AN ANALYTICAL, TRANSISTOR-LEVEL ENERGY MODEL FOR SRAM-BASED CACHES

Nikos Bellas, Ibrahim Hajj and Constantine Polychronopoulos,
Coordinated Sciences Laboratory,
1308 W. Main Str.
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: {nikos,hajj}@uivlsi.csl.uiuc.edu

Abstract

In this paper, we present a detailed, transistor-level energy model of on-chip caches that use SRAM technology. The energy estimation is based on the work by by Wilson and Jouppi [1], in which they propose a timing analysis model for SRAM-based caches. Our model uses run-time information of the cache utilization (number of accesses, number of hits, misses, input statistics etc.) gathered during simulation, as well as complexity and internal cache organization parameters (cache size, block size, associativity, banking etc.).

1 Introduction

The growing complexity of VLSI circuits [2] and the increasing operating frequency have led to unacceptably high levels of energy and power consumption in general purpose, high-performance processors. It is, therefore, important to have a thorough understanding of the power distribution in a processor to be able to design for low power. Previous published work corroborate the fact that the on-chip caches consume a substantial fraction in the energy budget of today's microprocessors. For example, the on-chip caches of the DEC 21164 Alpha processor consume about 25% of the total power of the chip [3], the on-chip caches of the StrongARM processor, which target low power applications, dissipate 43% of the total power, and in Pentium Pro they dissipate about 20% of the power.

Accurate energy models are, thus, deemed necessary as a prerequisite for design for low-energy caches. Related work has been done in [4], [5], and [6], among others. Most of these proposed techniques utilize models that consider the structure of the cache and use information about the utilization statistics of the cache. Our model is more detailed, and considers internal banking of the cache so that both the access time and the energy are reduced. The model equations are also very flexible since they allow the estimation of the energy of the cache under different parameters of cache complexity. We have succesfully applied this model to estimate the energy dissipated in the memory hierarchy of high performance processors [7].

The paper is organized as follows: in section 2 we describe the internal cache organization. In section 3 we derive the analytical energy model for each component of the cache. Experimental results are presented in section 4.

2 Internal Cache Organization

Fig. 1 from [1] shows the assumed internal cache organization. This is a very general model of an A-way set associative cache, with size of C bytes and a block size of B bytes. The operation of the cache is now briefly described.

The cache is organized as a collection of $S = \frac{C}{B \times A}$ sets, so that one set contains A blocks, or $B \times A$ bytes. The CPU issues an address to the cache consisting of three parts: the tag, the index and the offset. The index part has length $\log_2(S)$ bits, and is used to index the set from which the data will be retrieved. The offset part has length $\log_2(B)$ bits, and is used to select the appropriate word within a block to return to the CPU. Finally, the tag part is used to check whether there is a hit or a miss in the cache.

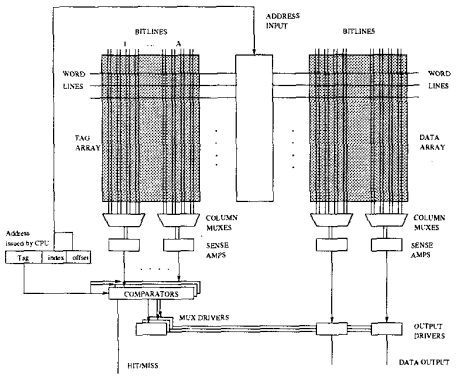

Figure 1: Cache Model

The cache consists of two arrays used to store the tag and the actual data. Each one of them is organized as a series of rows and columns so that there is one CMOS Static RAM cell at the intersection of a row and a column. In Fig. [1] we assume that one row in the data array stores a single set. The decoder first selects a row from the tag and data array using the index and offset bits of the CPU address. Each bitline is first precharged high. When the decoder makes the selection, each memory cell in that row pulls down one of its two bitlines, depending on the value of the cell.

A set of sense amplifiers monitors small changes in the bitlines and transforms them into legitimate voltage values. Usually, a sense amplifier is shared among several pairs of bitlines. Extra column multiplexers are used in both arrays to implement this sharing.

The information read from the tag array is compared to the tag bits of the address issued by the CPU. There are A such comparators for an A-associative cache. The result of the comparison is used to drive the output bus with the data that have been read, in the meantime, from the data array.

In most of today's caches, the tag and data arrays are broken row-wise and columnwise so that the time to access the data is reduced. Three new parameters are defined for that purpose for each of the two arrays. The parameter N_{dwl} shows how many times the data array is split vertically resulting into more and shorter wordlines. The parameter N_{dbl} shows how many times the data array is split horizontally resulting into more and shorter bitlines. Finally the parameter N_{spd}

0-7803-5471-0/99/$10.00©1999 IEEE

indicates how many sets are mapped into a single row. The tag array can be broken independently according to the parameters N_{twl}, N_{tbl}, and N_{tspd}.

Using these organizational parameters, each data subarray has $\frac{8 \times B \times A \times N_{spd}}{N_{dwl}}$ columns and $\frac{C}{B \times A \times N_{dbl} \times N_{spd}}$ rows. The total number of data subarrays is $N_{dbl} \times N_{dwl}$.

3 Energy Models for the Cache

Each one of the components of the cache are now analyzed in detail with respect to energy dissipation. The gate capacitance of a device or gate x is denoted as $C_{g,x}$ and its junction capacitance as $C_{j,x}$ from now on.

3.1 Energy Dissipation in the Decoder

Figure 2 shows the decoder architecture. The address from the CPU is partitioned to sets of three bits and is used to drive a set of 3-to-8 decoders. Each of these decoders asserts one out of eight outputs. The NOR gates have as many inputs as the number of 3-to-8 decoders. The output of the NOR gates are used to drive the selected wordlines through a wordline buffer. The 3-to-8 decoder can be implemented using eight, three-input NAND gates.

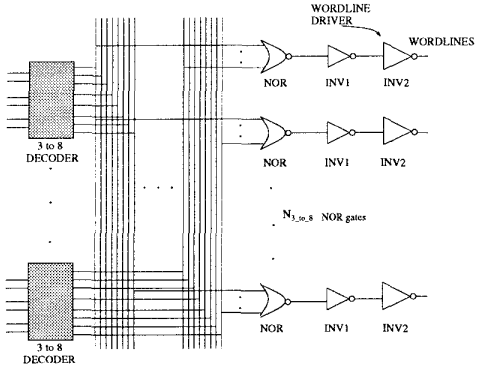

Figure 2: Address Decoder

In the actual implementation, the 3-to-8 decoders are actually shared among four subarrays as shown on Fig. 3, and are called collectivelly pre-decoders. The energy dissipation in the input of the pre-decoder is given by:

$$E_{pred\_input} = \frac{1}{2} \times V_{dd}^2 \times N_{d,inp} \times 2 \times [4 \times \lceil \frac{N_{dwl} N_{dbl}}{4} \rceil \times C_{g,pred\_inp} + 2 \times B \times A \times N_{dbl} \times N_{spd} \times C_{wordmetal}] + \frac{1}{2} \times V_{dd}^2 \times N_{d,inp} \times 2 \times [4 \times \lceil \frac{N_{twl} N_{tbl}}{4} \rceil \times C_{g,pred\_inp} + 2 \times B \times A \times N_{tbl} \times N_{tspd} \times C_{wordmetal}]$$

where $N_{d,inp}$ is the total number of transitions in the input address (computed during simulation). Note that there are $\lceil \frac{N_{dwl} N_{dbl}}{4} \rceil$ pre-decoders in the data subarray, and each address input bit or its complement is connected to all eight NAND gates.

For completion, we also give the energy dissipation due to the interconnect capacitance of the metal wires that transfer the address bits to the pre-decoder inputs. The wire length can be approximated by noting that the total edge length of the data array is approximately $8 \times B \times A \times N_{dbl} \times N_{spd}$. Using Fig. 3, we see that the length of the interconnect wire is about one quarter of this length.

The second contribution in the energy dissipation of the decoder comes from the junction capacitances of the NAND gates and the gate capacitances of the NOR gates (Fig. 2. It is given by:

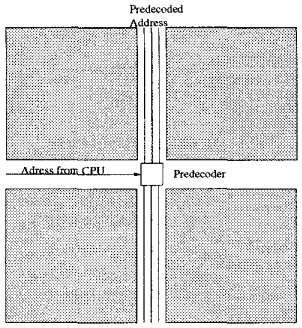

Figure 3: Banking of the Cache

$$E_{row\_dec} = \frac{1}{2} \times V_{dd}^2 \times N_{acc} \times \{2 \times \lceil \frac{N_{dwl} N_{dbl}}{4} \rceil \times N_{3\_to\_8,data} \times [C_{j,NAND} + \frac{C}{8 \times B \times A \times N_{dbl} \times N_{spd}} \times C_{g,NOR} + \frac{C}{2 \times B \times A \times N_{dbl} \times N_{spd}} \times C_{bitmetal}]\} + \frac{1}{2} \times V_{dd}^2 \times N_{acc} \times 2 \times C_{j,NOR} + \frac{1}{2} \times V_{dd}^2 \times N_{acc} \times \{2 \times \lceil \frac{N_{twl} N_{tbl}}{4} \rceil \times N_{3\_to\_8,tag} \times [C_{j,NAND} + \frac{C}{8 \times B \times A \times N_{tbl} \times N_{tspd}} \times C_{g,NOR} + \frac{C}{2 \times B \times A \times N_{tbl} \times N_{tspd}} \times C_{bitmetal}]\} + \frac{1}{2} \times V_{dd}^2 \times N_{acc} \times 2 \times C_{j,NOR}$$

where N_{acc} is the number of accesses in the cache, $N_{3\_to\_8,data}$ is the number of the 3-to-8 decoders for each data subarray, and $C_{bitmetal}$ is the capacitance per bit of the interconnect between the NAND and NOR gates.

The factor $\frac{C}{8 \times B \times A \times N_{dbl} \times N_{spd}}$, which is equal to (Number of rows in the subarray)/8, is used because each NAND gate in the pre-decoder drives that many NOR gates. Note also that the interconnect capacitance is also taken into consideration.

Each one of the 3-to-8 decoders modifies only two of its outputs for every access. In addition, only one NOR gate in each array evaluates its output to one, since only one wordline can be selected for each cache access (the term that contains the $C_{j,NOR}$ takes care of that effect). We also multiply by two since one wordline switches from one to zero and another from zero to one in every cache access.

3.2 Energy Dissipation in the Word Lines

In every clock cycle, a wordline will be charged and another one will be discharged (Fig. 4). The energy dissipation in the word lines is given by:

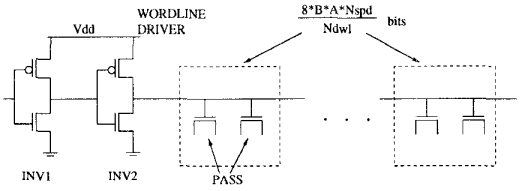

Figure 4: Word line architecture

$$E_{wordline} = V_{dd}^2 \times N_{acc} \times N_{dwl} \times (C_{j,INV1} + C_{g,INV2}) + V_{dd}^2 \times N_{acc} \times (8 \times B \times A \times N_{spd} \times 2 \times C_{g,PASS} + C_{j,INV2} + 8 \times B \times A \times N_{spd} \times C_{wordmetal}) + V_{dd}^2 \times N_{acc} \times N_{twl} \times (C_{j,INV1} + C_{g,INV2}) + V_{dd}^2 \times N_{acc} \times [(T + St) \times N_{tspd} \times 2 \times C_{g,PASS} + C_{j,INV2} + (T + St) \times N_{tspd} \times C_{wordmetal}]$$

The length of the tag is T bits, and there are also St bits for the status in each block. For example, the *valid* and the *dirty* bit are status bits. Each data subarray has $2 \times 8 \times B \times A \times N_{spd}$ pass transistors, and each tag subarray has $2 \times (T + St)$ pass transistors.

3.3 Energy Dissipation in the Bit Lines

The energy dissipated in the bit lines is due to the precharge phase, as well as to the read or write phase during which one of the two bitlines for every cell is discharged to a logic low value. We assume that the logic swing of the bitline is $\frac{1}{2}V_{dd}$. The contribution of this component to the energy dissipation is given by:

$$E_{bitline} = \frac{1}{2}(\frac{1}{2}V_{dd})^2 \times (N_{dbit,pr} \times C_{dbit,pr} + N_{dbit,r} \times C_{dbit,r} + N_{dbit,w} \times C_{dbit,w} + N_{tbit,pr} \times C_{tbit,pr} + N_{tbit,r} \times C_{tbit,r} + N_{tbit,w} \times C_{tbit,w} + \frac{1}{2} \times (N_{prech\_log} \times N_{acc} \times C_{prech})$$

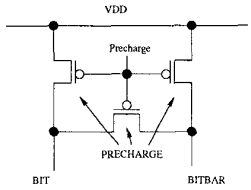

Figure 5: Bit line precharge circuit

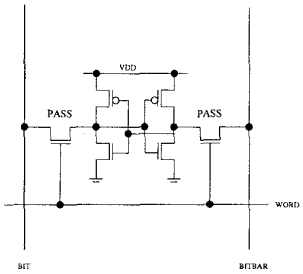

Figure 6: Memory cell

where the following notations have been used:

- $N_{dbit,pr} = N_{acc} \times 2 \times 8 \times B \times A \times N_{spd} \times \frac{1}{2}$ is the total number of precharges in all the bitlines in the data array. Note that only half of them switch to a logic high during precharge, since only those half have switced to logic low in the previous clock cycle.
- $N_{dbit,r} = N_{acc} \times 2 \times 8 \times B \times A \times N_{spd} \times \frac{1}{2}$ is the total number of read transitions in the bit lines.
- $N_{dbit,w} = N_{whit} \times (St + W_{averg,data\_write}) + \frac{1}{2} \times N_{rmiss} \times (2 \times 8 \times B \times A \times N_{spd})$ is the total number of write transistions in the bit lines. N_{whit} is the total number of write hits in the cache, and N_{rmiss} the total number of read misses.
- $N_{tbit,pr}, N_{tbit,r}, N_{tbit,w}$ are defined similarly for the tag array.
- $N_{prech\_log} = 8 \times B \times A \times N_{spd} \times N_{dbl} + A \times (T + St) \times N_{tspd} \times N_{tbl}$ is the number of bitline precharge circuits in the cache.
- $C_{dbit,pr} = C_{dbit,r} = C_{dbit,w} = \frac{C}{B \times A \times N_{dbl} \times N_{spd}} \times (\frac{1}{2} \times C_{j,PASS} + C_{bitmetal}) + 2 \times C_{j,PRECH} + C_{j,MUX}$ is the switched capacitance for each precharge (or read or write) transition. The junction capacitance of the pass transistor is multiplied by two since it is shared between two vertically adjacent cells. The $C_{j,MUX}$ is the junction capacitance of the column multiplexer at the other end of the bitline.

3.4 Energy Dissipation in the Comparators

Each one of the A comparators for an A-associative cache is designed using dynamic logic (Fig. 7), and it has a width of T bits.

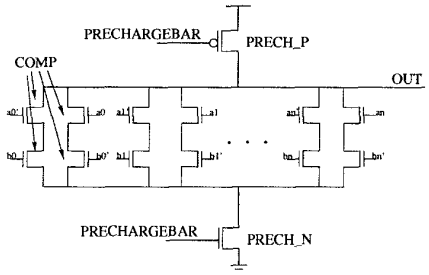

Figure 7: Comparator

The energy dissipation in the A comparators is given by:

$$E_{comp} = \frac{1}{2} \times V_{dd}^2 \times N_{acc} \times A \times (C_{j,PRECH\_P} + C_{g,MUX\_DRV}) + \frac{1}{2} \times V_{dd}^2 \times N_{misses} \times A \times [T \times 2 \times C_{j,COMP} + C_{j,PRECH\_N} + T \times C_{j,COMP} + C_{g,MUX\_DRV}] + \frac{1}{2} \times V_{dd}^2 \times N_{hits} \times (A-1) \times (T \times 2 \times C_{j,COMP} + C_{j,PRECH\_N} + T \times C_{j,COMP} + C_{g,MUX\_DRV}] + \frac{1}{2} \times V_{dd}^2 \times N_{hits} \times [T \times C_{j,COMP} + T \times 2 \times C_{j,COMP}] + \frac{1}{2} \times V_{dd}^2 \times N_{tag,inp} \times A \times 2 \times T \times C_{g,COMP}$$

where $N_{tag,inp}$ is the number of switches in the input of the evaluation transistors. We assume that in case of a miss, half of the paths to the ground will be on and half will be off. When a path is on, we have to multiply the junction capacitance of a transistor by two to approximate the capacitance switched. In case of a hit, only one out of the A comparators evaluates to one.

3.5 Energy Dissipation in the Multiplexer Drivers

Fig. 8 shows the context of the multiplexor drivers in the cache systems. Each multiplexor is responsible for controlling the multiplexing of the $8 \times B$ bits from each cache block onto the data bus that reads out b_0 bits towards the CPU or the main memory. There is one such multiplexer driver for each one of the A comparators in an A-way set associative cache. The implementation of a multiplexer is shown in Fig. 9

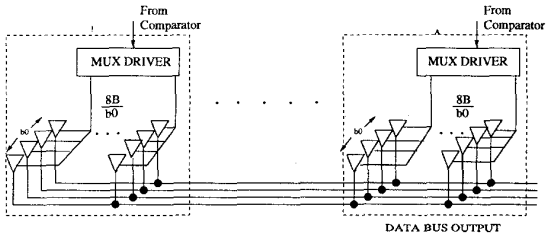

Figure 8: Data Bus outpus for multiplexer drivers

The energy dissipation in the multiplexer drivers is given by:

$$E_{mux\_driver} = \frac{1}{2} \times V_{dd}^2 \times N_{hits} \times (\frac{8 \times B}{b_0} \times C_{g,NOR} + C_{j,INV}) + \frac{1}{2} \times V_{dd}^2 \times N_{hits} \times (C_{g,DRV\_INV} + C_{j,NOR}) + \frac{1}{2} \times V_{dd}^2 \times N_{hits} \times (b_0 \times C_{g,OUTDRV} + C_{j,DRV\_INV} + 4 \times B \times A \times N_{spd} \times N_{dbl} \times C_{wordmetal})$$

3.6 Energy Dissipation in the Output Bus

The energy dissipation in the output busses towards the CPU and the main memory is given by:

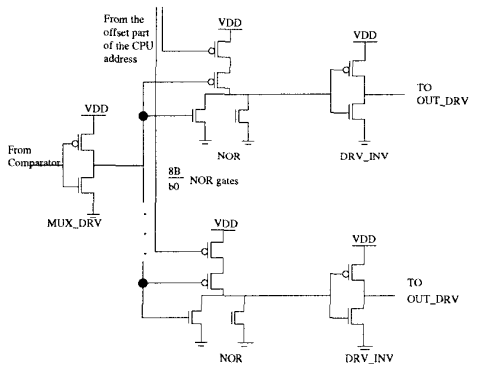

Figure 9: One of the A multiplexer drivers

$E_{output} = \frac{1}{2} \times V_{dd}^2 (N_{out,a2m} \times C_{out,a2m} + N_{out,d2m} \times C_{out,d2m} + N_{out,d2c} \times C_{out,d2c})$

where

- $N_{out,a2m} = \frac{1}{2} \times (N_{rmiss} + N_{whit} + N_{wmiss}) \times W_{addr\_bus}$, is approximately the total number of zero to one transitions in the address bus from the cache to the next level of memory (L2 cache or main memory). The address bus switches in case of a read miss, in case of a write hit (to write the data to the main memory), and in case of a write miss (to get tha data from the main memory). The address bus has W_{addr_bus} bits length.

- $N_{out,d2m} = \frac{1}{2} \times (N_{whit} + N_{wmiss}) \times W_{data\_bus}$, is approximately the total number of zero to one transitions in the data bus from the cache to the next level of memory. The data bus switches each time that data needs to be written in the memory.

- $N_{out,d2c} = \frac{1}{2} \times (N_{reads} \times W_{avg\_read})$, is approximately the total number of zero to one transitions in the data bus from the cache to the CPU. The factor W_{avg_read} is the average number of bits read from the cache.

- The capacitances for off-chip transfers are set equal to 20pF, while the capacitances for on-chip transfers are set equal to 0.5pf [1].

4 Experimental Results

The models presented in the previous section were used in conjunction with a simulator of the MIPS1 architecture to obtain realistic estimation of the energy consumption in on-chip caches. The gate and junction capacitances were obtained from [1] for a 0.8um process.

The number of accesses in the cache, number of hits, misses, as well as the switching activity in the CPU address were computed by simulation of a set of SPEC95 benchmarks. The following benchmarks were simulated: *101.tomcatv, 102.swim, 103.su2cor, 104.hydro2d, 129.compress, 130.li, and 134.perl* (described in Table 1). Fig. 10 show distribution of energy consumption on the various components of our cache model for an 8-Kbyte direct mapped I-Cache with a block size of 16 bytes.

The bit lines are by far the most energy consuming part of the cache. This is mainly due to the very large capacitance of the precharge transistors, as well as to the length of the bitlines. In every clock cycle, half of the bitlines will precharge, and then half will discharge to logic zero.

The energy dissipation of the internal cell, of the column decoders and the sense amplifiers was negligible and was not taken into consideration.

Table 1: Statistics for the 32KB I-Cache utilization

Benchmark	*Accesses in billions*	*Hit rate*	*Energy in Joules*
tomcatv	4.79	99.95%	24.98
swim	0.39	99.99%	2.017
su2cor	10.13	99.97%	52.85
hydro2d	4.5	99.90%	23.50
compress	0.037	99.99%	0.20
li	0.21	99.99%	1.08
perl	2.56	98.76%	13.45

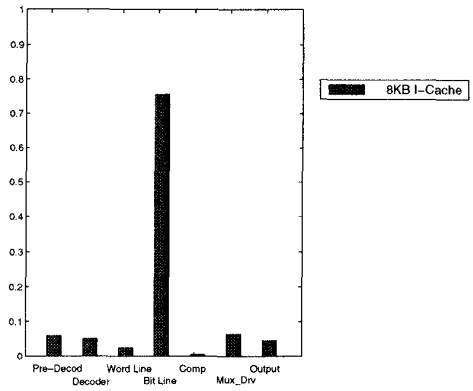

Figure 10: Distribution of energy dissipation in a direct mapped, 8KB I-Cache with a block size of16 bits

The model presented is a detailed characterization of SRAM-based caches which are routinely used for the implementation of the L1 and L2 memory hierarchy. The model can be used for fast, yet accurate estimation of the energy distribution in the various parts of the memory subsystem with different degrees of utilization characteristics and cache complexity.

References

[1] S. Wilson and N. Jouppi, "An enhanced access and cycle time model for on-chip caches," tech. rep., DEC WRL 93/5, July 1994.

[2] A. Veneris and I. Hajj, "Error Diagnosis and Correction in VLSI Digital Circuits," in IEEE Midwest Symposium on Circuits and Systems, pp. 1005-1008, 1997.

[3] J. Edmondon, "Internal organization of the Alpha 21164, a 300-MHz 64-bit quad-issue CMOS RISC microprocessor," Digital Technical Journal, vol. 7, no. 1, pp. 119-135, 1995.

[4] M. Camble and K. Ghose, "Analytical energy dissipation models for low power caches," in Proceedings of the International Symposium of Low Power Electronics and Design, pp. 143-148, Aug. 1997.

[5] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," IEEE Journal of Solid-State Circuits, vol. 29, pp. 663-670, June 1994.

[6] U. Ko, T.Balsara, and A. Nanda, "Energy optimization of multilevel cache architectures for RISC and CISC processors," IEEE Transactions on VLSI Systems, vol. 6, pp. 299-308, June 1998.

[7] N. Bellas, I. Hajj, C. Polychronopoulos, and G. Stamoulis, "Architectural and compiler support for energy reduction in the memory hierarchy of high performance microprocessors," in Proceedings of the International Symposium of Low Power Electronics and Design, pp. 70-75, Aug. 1998.