlines+SA} + E_{Cntrl} + E_{Con} \\
&= E_1 * (\#\ of\ row\ addr.\ changes) + E_2 + E_3 + E_4 \\
&= K_1 * N_{row\_chg} + K_2 \qquad (5) \\
E_{wrt} &= E_{Dec} + E_{Bitlines} + E_{Cntrl} + E_{MemWrt} + E_{Con} \\
&= E_1 * (\#\ row\ addr.\ changes) + E_2 + E_3 * (\#\ trans\ on \\
&\quad col.\ addr.) + E_4 * (\#\ mem\ trans.\ due\ to\ wrt) + E_5 \\
&= K_1 + K_2 * N_{row\_chg} + K_3 * N_{col\_trans} \\
&\quad + K_4 * N_{mem\ trans} \qquad (6)
\end{aligned}
$$

	Read		Write	
Sub-circuit	%total	%error	%total	%error
Decode	3.3	-2.0	4.9	0
Control	35.1	-8.7	18.7	-4.2
Bitlines+SA/DRV	61.5	+6.1	74.9	+0.5
Memory_cells	0.2	-1	1.5	-14.2
Total energy	–	-0.7	–	-0.7

Table 3. Energy Distribution and Accuracy for a Conventional Array Structure

6 Conclusions and Future Work

In this paper, we proposed a methodology to generate simulation based analytical models as a function of primary input and outputs. These models can be used for accurate processor power estimation at an application level and for micro-architecture level design space exploration. Experimental results show that the models are highly accurate with an error margin of less than 8%. The simulation times for extracting the power models is in the order of few hours. However, this is only a one time effort needed for extracting the analytical models. Although the experiments were done on the Motorola e500 processor core based array structures, the methodology is generic and is applicable to array structures of any system design. The energy models generated using this methodology are oriented towards the already existing designs and for a specific technology. Future work will involve developing models which are a function of technology and implementation styles. Also, we plan to develop a characterization methodology for other micro-architectural components.

References

[1] Mentor graphics corporation, 1999.
[2] Synopsys corporation. powermill data sheet, 1999.
[3] D. Brooks, et al. Wattch: a framework for architectural-level power analysis and optimizations. In *ISCA*, pages 83–94, 2000.
[4] J. Montanaro. et al. A 160-MHz, 32b, 0.5-W CMOS RISC microprocessor. In *IEEE JSSC*, pages 1703–1712, Nov' 96.
[5] R. Evans, et al. Energy consumption modeling and optimization for srams. In *IEEE JSSC*, pages 571–579, 1995.
[6] M. Kamble, et al. Analytical energy dissipation models for low power caches. In *ISLPED*, 1997.
[7] Y. Li and J. Henkel. A framework for estimating and minimizing energy dissipation of embedded HW/SW systems. In *DAC*, 1998.
[8] H. Mehta, et al. Energy characterization based on clustering In *DAC*, pages 702–707, 1996.
[9] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley, 1985.
[10] K. Roy, et al. Software design for low power. In *NATO ASI Series on Low Power Design in DSE*, Aug. 1996.
[11] W-T. Shiue and C. Chakraborthy. Memory exploration for low power, embedded system In *DAC*, pages 140–145, 1999.
[12] P. Hicks, M. Walnock, and R. Owens. Analysis of power consumption in memory hierarchies In *ISLPED*, pages 239–244, 1997.
[13] C. Su and A. Despain. Cache design tradeoffs for power and performance optimization: a case study In *ISLPED*, pages 63–68, 1995.
[14] N. Vijaykrishnan, et al. Energy-driven integrated hardware-software optimizations using simplepower. In *ISCA*, pages 95–106, 2000.
[15] S. Wilton and N. Jouppi. *An enhanced access and cycle time model for on-chip caches*. WRL Research Report 93/5, June 1994.
[16] W. Ye, et al. The design and use of simplepower: A cycle-accurate energy estimation tool. In *DAC*, pages 340–345, 2000.
[17] V. Zyuban, et al. The energy complexity of register files. In *ISLPED*, pages 305–310, 1998.