

Value-Conscious Cache: Simple Technique for Reducing Cache Access Power

Yen-Jen Chang
CS Department
National ChungHsing University
Taichung, Taiwan 402
ychang@orchid.ee.ntu.edu.tw

Chia-Lin Yang
CSIE Department
National Taiwan University
Taipei, Taiwan 106
yangc@csie.ntu.edu.tw

Feipei Lai
CSIE & EE Department
National Taiwan University
Taipei, Taiwan 106
flai@cc.ee.ntu.edu.tw

Abstract

Most microprocessors employ the on-chip caches to bridge the performance gap between the processor and main memory. However, the cache accesses usually contribute significantly to the total power consumption of the chip. Based on the observation that an overwhelming majority of the cache access bits are '0', in this paper we propose a value-conscious (VC) cache to reduce the average cache power consumption during an access. Unlike the conventional cache with differential-bitline implementation, the VC cache is a single-bitline design. Depending on the access bit value, the VC cache can dynamically prevent the bitline from being discharged such that the power dissipated in accessing '0' is much less than the power dissipated in accessing '1'. The implementation of the VC cache is a circuit-level technique, which is software independent and orthogonal to other low power techniques at architecture-level. The experimental results based on the SPEC2000 and MediaBench traces show that without compromise of both performance and stability, by exploiting the prevalence of '0' bits in access data the VC cache can reduce the average cache read and write power by about 18%~22% and 36%~40%, respectively.

1. Introduction

Power consumption is an increasingly pressing problem in modern system design, especially for the advanced microprocessors and the portable devices with battery powered. Because the on-chip caches are accessed frequently and usually implemented using arrays of densely packed SRAM cells for high performance, they consume a significant fraction of total chip power (e.g., 25% for DEC 21164 [1], and 43% for SA-110 [2]). Clearly, the caches are the most attractive targets for power reduction. Studies show that a large portion of cache power is dissipated in driving the long bitlines with large capacitance. The most important feature of the

conventional cache is that, on every access, regardless of the access data half of the bitlines would be discharged to low and then precharged to high before the next access. In other words, for each access the conventional cache power consumption is fixed and independent of the access data.

By examining the access data of the benchmark programs, we observe that an overwhelming majority of the access bits are '0'. Thus, we propose a *value-conscious (VC)* cache, which exploits the prevalence of '0' bits to reduce the average cache power consumption during an access. The VC cache can prevent the bitlines from being discharged in accessing '0', such that the power dissipated in accessing '0' is much less than the power dissipated in accessing '1'. Therefore, unlike the conventional cache where the power consumption is not sensitive to the '0' percentage of the access data, for each access the power consumption of the VC cache is varied with the '0' percentage of the access data. The more '0' bits the access data contain, the more power the VC cache can save.

The VC cache is a single-bitline design. In read access, the sense amplifier is the most critical adaptation to guarantee the VC cache can perform a read well. We develop a *zero-sensitive (ZS)* sense amp that is able to sense out the proper value using only one input. In write access, writing cell state from low to high is considerably more difficult in single-bitline configuration because it presents conditions similar to that of the read mode. Instead of the traditional boosted wordline technique [3], we develop a *zero-sensitive (ZS)* cell where a tail transistor is used to disconnect the pull-down path, such that writing cell state from low to high is easy to achieve in the VC cache with single-bitline implementation. To compensate the stability loss due to the asymmetrical inverter pair used in the ZS cell, the penalty to be paid is a 13% increase in cache area.

We evaluate the 0/1 distribution of the access data from the SPEC2000 and MediaBench benchmarks, and all of the power consumption data are obtained from the HSPICE simulation of the extracted layout in TSMC 0.35 μ m technology with a 3.3V supply. The experimental results show that by minimizing the power dissipated in accessing '0', the

This work was supported by the National Science Council of Taiwan under grant no. NSC91-2215-E-002-043 and NSC92-2213-E-002-014.

VC cache can reduce the average cache access power drastically. Compared to the conventional cache, without impairing the stability and performance, on average the VC cache results in around 20% reduction in the average cache read power and 37% reduction in the average cache write power.

The rest of this paper is organized as follows. Section 2 gives an overview of our motivation and the proposed VC cache. Next, in Section 3 we describe the circuitry developed for the VC cache, and then the impacts of the VC cache on stability, access delay, cache area and power reduction are provided in Section 4. The experimental results are given in Section 5, and Section 6 offers some brief conclusions.

2. Preliminary

2.1 0/1 Distribution

Fig. 1 shows the proportion of ‘0’ bits to the total reference bits (referred to as *zero rate*) examined from the execution traces of the benchmark programs. From this figure, over 80% of the instruction bits are ‘0’, and about 75% of the data bits are ‘0’. Because we use the PISA ISA, which is a 64-bit instruction format, to evaluate the 0/1 distribution, the 0/1 distribution of instruction bits is highly skewed towards zero. Unlike the conventional cache where the power dissipated in both accessing ‘1’ and ‘0’ are the same, motivated by the extremely asymmetric distribution of ‘0’ and ‘1’ bits, we propose a *value-conscious* (VC) cache, in which the power dissipated in accessing ‘0’ is much less than the power dissipated in accessing ‘1’. By exploiting the most of the access data are ‘0’ bits, the proposed VC cache can effectively reduce the average cache power consumption during an access.

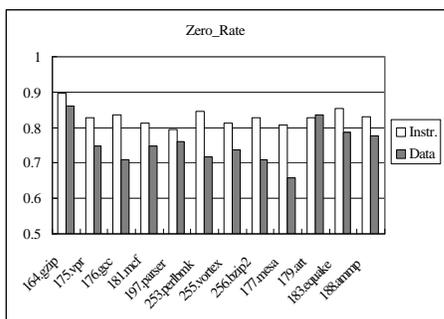


Figure 1. Zero rates of instruction and data references for SPEC2000.

2.2 Related Work

Traditionally, the cache read power can be reduced significantly by employing a pulsed-wordline technique [4] to turn off the wordline when a sufficient voltage differential has developed on the bitlines. However, it is not sensitive to the read value. The half-swing pulse-mode technique [5] was used to reduce the bitlines swing during cache writes by half of the conventional technique. However, using a $V_{DD}/2$ reference for bitlines potentially leads to cell instability during the cache reads. Due to the asymmetry distribution of ‘0’ and ‘1’ bits, a

single-ended read bitline [6] is proposed to minimize the number of bitline transitions. The bit cells of register file can be modified such that reading a zero causes no bitline discharge. In [7], a dynamic zero compression scheme was proposed to reduce the energy required for cache accesses by only writing and reading a single bit for every zero-value byte. The major disadvantage of the dynamic zero compression is that the power reduction is limited by the cluster of ‘0’ bits. This is especially unfavorable for instruction due to the instruction format. In contrast, the proposed VC cache can effectively reduce the cache access power without the necessity for the cluster of ‘0’ bits.

3. Value-Conscious (VC) Cache

3.1 Zero-Sensitive (ZS) Cell

To implement the VC cache, the memory cell must be adapted to zero-sensitive. The zero-sensitive means that the bitline discharge depends on whether the access bit value is ‘0’ or not. If the access bit is ‘0’, then the bitline should be prevented from being discharged. Otherwise, bitline discharge occurs as a typical access. Fig. 2 shows the schematic of the *zero-sensitive* (ZS) cell and its relative signals, where the read wordline (RWL) and write wordline (WWL) are used to select a cell for reading and writing, respectively, and the write select (WS) is used to facilitate the correct write. An access is initiated by precharging the data line (DL) to V_{DD} .

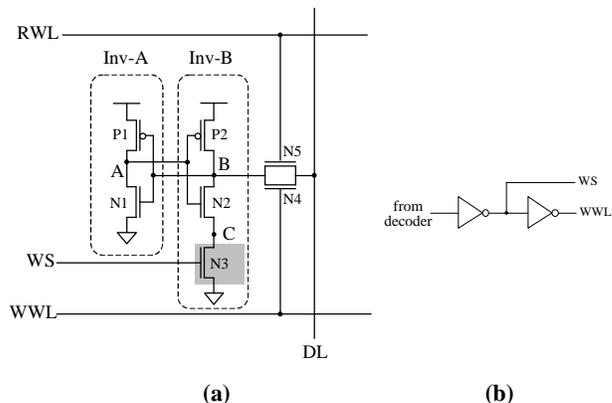


Figure 2. (a) Zero-sensitive (ZS) cell. (b) The generation of write select (WS) and write wordline (WWL) signals.

Read mode: In the read mode, WWL is set to 0 and thus the tail transistor $N3$ is turned on to activate the Inv-B. As RWL is asserted, the access transistor $N5$ connects DL to the cell. In reading ‘0’, because node B is held high, the DL would maintain a high voltage as in the precharge phase. This implies no bitline discharge incurred in reading ‘0’. In reading ‘1’, node A is held high to turn on the transistors $N2$ and $N3$. As RWL is asserted, the DL with initial high state would be discharged to low through transistors $N5$, $N2$ and $N3$.

Write ‘1’ mode: In the write ‘1’ mode, node B must be written to low that is done by setting DL to 0 and asserting WWL. The first possible case is writing the cell state from ‘1’

to '1' (1->1). Because both nodes B and DL are 0, no state transition arises in this case. Another possible case is 0->1. In this case because access transistor $N4$ has much larger conductance than $P2$, it is easy to flip the cell state from '0' to '1' by discharging node B through $N4$.

Write '0' mode: In the write '0' mode, node B must be written to high that is done by setting DL to V_{DD} and asserting the WWL . The first possible write pattern is 0->0. Because both nodes B and DL are high, no state transition arises in this case. Another possible write pattern is 1->0. In this case, if the ZS cell has no tail transistor $N3$ (that is equal to the conventional single-bitline cell), writing node B from low to high is considerably more difficult because it presents conditions similar to that of the read mode. The boosted wordline technique [3] is a traditionally solution to this problem, but the disadvantages are the potential instability and the hardware overheads.

Instead of the boosted wordline technique, in the ZS cell, we use a tail transistor $N3$ to facilitate writing node B from low to high. In this case, because $N3$ is turned off by WS before asserting WWL (as illustrated in Fig. 2(b)), the pull-down path of the Inv-B is disconnected. Therefore, it is easy to flip the cell state from '1' to '0' by charging node B through $N4$.

3.2 Zero-Sensitive (ZS) Sense Amplifier

The baseline sense amplifier design is a conventional latch sense amplifier with two inputs. Because the ZS cell is a single-bitline design, we must modify the conventional sense amplifier to be able to sense out the proper value using only one input, as shown in Fig. 3, which is called *zero-sensitive (ZS) sense amplifier*. Besides the DL input, the ZS sense amp uses another static input HI that is synchronized by precharge signal (the same as the precharge signal of the data line DL) and always precharged to high. Compared to the conventional sense amplifier, there are three additional NMOS transistors ($N2$, $N3$ and $N4$) in the ZS sense amp. There are two possible operations in the ZS sense amp: *differential sense* and *non-differential sense*.

(1) In reading '0' (DL retains high), because the nodes out and $-out$ follow the values of HI and DL that implies $out = -out = V_{DD}$, this case is *non-differential sense*. As the sense

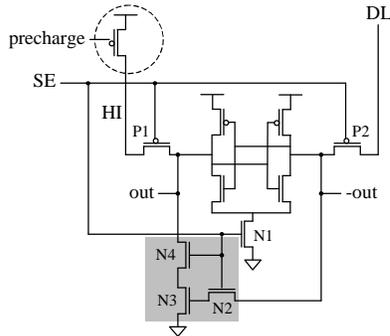


Figure 3. Zero-sensitive (ZS) sense amplifier.

enable (SE) signal is asserted, the transistor $N2$ conducts to pass $-out$ signal to turn on $N3$, and then the out signal is pulled down. Thus, the out voltage is lower than the $-out$ voltage. After the inverter pair amplifies this differential to a full rail-to-rail signal, the voltage of nodes out and $-out$ are 0 and V_{DD} respectively. As shown in Fig. 4(a) obtained from *HSPICE* post-layout simulation, the ZS sense amp would sense out '0' in the case of non-differential sense.

(2) The other operation is the differential sense. In this case, the value stored in the accessed cell is '1' that results in the DL discharge. When SE is asserted, the initial state is that the $-out$ voltage is lower than the out voltage. Due to the use of pulsed-wordline technique [4], the $-out$ voltage is about 2.5V~3V. Thus the pull-down transistor $N3$ is still turned on lightly by $-out$ signal, but it will soon be turned off. This is because the inverter pair amplifies the initial voltage differential such that the node $-out$ would be pulled down to 0 to turn off $N3$. Fig. 4(b) shows the ZS sense amp senses out '1' in the case of differential sense.

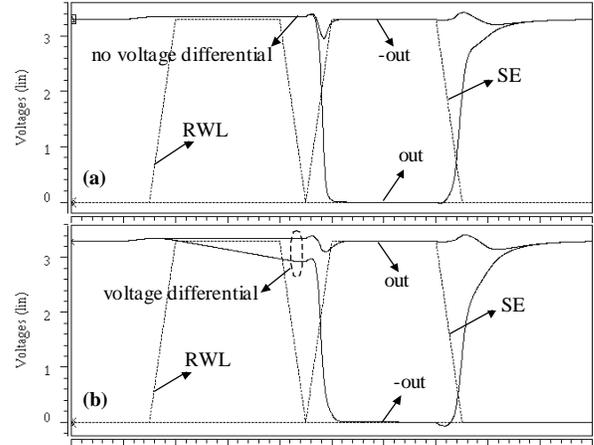


Figure 4. The *HSPICE* waveform of the ZS sense amplifier. (a) Non-differential sense. (b) Differential sense.

4. Stability, Access Delay and Power Reduction

This section provides the detailed analysis of the proposed VC cache from various criteria. We first estimate the impacts of the VC cache on the stability and access delay. With the same stability and performance as the conventional cache, the power reduction of the VC cache is provided.

4.1 Stability

The first consideration in the SRAM cell design is the stability that is the ability to hold a stable cell state. In general, the static noise margin (SNM) is an important parameter in determining the cell stability. As shown in Fig. 5(a), the major difference between the conventional cell and the ZS cell is the Inv-B, in which the additional tail transistor $N3$ results in an asymmetrical inverter pair that potentially degrades the stability. According to the results shown in [8], the SNM of the

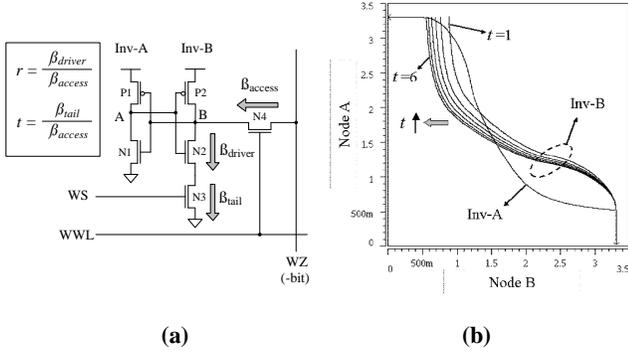


Figure 5. (a) The SNM of the ZS cell (SNM_{ZS}) is determined by the cell ratio r and tail ratio t . (b) Graphical representation of the SNM_{ZS} . It increases with the tail ratio t if the cell ratio is fixed ($r=1$ in this case).

conventional SRAM cell (SNM_{Conv}) increases with the cell ratio r , defined by $r = \beta_{driver} / \beta_{access}$. β_{driver} and β_{access} are the W/L ratios of driver transistor ($N2$) and access transistor ($N4$), respectively. In the ZS cell, because the tail transistor $N3$ is on the critical path in driving node B to low, besides the cell ratio, the SNM of the ZS cell (SNM_{ZS}) is also determined by the ratio of β_{tail} to β_{access} , referred to as tail ratio $t = \beta_{tail} / \beta_{access}$, in which β_{tail} is the W/L ratio of tail transistor $N3$. Fig. 5(b) shows how the SNM_{ZS} varies with the tail ratio. The SNM_{ZS} would increase with the tail ratio if the cell ratio is fixed. Keeping the SNM_{ZS} the same as the SNM_{Conv} can be achieved by appropriate choice of r and t . Fig. 6 shows the SNM_{ZS} in different combinations of r and t . The key observation is that when the cell ratio is 3 and the tail ratio is 5, the SNM_{Conv} and SNM_{ZS} are almost the same value $654mV$.

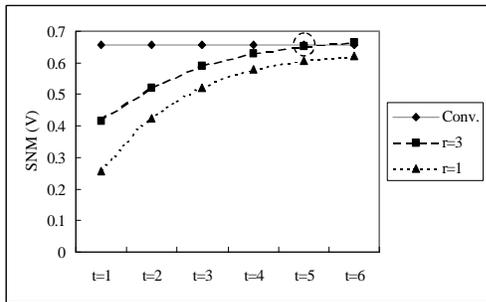


Figure 6. The SNM_{ZS} in different combination of cell ratio r and tail ratio t .

4.2 Access Delay

Read delay: We define the *read delay* as the elapsed time from asserting RWL to the correct sense output, which comprises the bitline discharge and sense time. Compared to the conventional sense amp, because the augmented transistors used in the ZS sense amp do not induce any visible sense delay, we only concern the bitline discharge time. In the VC cache, due to no bitline discharge incurred in reading '0', we only consider the case of reading '1', in which the DL would be discharged to low through the driver transistor $N2$ and tail

transistor $N3$ of Inv-B. Because $N3$ is always turned on in the read mode, similar to SNM, the read '1' delay also depends on both the cell and tail ratios. For a better SNM, the cell ratio is fixed to be 3 and Fig. 7 shows how the read '1' delay varies with the tail ratio. It is clear from this figure that when the tail ratio is 5, the read '1' delays of both the conventional and ZS cells are almost the same $1.23ns$. Thus, we conclude the VC cache has no read delay penalty.

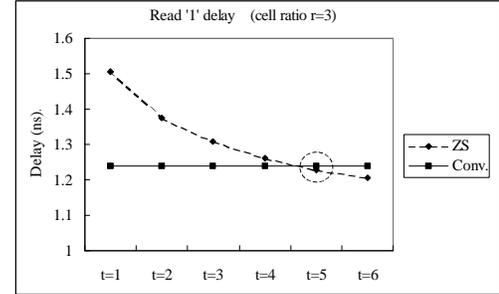


Figure 7. The read '1' delay varies with the tail ratio if cell ratio is fixed to be 3.

Write delay: The *write delay* is defined as the elapsed time from asserting WWL to the states of both nodes A and B become steady. There are four cases in write operation: writing the cell state from '0' to '0' (0->0), '0' to '1' (0->1), '1' to '0' (1->0) and '1' to '1' (1->1). Due to no state transition in cases of 0->0 and 1->1, we only consider the write delay in cases of 0->1 and 1->0.

(1) In the case of 0->1, by setting DL to 0 and then asserting WWL , node B with initial high state would be discharged to low. Compared to the traditional differential-bitline design, because in the ZS cell the state transition is driven by only one path, the 0->1 write delay of ZS cell is slightly larger than that of the conventional cell, as shown in Table 1. In determining write cycle, this minor difference can be ignored. (2) In the case of 1->0, by setting DL to V_{DD} and then asserting WWL , node B with initial low state would be driven to high to flip the state of node A . As shown in Table 1, because $N3$ is turned off in the write mode, the 1->0 write delay of the ZS cell is even smaller than that of the conventional cell.

Table 1. Write delay summary.

Write Delay (ns)	0->1 Write Delay	1->0 Write Delay
Conv.	0.7531	0.7533
SWDR	0.7587	0.7512

4.3 Power Reduction

Based on the analyses described above, we conclude that the ZS cell does not compromise either stability or access delay when the cell ratio is 3 and tail ratio is 5. In the ZS cell, WS signal is used to guarantee the correct write operation. Because it shares the load capacity of WWL , the additional WS does not induce any power penalty. Table 2 shows the column

power consumption for various access patterns. Note that in the conventional cache the power dissipated in reading ‘1’ and ‘0’ are the same. In contrast, the VC cache reduces the column power consumption by 54.44% in reading ‘0’. Because the VC cache is only beneficial to reading ‘0’, the additional transistors used in the ZS cell and ZS sense amp would induce a little more power consumption than the conventional cache in reading ‘1’. Similar to the read operation, in the conventional cell, regardless of write pattern, the column power consumptions are the same. Compared to the conventional cache, in the 1->0 write pattern, the VC cache reduces the column power consumption by 97.21%. Due to no state transition and bitline discharge, even 98.82% the column power reduction can be achieved in the 0->0 write pattern.

Table 2. Summary of column power consumption for various access patterns.

Column Power (mW)		Conv.	VC	Reduction
Read	0	3.04E-01	1.39E-01	54.44%
	1	3.04E-01	3.11E-01	-2.43%
Write	1->0	4.65E-01	1.30E-02	97.21%
	0->0	4.37E-01	5.17E-03	98.82%
	1->1	4.35E-01	4.20E-01	3.50%
	0->1	4.92E-01	4.64E-01	5.70%

4.4 Cache Area

From Fig. 3, clearly, the size of ZS sense amp is larger than the conventional sense amp. According to the cache area model presented in [9], we can ignore the area overhead introduced by the ZS sense amp because the area of sense amplifiers contributed to the total cache area is negligible. Compared to the conventional SRAM cell, the transistors used in the ZS cell is increased from 6 to 7. In addition, to compensate the stability loss due to the asymmetrical inverter pair in the ZS cell, we have to enlarge the cell ratio and tail ratio, and thus the cell area is increased from $87.21\mu\text{m}^2$ to $103.36\mu\text{m}^2$. Most area overhead is introduced by the large driver transistor $N2$ and tail transistor $N3$ that imposes around a 18.5% cell area overhead. By using *CACTI3.0* tool [9], we obtain the percentage of data-array of the total cache area is about 70% for a 32KB 2-way or 4-way cache. Thus, the overall cache area overhead is roughly $18.5\%*70\%=13\%$.

5. Experimental Results

We have assumed a baseline on-chip cache architecture with split instruction and data caches, which are a 32KB, 2-way instruction cache (IC) and a 32KB 4-way data cache (DC), respectively. To avoid an explosion in the number of simulations, the block size for both caches is 32 bytes. Instead of the conventional implementation, the baseline caches were implemented with the way-prediction scheme [10], which is an effective technique for reducing cache power. The way-prediction scheme reduces the cache power consumption by accessing only a single predicted cache way instead of accessing all the cache ways.

We use *SimpleScalar* [11] to evaluate the 0/1 distribution of the cache access data for both the *SPEC2000* and *MediaBench* benchmark suits. The *SPEC2000* is a suit of general-purpose programs, and the *MediaBench* is a suite of applications focus on multimedia and communications systems. To get a good mix of CPU-intensive and memory-intensive loads, we use four integer CINT2000 benchmarks (gzip, gcc, perlbnk, vortex), four floating-point CFP2000 benchmarks (mesa, art, equake, ammp) and three integer *MediaBench* benchmarks (adpcm, jpeg, gsm).

5.1 Access Pattern Distribution

Table 3 summarizes the access pattern distribution of both IC and DC for benchmarks. Because the result difference between the integer and floating-point benchmarks is hardly noticeable, we do not present these two benchmarks individually. From this table, the prevalence of ‘0’ bits in the cache access data is observed in both the general-purpose and multimedia programs. The higher percentage of access ‘0’ (including read ‘0’ and write ‘0’) means that the proposed VC cache is more efficient in reducing the cache power consumption during an access.

Table 3. Access pattern distributions of both IC and DC for benchmarks.

Pattern	SPEC2000		MediaBench		
	IC	DC	IC	DC	
Read	0	85.93%	77.18%	86.64%	65.27%
	1	14.07%	22.82%	13.36%	34.73%
Write	0->0	77.36%	71.52%	79.23%	92.40%
	1->0	7.72%	21.26%	6.35%	2.41%
	0->1	9.81%	3.45%	10.88%	4.25%
	1->1	5.10%	3.77%	5.74%	0.94%

5.2 Average Cache Power Consumption During an Access

Assume that the power dissipated in the cache data-array can be simplified as $P_{data_array}=P_{way}\times A$, in which P_{way} is the power consumption per cache way, and A is the degree of associativity. The power consumption per cache way is given by: $P_{way}=CP_{ave}\times N_{col}$. CP_{ave} is the average column power consumption and N_{col} is the number of columns per cache way. Because the block size is fixed 32 bytes, N_{col} is 256.

We define the *average read column power (ARCP)* and *average write column power (AWCP)* as the power dissipated in one column during each read and write, respectively. They are given by:

$$ARCP=(CP_0\times R_0)+(CP_1\times R_1) \quad (1)$$

$$AWCP=(CP_{0->0}\times R_{0->0})+(CP_{1->0}\times R_{1->0})+(CP_{0->1}\times R_{0->1})+(CP_{1->1}\times R_{1->1}) \quad (2)$$

CP_0 is the column power in reading ‘0’, and R_0 is the ratio of the read ‘0’ to all read bits. $CP_{0->0}$ is the column power dissipated in the 0->0 write pattern, and $R_{0->0}$ is the ratio of the 0->0 write pattern to all write operations. Depending on cache configuration, the column power consumption for various

Table 4. Column power consumption for various access patterns.

Column Power (mW)		IC (32K, 2-way)		DC (32K, 4-way)	
		Conv.	VC	Conv.	VC
Read	0	3.46E-01	1.57E-01	3.04E-01	1.39E-01
	1	3.46E-01	3.65E-01	3.04E-01	3.11E-01
Write	0->0	5.62E-01	5.43E-03	4.37E-01	5.17E-03
	1->0	5.65E-01	1.52E-02	4.65E-01	1.30E-02
	0->1	5.56E-01	5.32E-01	4.92E-01	4.64E-01
	1->1	5.61E-01	5.10E-01	4.35E-01	4.20E-01

access patterns are listed in Table 4. Applying the data shown in Tables 3 and 4 to Equations (1)(2), the cache power dissipated in data-array per access are obtained and summarized in Table 5, in which *Conv+WP* is the conventional cache implemented with way-prediction scheme.

Table 5. Cache power dissipated in data-array.

(a) Cache power dissipated in data-array per read access.

RP _{data_array} (mW)	Base (Conv+WP)	VC	
		SPEC2000	MediaBench
IC (32K-2W)	88.66	47.76	47.38
DC (32K-4W)	77.82	45.56	50.83

(b) Cache power dissipated in data-array per write access.

WP _{data_array} (mW)	Base (Conv+WP)	VC	
		SPEC2000	MediaBench
IC (32K-2W)	143.60	21.40	23.65
DC (32K-4W)	117.01	9.80	7.36

Note that the power dissipated in data-array is only a part of total cache power consumption. By using the *CACTI3.0* estimation tool [9], in the VC cache the considered partial components during a read, i.e., data bitlines and data sense amps, contribute roughly 44.2% and 53.6% to the total cache read power for the baseline IC and DC, respectively. The considered partial components during a write, i.e., only data bitlines, contribute 47.4% and 38.3% to the total cache write power for the baseline IC and DC, respectively. To obtain the power reduction of the entire cache, the power reduction in data-array must be multiplied by the above fraction parameter. Compared to the baseline conventional cache implemented with the way-prediction, the use of VC cache can result in roughly 18%~22% reduction in total cache power consumption during a read access, and reduces the total cache power consumption during a write access by about 36%~40%.

6. Conclusions

Low power cache memories have become a critical component in many applications, such as advanced microprocessors, handheld devices, embedded systems and SoC. Based on most of the cache access bits are '0', in this paper we propose a novel *value-conscious (VC) cache* to

reduce the average cache power consumption during an access. Unlike the conventional cache where the power dissipated in both accessing '1' and '0' are the same, an important feature of the VC cache is that the power dissipated in accessing '0' is much less than the power dissipated in accessing '1'. The experimental results show that while retaining the same stability and performance as the conventional cache, with a 13% area penalty the VC cache can reduce the average cache read power up to 22% and reduce the average cache write power up to 40%. As expected, with the advent of 64-bit architecture, the prevalence of '0' bits would be further enlarged. This implies the VC cache would be more efficient in reducing the cache access power.

References

- [1] J. F. Edmondson et al., "Internal Organization of the Alpha 21164, a 300-MHz 64-bit Quad-issue CMOS RISC Microprocessor," Digital Technical Journal, Vol. 7, No. 1, 1995, pp. 119-135.
- [2] J. Montanaro et al., "A 160 MHz, 32b 0.5W CMOS RISC Microprocessor," in IEEE ISSCC 1996 Digest of Papers, 1996.
- [3] M. Ukita et al., "A Single-Bit-Line Cross-Point Cell Activation (SCPA) Architecture for Ultra-Low-Power SRAM's," IEEE Journal of Solid-State Circuits, Vol. 28, No. 11, Nov. 1993, pp. 1114-1118.
- [4] B. Amrutur and M. Horowitz, "Techniques to Reduce Power in Fast Wide Memories," in Proc. of Symposium on Low Power Electronics, Oct. 1994, pp. 92-93.
- [5] K. W. Mai et al., "Low-Power SRAM Design Using Half-Swing Pulse-Mode Techniques," IEEE Journal of Solid-State Circuits, Vol. 33, No. 11, Nov. 1998, pp. 1659-1671.
- [6] J. Tseng and K. Asanovic, "Energy-Efficient Register Access," in Proc. of Symposium on Integrated Circuits and Systems Design, Manaus, Brazil, Sept. 2000.
- [7] L. Villa, M. Zhang and K. Asanovic, "Dynamic Zero Compression for Cache Energy Reduction," in Proc. of 33rd International Symposium on Microarchitecture Micro-33, 2000, pp. 214-220.
- [8] E. Seevinck, F. J. List and J. Lohstroh, "Static-Noise Margin Analysis of MOS SRAM Cells," IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 5, Oct. 1987, pp. 748-754.
- [9] P. Shivakumar and N. P. Jouppi, "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," COMPAQ WRL Research Report, 2001/2.
- [10] K. Inoue, T. Ishihara and K. Murakami, "Way-Predicting Set-Associative Cache for High Performance and Low Energy Consumption," in Proc. of ISLPED, 1999, pp. 273-275.
- [11] D.C. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0," Computer Architecture News, 25 (3), pp. 13-25, June, 1997. Extended version appears as UW Computer Sciences Technical Report #1342, June 1997.