Zero-Aware Asymmetric SRAM Cell for Reducing Cache Power in Writing Zero

Yen-Jen Chang, *Member, IEEE*, Feipei Lai, *Senior Member, IEEE*, and Chia-Lin Yang, *Member, IEEE*

***Abstract*—Most microprocessors employ the on-chip caches to bridge the performance gap between the processor and the 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 values written to the cache are "0," in this paper we propose a zero-aware SRAM cell with an asymmetric inverter pair, called *ZA* cell, to minimize the cache power consumption in writing "0." The ZA cell uses a circuit-level technique, which is software independent and orthogonal to other low-power techniques at architecture-level. Compared to the conventional SRAM cell, the experimental results based on the SPEC2000 and MediaBench traces show that without compromise of both performance and stability, the ZA cell can reduce the average cache write power consumption over 60% for both the baseline instruction and data caches. In particular, the ZA cell is attractive in the data caches, which reveal the high write-zero rate.**

***Index Terms*—Asymmetric, cache write power, low power, on-chip caches, SRAM cell, zero-aware.**

I. Introduction

As THE DEMAND for the portable devices and battery powered embedded systems continuously increases, the power consumption has become an important consideration in the microprocessor and system designs. Since the on-chip caches can effectively reduce the speed gap between the processor and main memory, almost modern microprocessors employ them to boost system performance. These on-chip caches are usually implemented using arrays of densely packed SRAM cells for high performance. Studies show that the power dissipated by the caches is usually a significant part of the total chip power. For example, in Alpha 21 164 processor, the caches consume about 25% of the total chip power [1], and StrongARM processor (SA-110), which targets on low-power applications, dissipates about 43% of the total chip power in the caches [2]. Clearly, the caches are the most attractive targets for power reduction.

Cache accesses include read and write operations. Because cache reads occur more frequently than writes, especially for the instruction cache, most low-power techniques focus on reducing the cache power dissipated in reading (i.e., cache read power). However, the write power is usually larger than the read power due to the large power dissipated in driving the cell bitlines to full swing. If a cache has a low hit ratio (that implies a large amount of cache writes), the power dissipated in the write operations become significant. Therefore, in this paper we concentrate on reducing the cache write power.

By examining the write data of the benchmark programs, we observe that an overwhelming majority of the write bits are "0." Based on this observation, we propose a novel zero-aware SRAM cell, called *ZA* cell, which drastically reduces the cache power dissipated in writing "0." The ZA cell consists of an asymmetric inverter pair, one write port with single bitline and one read port with differential bitlines. The most important characteristics of the ZA cell are summarized as follows. First, in the conventional SRAM cell, because one of two bitlines must be discharged to low regardless of written value, the power consumption in both writing "0" and "1" are the same. In contrast, the ZA cell uses the complement of input data to perform the write operation that prevents the single write bitline from being discharged if the written value is "0." Therefore, the write "0" power is far less than the write "1" power in the ZA cell. Second, 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. The boosted wordline technique [3] is a traditional solution to this problem, but it induces cell instability and hardware overheads. Instead of the boosted wordline technique, we use a tail transistor to disconnect the pull-down path, such that writing cell state from low to high is easy to be achieved in the ZA cell.

We evaluate the 0/1 distribution of the write 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 μm technology with a 3.3-V supply. The results show that by minimizing the power dissipated in writing "0," the ZA cell can reduce the average cache write power consumption up to 61% and 68% for the baseline instruction and data caches, respectively, while retaining the same stability and performance as the conventional cell with a cache area increased by 8.8%.

The rest of this paper is organized as follows. Section II presents our motivation and the initial examination of write data, which reveals the asymmetry distribution of "1" and "0" bits. In Section III, we describe the circuitry of the proposed ZA cell, and compare it to the related work. Then, the impacts of the ZA cell on the stability, access delay, cell area, and write power consumption are provided in Section IV. Experimental results are given in Section V, and Section VI offers some brief conclusions.

Manuscript received March 21, 2003; revised August 7, 2003.

Y.-J. Chang is with the Department of Computer Science, National Chung-Hsing University, Taichung 402, Taiwan, R.O.C. (e-mail: ychang@cs.nchu.edu.tw).

F. Lai is with the Departments of Computer Science and Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: flai@cc.ee.ntu.edu.tw).

C.-L. Yang is with the Department of Computer Science, National Taiwan University, Taipei, Taiwan, 402R.O.C. (e-mail: yangc@csie.ntu.edu.tw).

Digital Object Identifier 10.1109/TVLSI.2004.831471

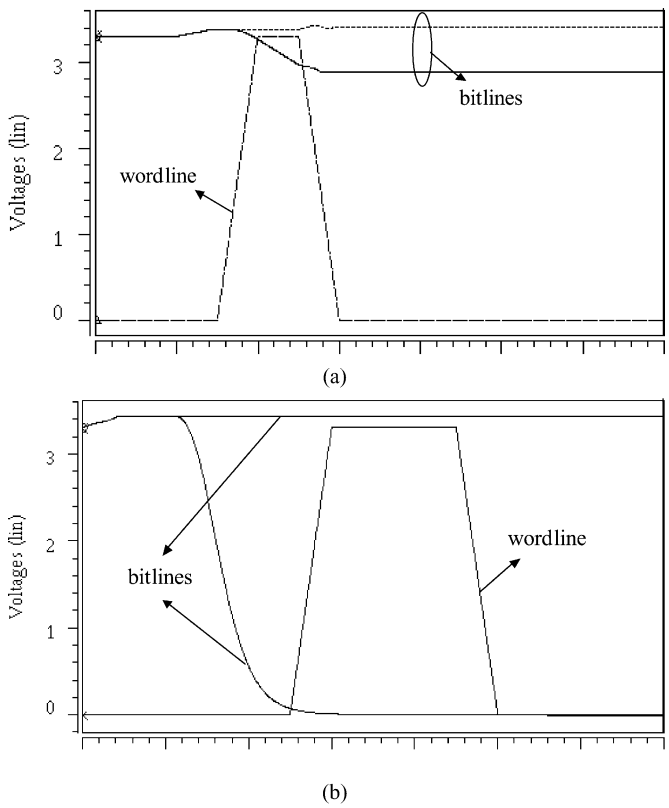


Fig. 1. HSPICE waveform of (a) cache read with pulsed-wordline technique and (b) conventional cache write.

II. Characteristics of the Cache Write Operation

In this section, we first identify the characteristics of the cache write operations for the application programs. We present the distribution of "0" and "1" bits in the write data, and then indicate the problems which we aim to tackle in this paper.

A. Cache Read versus Cache Write

In a cache, the major power-consuming components are bitlines, wordlines, sense amplifiers, decoders and output drivers. In general, the bitlines are the most power consuming component [4], [5]. As indicated in [4], for a 64-K 2-way associative cache, the bitlines are responsible for 40% of the total cache power consumption. This is because the large power dissipates in driving the long bitlines with large capacitance.

As shown in Fig. 1(a), the power consumption of cache read can be reduced significantly by employing a pulsed-wordline technique to turn off the wordline when a sufficient voltage differential has developed on the bitlines [6]. Our initial simulation shows that the pulsed-wordline technique can reduce the power dissipated in bitlines swing by about 80%. Compared to the cache read, in order to flip the cell state correctly, the cache write typically consumes considerably large power due to the full voltage swing on the bitlines, as shown in Fig. 1(b). The half-swing pulse-mode technique [7] was used to reduce the bitlines swing during cache writes by half of the conventional technique. However, using a $V_{\rm DD}/2$ reference for bitlines potentially leads to cell instability during the cache reads.

In the instruction cache (IC), because all accesses are cache reads, cache writes only occur in case of misses. In the data cache (DC), the cache writes also arise in the execution of STORE instruction besides the cache misses. The proportion of cache writes to reads is about 1 : 2 in DC [8]. Although the frequency of cache writes is less than that of cache reads, due to the large power consumption, the impact of cache write on the total cache power consumption cannot be ignored, especially for the data caches or the instruction caches with a high miss ratio.

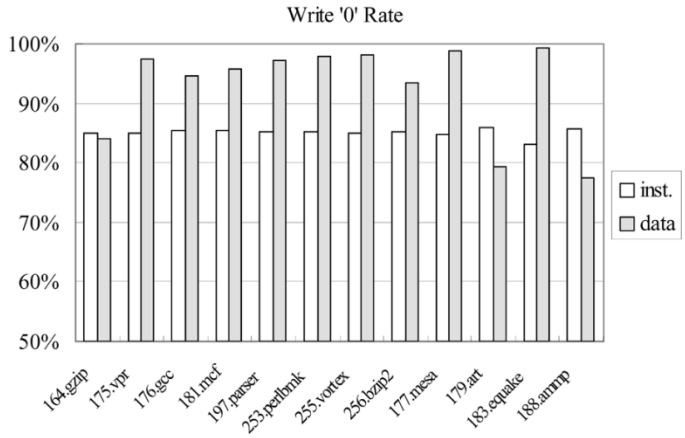


Fig. 2. Write-zero rates of instruction and data caches for *SPEC2000*.

B. 0/1 Distribution of the Write Data

Fig. 2 shows the proportion of "0" bits to the total cache write bits (referred to as *write-zero rate*) examined from the execution traces of the SPEC2000 benchmarks. From this figure, around 85% of the instruction write bits are "0," and over 90% of the data write 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 write bits is highly skewed toward zero. Unlike the conventional cache where the power dissipated in both writing "1" and "0" are the same (this is explained in detail in the following section), motivated by the extremely asymmetric distribution of "0" and "1" bits in the write data, we propose a zero-aware (ZA) SRAM cell, in which the power dissipated in writing "0" is much less than the power dissipated in writing "1." By exploiting the prevalence of "0" bits in the write data, the proposed ZA cell can effectively reduce the average cache power consumption during a write.

III. Zero-Aware Asymmetric SRAM Cell

A. Power Distribution During a Write

To read/write information from/to multiple locations without addressing or data contentions, in this paper we consider a cell model with one read port and one write port. Fig. 3 shows the column circuit for the conventional write port, which consists of two bitlines (bit and $-bit$) and S memory cells. In the dynamic logic design, the bitlines are usually initially precharged and equilibrated to supply voltage ($V_{\rm DD}$). When the write enable (*WE*) signal is asserted, the input data and its complement are placed on the bit and $-bit$ bitlines, respectively. Then, by asserting the write wordline (*WWL*) signal, the access transistors *N3* and *N4* connect the bitlines to the cell to write the data. For example, to write the cell state from "1" to "0," the bit line is pulled down to almost 0 V. Because the bit line has to be

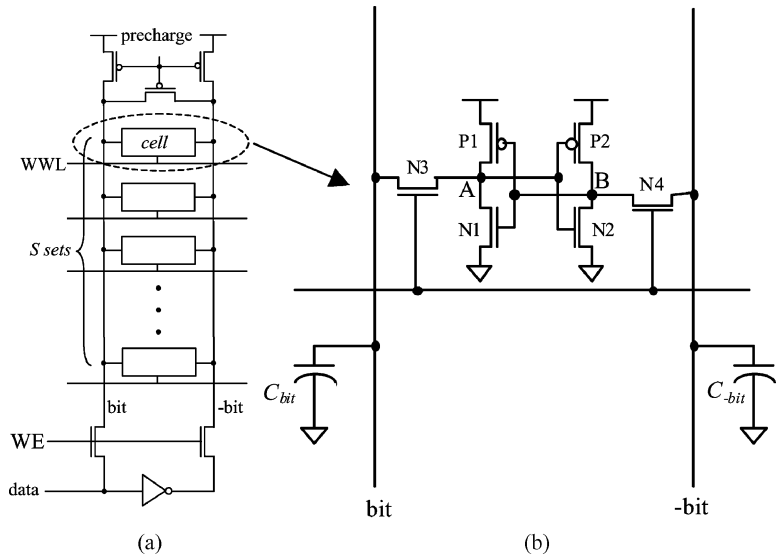


Fig. 3. Column circuit for the conventional write port.

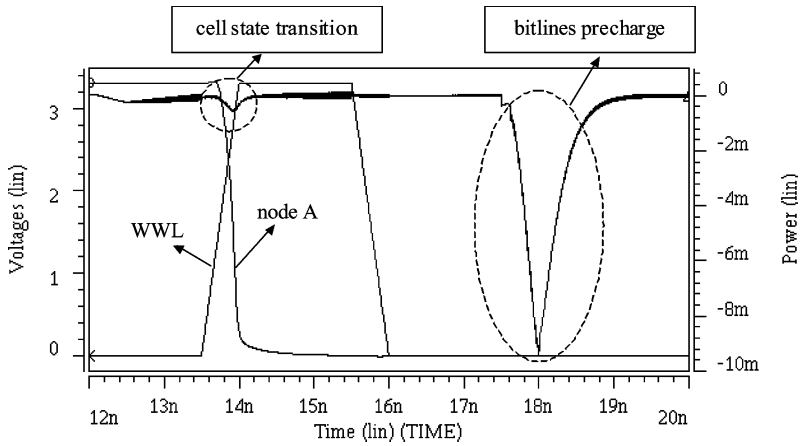


Fig. 4. Power distribution during a write for the conventional cell.

precharged to $V_{\rm DD}$ before the next write, the swing is very large thereby contributing considerable power consumption.

To further analyze the power consumption of bitlines swing, a cache column with 128 cells was implemented in TSMC 0.35-μm technology with a 3.3-V supply. As shown in Fig. 4, there are two power-consuming phases during a write operation. 1) The first power-consuming phase is the state transition phase, in which the inverter switch induces the short-circuit power dissipation. This transition power consumption only arises in the state transition cases. 2) The second power-consuming phase is the precharge phase. Before the next write, one of the two bitlines would be precharged to V_{DD}. This bitline precharge is independent of the written value. Because the transition power consumption is usually negligible, the power dissipated in writing "1" and "0" are almost the same.

B. Zero-Aware (ZA) Asymmetric SRAM Cell

Unlike the conventional cell, where the power dissipated in writing "1" and "0" are the same, we propose a zero-aware SRAM cell with an asymmetric inverter pair, called *ZA* cell, to reduce the average write power by minimizing the power dissipated in writing "0." Fig. 5 shows the schematic of the ZA cell and its relative signals, where the write select *(WS)*, write wordline *(WWL)* are used to select a cell for writing, and the data line *(WZ)* is used for signaling whether the current operation is write "0" or not.

As shown in Fig. 5(a), because the write port of the ZA asymmetric cell is a single-bitline configuration, we can use the input data (bit) or its complement ($-bit$) to perform the write operation. The only difference is that the bitline is connected to node *A* or *B*. Based on the most write data are "0" (as indicated in Section II-B), if we use bit value to perform the write operation, the frequency of bitline discharge/precharge would be very high. Thus, to reduce the number of bitline discharge/precharge, in the ZA cell we use $-bit$ value (i.e., *WZ*) to perform the write operation instead of bit value.

Compared to the conventional SRAM cell, the proposed ZA cell contains one *NMOS* transistor *N3* controlled by *WS* signal. It results in an asymmetric inverter pair: Inv-A and Inv-B. *N3* is a key tail transistor to flip Inv-B state from low to high correctly, which is traditionally difficult in the single-bitline configuration. The detailed operation of the ZA cell is described below.

1) Read Mode: In the read mode, *WWL* is held to 0 and the tail transistor *N3* is turned on to activate the Inv-B. Because we consider the cell with split one read port and one write port, the read port has read wordline (*RWL*) for cell selection, which is different from the write wordline (*WWL*) of the write port.

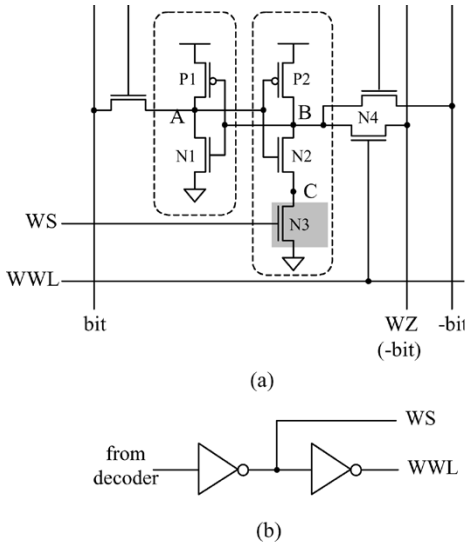


Fig. 5. (a) Zero-aware asymmetric cell. (b) The generation of write select (*WS*) and write wordline (*WWL*) signals.

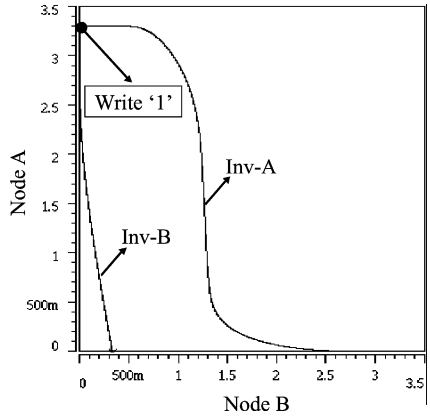


Fig. 6. Electrical characteristics of the inverter pair in the ZA cell during the write "1" mode.

Therefore, the read operation of the ZA cell is the same as that of the conventional cell.

2) Write "1" Mode: In the write "1" mode, node *B* must be written to low that is done by setting *WZ* 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 *WZ* 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*. The electrical characteristics of the inverters in the ZA cell during the write "1" mode are shown in Fig. 6.

3) Write "0" Mode: In the write "0" mode, node *B* must be written to high that is done by setting *WZ* to V_{DD} and asserting the *WWL*. The first possible write pattern is 0-$>$ 0. Because both nodes *B* and *WZ* are high, no state transition arises in this case.

Another possible write pattern is 1-$>$ 0. In this case, if the ZA 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. It was used to decouple the transfer curves of the cell inverters and eliminate one of the stable states during the boost, as shown in Fig. 7(a). The disadvantages of the boosted wordline technique are the potential instability and the hardware overheads.

Instead of the boosted wordline technique, in the ZA 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. 5(b)], the pull-down path through driver transistor *N2* is disconnected. Therefore, it is easy to flip the cell state from "1" to "0" by charging node *B* through *N4*. The electrical characteristics of the inverters in the ZA cell during the write "0" mode are shown in Fig. 7(b).

C. Related Work

Based on the same observation that the cache access stream exhibits the strong bias toward zero at the bit level, there are many techniques proposed to reduce the cache power consumption. A single-ended read bitline [9] was 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 [10], another scheme for ROMs and small RAMs with single-ended bitlines is proposed to conditionally invert stored word to reduce the total number of bitline discharges. The dynamic zero compression (DZC) scheme [5] was proposed to reduce the energy required for cache accesses by only writing and reading a single bit for every zero-value byte. The DZC method must add an additional zero indicator bit (ZIB) to each byte that indicates whether this byte contains all zero bits. On a write to the cache, only the ZIB is written if the byte is zero, otherwise, both the data bits and the ZIB are written. 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. By contrast, the proposed ZA cell can effectively reduce the cache power consumption without the necessity for the cluster of "0" bits.

Unlike the techniques described above that mainly reduce the cache dynamic power, Azizi *et al.* proposed an asymmetric SRAM cell [11], [12], in which a selected set of transistors are implemented with *high*-V_T (threshold voltage) to reduce leakage power when the cell is storing a zero (the common case). One obvious difference between our and Azizi's work is that the technique proposed by Azizi *et al.* maintains the traditional SRAM architecture to reduce the leakage power consumption in storing "0." In contrast, we modify the traditional SRAM architecture to reduce the dynamic power consumption in writing "0." In their method, the word "asymmetric" means that two different threshold voltages (*high*-V_T and *regular*-V_T) are used in the invert pair of SRAM cell, but in our method, the word "asymmetric" means that the number of transistors used in the invert pair of SRAM cell is different, i.e., Inv. A contains two transistors and Inv. B contains three transistors [as shown in Fig. 5(a)].

IV. Stability, Access Delay and Power Consumption

This section provides the detailed analysis of the proposed ZA cell from various criteria. We first estimate the impacts of the ZA cell on the stability and performance that includes the read and write delays. With the same stability and performance

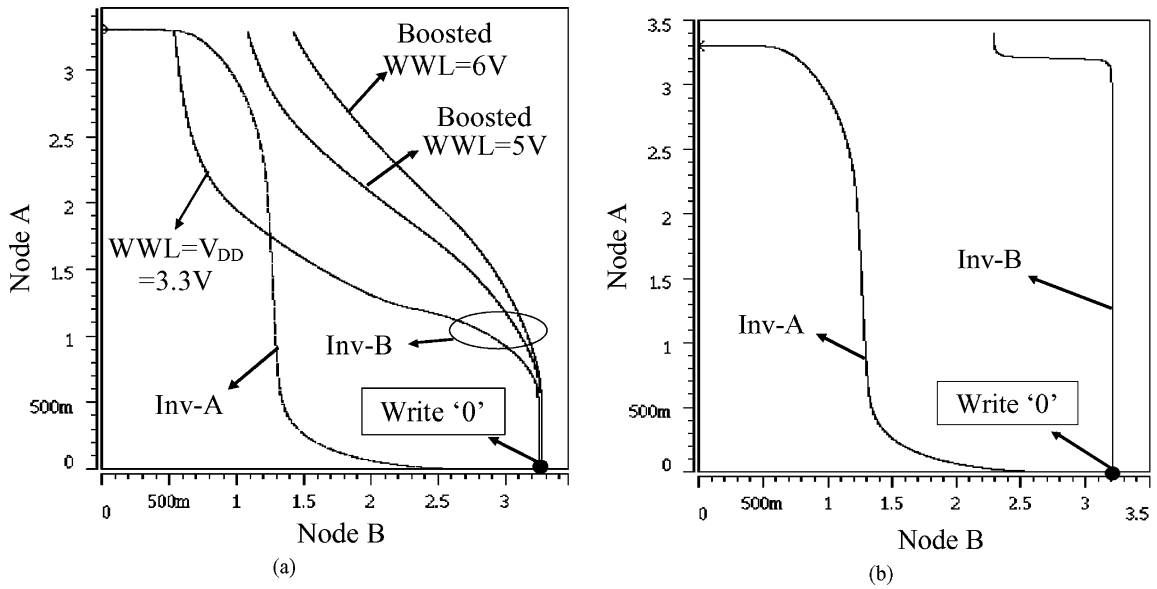


Fig. 7. Electrical characteristics of the inverter pair in the ZA cell during the write "0" mode. (a) Without the tail transistor *N3*. (b) With the tail transistor *N3*.

as the conventional SRAM cell, the cell area and write power reduction of the ZA cell are provided.

A. 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. The SNM of SRAM cell is defined as the maximum value of noise that can be tolerated by the cross-coupled inverters before altering state. In this paper, we do not consider the process variation in the analysis of cell stability, because it depends on the process and cannot be controlled by the architects. A basic understanding of the SNM is obtained by drawing and mirroring the inverter characteristics, and then finding the maximum possible square between them. According to the results shown in [13], the analytical SNM of the conventional SRAM cell ($\mathrm{SNM_{Conv}}$) is characterized with the following expressions:

$$\mathrm{SNM_{Conv}} = V_T - \frac{1}{k+1} \cdot \left(\frac{V_{\mathrm{DD}} - \frac{2r+1}{r+1} V_T}{1 + \frac{r}{k(r+1)}} - \frac{V_{\mathrm{DD}} - 2V_T}{1 + k\frac{r}{q} + \sqrt{\frac{r}{q}\left(1 + 2k + \frac{r}{q}k^2\right)}} \right) \quad (1)$$

where

r = $\beta_{\mathrm{driver}}/\beta_{\mathrm{access}}$;
q = $\beta_{\mathrm{load}}/\beta_{\mathrm{access}}$;
k = $(r/r+1)\left(\sqrt{(r+1/(r+1-V_s^2/V_r^2))} - 1\right)$;
V_s = $V_{\mathrm{DD}} - V_T$;
V_r = $V_s - r/(r+1)V_T$.

In this analytical SNM model, the threshold voltages of the NMOS and PMOS are assumed equal, and the second-order effects such as mobility reduction and velocity saturation are neglected. From (1), the SNM is only dependent on the threshold voltage V_T, V_{DD} and the β ratios r and q, not on the absolute value of β. r is referred to as *cell ratio* and defined by $\beta_{\mathrm{driver}}/\beta_{\mathrm{access}}$, in which β_{driver}, β_{load}, and β_{access} are the W/L ratios of driver transistors (*N1, N2*), load transistors (*P1, P2*) and access transistors (*N3, N4*), respectively. Clearly, to design the cells with large SNM, r must be enlarged, which of course is constrained by the requirements of small cell area and correct read/write operation.

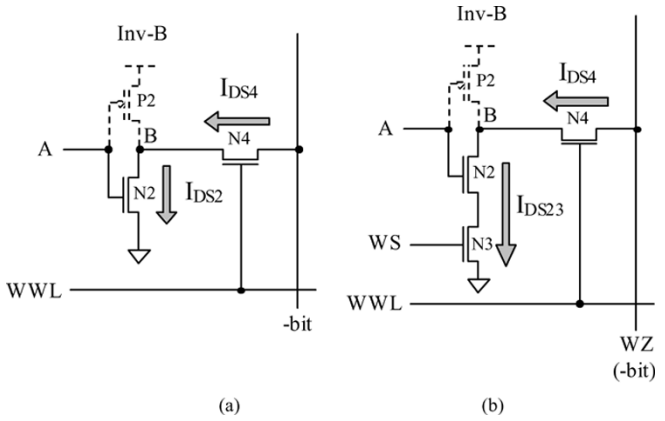


Fig. 8. The difference between the conventional and ZA cells. (a) The Inv-B of the conventional cell. (b) The Inv-B of the ZA cell. The devices shown in dotted are assumed to be nonconducting.

As shown in Fig. 8, the major difference between the conventional cell [Fig. 3(b)] and the ZA cell is the *Inv-B*. In the ZA cell [Fig. 8(b)], because the tail transistor *N3* is on the critical path in driving node *B* to low, it results in an asymmetrical inverter pair that potentially degrades the stability. Based on the analytic model used in [13], *N4* and *N2* operate in the saturation and linear regions, respectively. For the conventional cell, we equate the drain currents of *N2* and *N4* as follows:

$$\begin{aligned} I_{\mathrm{DS4}} &= I_{\mathrm{DS2}} \\ \frac{1}{2}\beta_{N4}(V_{\mathrm{GS4}} - V_T)^2 &= \beta_{N2} V_{\mathrm{DS2}} \left(V_{\mathrm{GS2}} - V_T - \frac{1}{2}V_{\mathrm{DS2}} \right) \\ (V_{\mathrm{GS4}} - V_T)^2 &= 2\frac{\beta_{N2}}{\beta_{N4}} V_{\mathrm{DS2}} \left(V_{\mathrm{GS2}} - V_T - \frac{1}{2}V_{\mathrm{DS2}} \right) \\ (V_{\mathrm{GS4}} - V_T)^2 &= 2rV_{\mathrm{DS2}} \left(V_{\mathrm{GS2}} - V_T - \frac{1}{2}V_{\mathrm{DS2}} \right). \end{aligned} \quad (2)$$

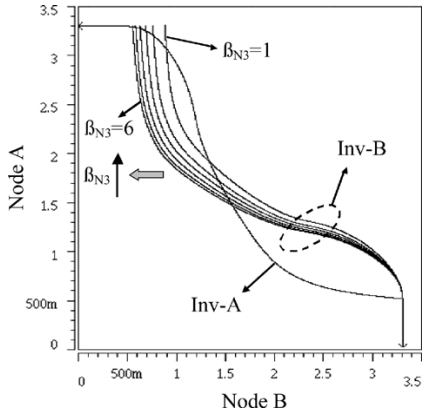


Fig. 9. Graphical representation of the $\mathrm{SNM_{ZA}}$. It increases with the β_{N3} if the β_{N2} and β_{N4} are fixed ($\beta_{N2} = 2$ and $\beta_{N4} = 1$ in this case).

For the ZA cell, because *N2* and *N3* are connected serially, the total resistance $R_{23} = R_2 + R_3$, where

$$
\begin{aligned}
R_2 &= \frac{1}{\beta_{N2}\left(V_{\mathrm{GS2}} - V_T - \frac{V_{\mathrm{DS2}}}{2}\right)} \\
R_3 &= \frac{1}{\beta_{N3}\left(V_{\mathrm{GS3}} - V_T - \frac{V_{\mathrm{DS3}}}{2}\right)} \\
\Rightarrow R_{23} &= R_2 + R_3 \\
&= \frac{\beta_{N2} + \beta_{N3}}{\beta_{N2}\beta_{N3}\left(V_{\mathrm{GS23}} - V_T - \frac{V_{\mathrm{DS23}}}{2}\right)} \\
I_{\mathrm{DS4}} &= I_{\mathrm{DS23}} = \frac{V_{\mathrm{DS23}}}{R_{23}} \\
\Rightarrow \frac{1}{2}\beta_{N4}(V_{\mathrm{GS4}} - V_T)^2 &= \frac{\beta_{N2}\beta_{N3}}{\beta_{N2} + \beta_{N3}} V_{\mathrm{DS23}} \\
&\quad \times \left(V_{\mathrm{GS23}} - V_T - \frac{1}{2}V_{\mathrm{DS23}}\right) \\
(V_{\mathrm{GS4}} - V_T)^2 &= 2\frac{\beta_{N2}\beta_{N3}}{\beta_{N4}(\beta_{N2} + \beta_{N3})} V_{\mathrm{DS23}} \\
&\quad \times \left(V_{\mathrm{GS23}} - V_T - \frac{1}{2}V_{\mathrm{DS23}}\right) \\
(V_{GS4} - V_T)^2 &= 2r' V_{\mathrm{DS23}} \\
&\quad \times \left(V_{\mathrm{GS23}} - V_T - \frac{1}{2}V_{\mathrm{DS23}}\right). \quad (3)
\end{aligned}
$$

In (3), for simplicity we assume V_{GS2} and V_{GS3} are equal to V_{GS23} (the equivalence gate-to-source voltage of *N2* and *N3*), as well as V_{DS2} and V_{DS3} are equal to V_{DS23} (the equivalence drain-to-source voltage of *N2* and *N3*). Compared to (2), the cell ratio of the ZA cell is $r' = \beta_{N2}\beta_{N3}/(\beta_{N4}(\beta_{N2} + \beta_{N3}))$ that means besides the β_{N2}, the SNM of the ZA cell ($\mathrm{SNM_{ZA}}$) is also dependent on the β_{N3} (the W/L ratio of tail transistor *N3*). This analytic estimation model would be verified by the later experimental results. Fig. 9 shows how the $\mathrm{SNM_{ZA}}$ varies with the β_{N3}. The asymmetry in the ZA butterfly curve and the decrease of SNM are the results of two internal asymmetric inverters. The $\mathrm{SNM_{ZA}}$ would increase with the β_{N3} if the β_{N2} and β_{N4} are fixed.

In the ZA cell, we have to enlarge the β_{N2} and β_{N3} to compensate the stability loss due to the asymmetrical inverter pair. Based on (3), to maintain the $\mathrm{SNM_{ZA}}$ the same as the

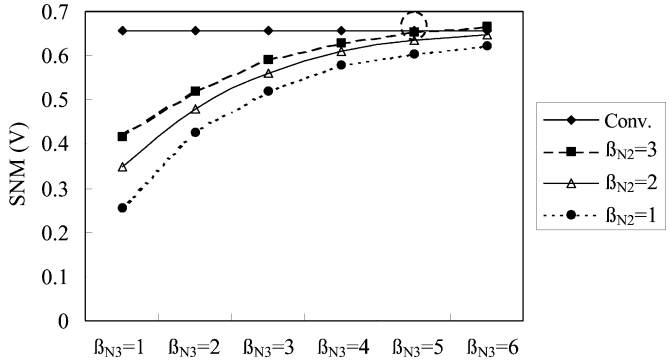


Fig. 10. The $\mathrm{SNM_{ZA}}$ in different combination of the β_{N2} and β_{N3}. ($\beta_{N4} = 1$).

$\mathrm{SNM_{Conv}}$, the ZA cell ratio (r') must equal the conventional cell ratio (r), which can be achieved by appropriate choice of β_{N2} and β_{N3}. In the following comparison, we assume the β_{N2} and β_{N4} of the conventional cell are 2 and 1, respectively (i.e., the conventional cell ratio r is fixed to 2). Obtained from the HSPICE simulation, Fig. 10 shows the $\mathrm{SNM_{ZA}}$ in different combinations of β_{N2} and β_{N3} when $\beta_{N4} = 1$. The key observation is that when β_{N2} is 3 and β_{N3} is 5, the $\mathrm{SNM_{Conv}}$ and $\mathrm{SNM_{ZA}}$ are almost the same value 654 mV. The ZA cell ratio $r' = 15/8$ approximates the conventional cell ratio $r = 2$ that verifies the estimation model developed in (3), although the equation is slightly inaccurate due to simplicity.

B. Access Delay

1) Read Delay: Except for the cell core, because the peripheral circuits in both the conventional and ZA cells are the same, we define the *read delay* as the elapsed time from asserting *RWL* to the sufficient bitline swing for correct data sensing. According to the results given in [6], the reasonable bitline swing is 10% of the full supply voltage. Depending on the value stored in the cell, there are two cases in read delay: one is read "0" delay and the other is read "1" delay.

a) In the case of read "0," the bit line would be discharged to low through the driver transistor *N1* of Inv-A. This path is identical to the conventional cell in reading "0." Thus, the read "0" delays are of the same 1.2385 ns for both the conventional and ZA cells.

b) In the case of read "1," the $\overline{\text{bit}}$ line would be discharged to low through the driver transistor *N2* and the tail transistor *N3* of Inv-B. Fig. 11(a) shows the critical path of reading "1." Because *N3* is always turned on in the read mode, similar to SNM, the read "1" delay also depends on both β_{N2} and β_{N3}. For a better SNM, the β_{N2} is fixed to be 3 and Fig. 11(b) shows how the read "1" delay varies with β_{N3}. It is clear from this figure that when β_{N3} is 5, the read "1" delays of both the conventional and ZA cells are almost the same 1.23 ns.

2) Write Delay: The *write delay* is defined as: the elapsed time from which asserting *WWL* to the states of both nodes *A* and *B* become steady. Because in the write mode the tail transistor *N3* is turned off that disconnects the path driving node *B* to low, the write delay is independent of the *N2* and *N3*. There are four cases in write operation: writing the cell state from "0" to "0" (0- $>$ 0); "0" to

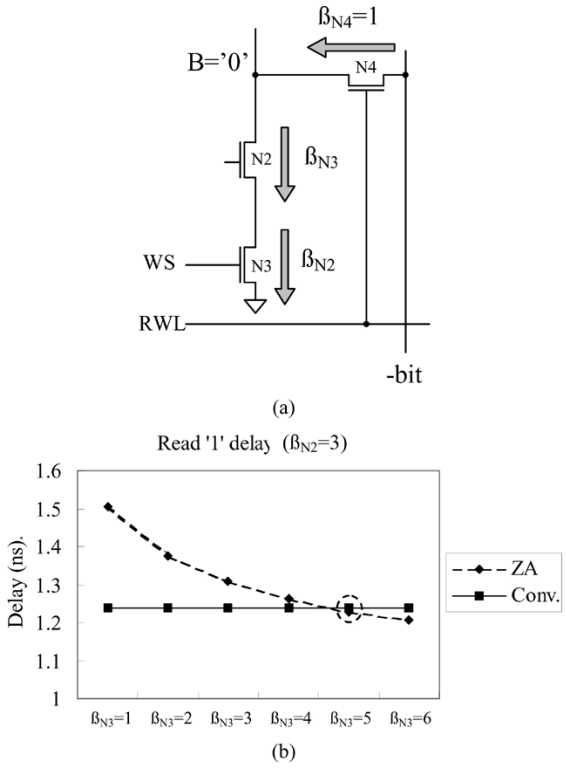


Fig. 11. (a) The critical path of reading "1." (b) The read "1" delay varies with the β_{N3} if the $\beta_N = 3$ and $\beta_{N4} = 1$.

"1" (0-$>$ 1); "1" to "0" (1-$>$ 0); and "1" to "1" (1-$>$ 1). Due to no state transition occurred in cases of 0-$>$ 0 and 1-$>$ 1, we only consider the write delay in cases of 0-$>$ 1 and 1-$>$ 0.

a) In the case of 0-$>$ 1, by setting *WZ* to *0* and then asserting *WWL*, node *B* with initial high state would be discharged to low, as shown in Fig. 12(a). Compared to the traditional write port with differential bitlines, because in the ZA cell the state transition is driven by only one path, the 0- $>$ 1 write delay of ZA cell is slightly larger than that of the conventional cell, as shown in Table I. Nevertheless, in determining write cycle, this minor difference can be ignored.

b) In the case of 1-$>$ 0, by setting *WZ* 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*. Fig. 12(b) shows the path in writing cell state from "1" to "0." From Table I, because *N3* is turned off in the write mode, the 1-$>$ 0 write delay of the ZA cell is even smaller than that of the conventional cell.

C. Write Power Reduction

Based on the analysis described above, in contrast with the conventional cell with $\beta_{N2} = 2$, we conclude that the ZA cell does not compromise either stability or access delay if β_{N2} and β_{N3} are enlarged to 3 and 5, respectively. Fig. 13 shows the power distribution of the ZA cell in the 1- $>$ 0 write pattern. Compared to the conventional cell (Fig. 4), besides the state transition power, almost no precharge power arises in this write pattern. In the 0- $>$ 0 write pattern, even the state transition power does not arise. Consequently, in writing "0" (1- $>$ 0 or 0-$>$ 0), the ZA cell consumes far less power than the conventional cell.

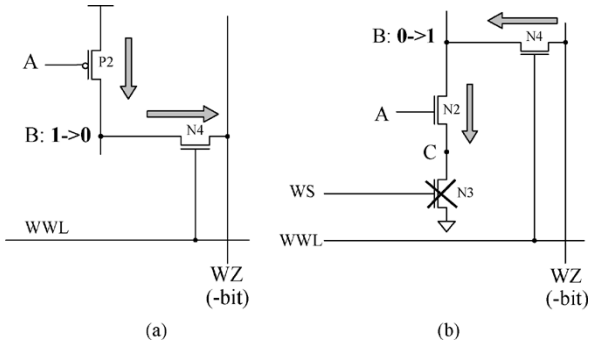


Fig. 12. (a) The path in writing cell state from "0" to "1." (b) The path in writing cell state from "1" to "0."

TABLE I
WRITE DELAY SUMMARY

Write Delay (ns)	0->1 Write Delay	1->0 Write Delay
Conv.	0.7512	0.7513
ZA	0.7558	0.7502

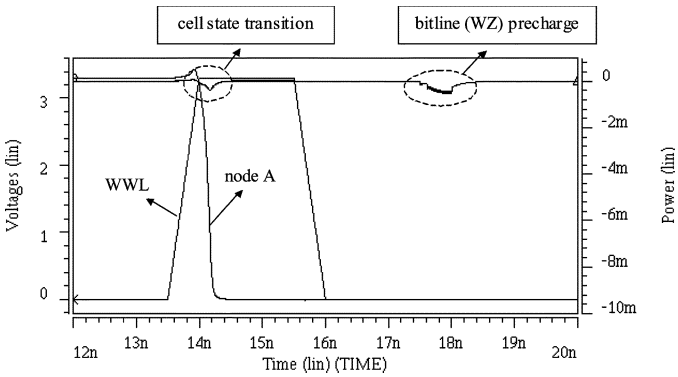
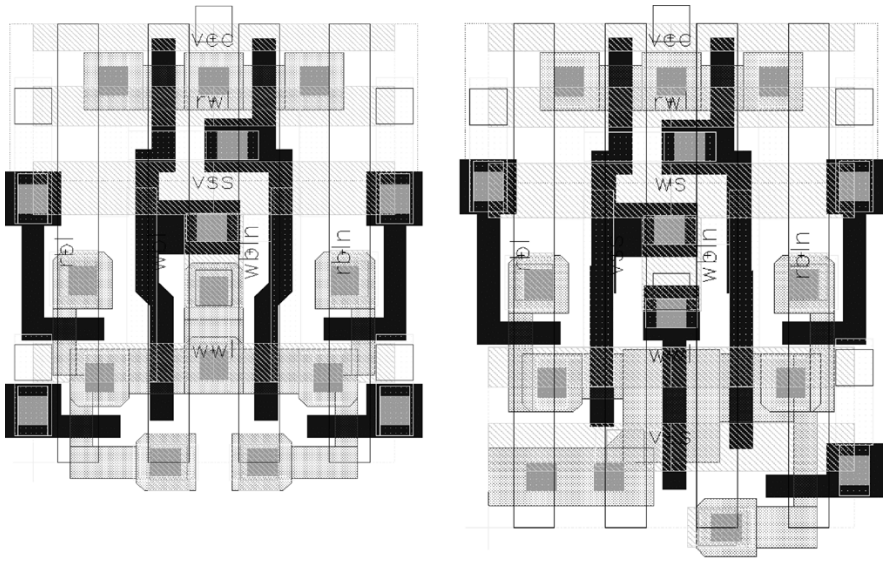


Fig. 13. Power distribution of ZA cell in the 1- $>$ 0 write pattern.

In the ZA cell, the *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 II shows the column power consumption for various write patterns, where one column consists of 128 cells. In the conventional cell, regardless of write pattern, the column power consumptions are almost the same. Compared to the conventional cell, in the 1-$>$ 0 write pattern, the ZA cell reduces the column power consumption by 97.07%. Due to no state transition and bitline discharge, even by 98.75% the column power reduction can be achieved in the 0- $>$ 0 write pattern.

D. Cell Area

Both the conventional and ZA cells have eight transistors for one read port and one write port. The ZA cell saves one bitline in the write port, but the *WS* signal line has to be paid to guarantee the correct write operation. Thus, the costs of wire interconnection in both cells are almost the same. As described previously, to compensate the stability and performance losses due to the asymmetrical inverter pair in the ZA cell, we have to enlarge β_{N2} and β_{N3}. Fig. 14 shows the physical layouts of the conventional and ZA cells. The conventional cell size is 5.49 $\times$ 6.89 and the proposed ZA cell size is 5.49 $\times$ 7.76. Note that the width of our ZA cell is purposely retained the same as the width of the



(a) The conventional cell layout. (b) The ZA cell layout.

Fig. 14. Physical layouts of the conventional and ZA cells.

TABLE II
SUMMARY OF WRITE POWER DISSIPATED IN ONE COLUMN WITH 128 CELLS

Column Power (mW)	Conv.	ZA	***Reduction***
1->0	4.32E-01	1.27E-02	*97.07%*
0->0	4.06E-01	5.09E-03	*98.75%*
1->1	4.05E-01	3.92E-01	*3.13%*
0->1	4.57E-01	4.32E-01	*5.51%*

conventional cell, such that both cells have the same power dissipated in the wordlines. Compared to the conventional cell, the ZA cell area is increased from 37.80 μm^2 to *42.56* μm^2. Most area overhead is introduced by the large driver transistor *N2* and tail transistor *N3* of Inv-B that imposes around a 12.6% cell area overhead. By using *CACTI 3.0* tool [14], we obtain that the percentage of cell-array of the total cache area is about 70% for a 32 kB 2-way or 4-way cache. Thus, the overall cache area overhead is roughly $12.6\% * 70\% = 8.8\%$.

V. EXPERIMENTAL RESULTS

For the results presented in this study, we use the on-chip level-one (L1) cache architecture with split instruction and data caches, which are a 32 KB, 2-way instruction cache (IC) and a 32 KB 4-way DC, respectively. To avoid an explosion in the number of simulations, the block size for both caches is fixed to be 32 bytes.

A. Benchmarks

We use *SimpleScalar* [15] to evaluate the 0/1 distribution of the write data for both the SPEC2000 and MediaBench benchmark suits. The SPEC2000 [16] is a suit of general-purpose programs, which is specifically developed to assist in commercial evaluation and marketing of desktop computing systems. The MediaBench [17] is a suite of applications focus on multimedia and communications systems. To get a good mix of CPU-intensive and memory-intensive loads, we employ four CINT2000 benchmarks (164.gzip, 176.gcc, 253.perlbmk, 255.vortex), four CFP2000 benchmarks (177.mesa, 179.art, 183.equake, 188.ammp) and three integer MediaBench benchmarks (adpcm, jpeg, and gsm). Each benchmark from the MediaBench suit has two separate programs: encoding and decoding. Table III summarizes the benchmarks, provides a brief description of them, and indicates the number of instructions and data simulated for each workload.

B. Results and Discussions

A cache consists of tag array and data array, which are used to store the tag and the actual data, respectively. The tag is the high-order bits of the address for determining whether the current access is a hit or miss. Because the program size is usually an insignificant fraction of the entire address space, most tag bits are "0." As expected, with the advent of 64-bit address space, the prevalence of "0" bits would be further enlarged.

1) Write Pattern Distribution: Table IV shows the write pattern distribution of both tag and data arrays for IC and DC. For SPEC2000, because the difference between integer and floating-point programs is hardly noticeable, we do not present these two benchmarks separately. From this table, we summarize the most important aspects. For SPEC2000 as shown in Table IV(a), the percentage of the $0\text{-}>0$ write pattern, referred to as $0\text{-}>0$ *rate*, is over 70% for all cases except for the DC tag. In contrast, Table IV(b) shows the $0\text{-}>0$ rate of MediaBench is larger than that of SPEC2000 for all cases, especially for the DC tag and data. In these two cases, the $0\text{-}>0$ rate of MediaBench is larger than that of SPEC2000 by 21%. This means the write characteristics of the multimedia programs possess high $0\text{-}>0$ rate. It is particularly beneficial to the proposed ZA cell, which hardly consumes power in the $0\text{-}>0$ write pattern.

2) Average Column Write Power (ACWP): We define the ACWP as the power dissipated in one column during each write. Because there are four write patterns, by definition, the ACWP is given by

$$\begin{aligned}\text{ACWP} = &(\text{CP}_{0\text{->}0} \times R_{0\text{->}0}) + (\text{CP}_{1\text{->}0} \times R_{1\text{->}0}) \\ &+ (\text{CP}_{0\text{->}1} \times R_{0\text{->}1}) + (\text{CP}_{1\text{->}1} \times R_{1\text{->}1}) \quad (4)\end{aligned}$$

TABLE III
BENCHMARKS SUMMARY

Category	Benchmark	Description	Instr. Count	Data Count
CINT2000	164.gzip	Compression	81641529094	26630126869
	176.gcc	Programming Language Compiler	78423865621	31384113066
	253.perlbmk	PERL Programming Language	43730351658	20095105101
	255.vortex	Object-oriented Database	168619585094	88466806040
CFP2000	177.mesa	3-D Graphics Library	492137176762	246401727733
	179.art	Image Recognition/Neural Networks	181402289419	78280973858
	183.equake	Seismic Wave Propagation Simulation	597511951360	235108186746
	188.ammp	Computational Chemistry	1924889017223	542422636930
MediaBench	adpcm	A speech compression and decompression program	enc: 599217699	enc: 79984352
			dec: 498397938	dec: 79957740
	gsm	European GSM 06.10 provisional standard for fullrate speech	enc: 1863215361	enc: 416445281
			dec: 619578116	dec: 79055392
	jpeg	A standardized image compression and decompression program	enc: 101087026	enc: 28938666
			dec: 25544574	dec: 8743829

TABLE IV
WRITE PATTERN DISTRIBUTION OF BOTH TAG AND DATA ARRAYS FOR IC AND DC

(a) SPEC2000 benchmarks.

SPEC2000		0->0	1->0	0->1	1->1
IC	tag	75.57%	6.19%	8.01%	10.22%
(32K 2-way)	data	77.36%	7.72%	9.81%	5.10%
DC	tag	58.87%	9.53%	9.56%	22.04%
(32K 4-way)	data	71.52%	21.26%	3.45%	3.77%

(b) MediaBench benchmarks.

MediaBench		0->0	1->0	0->1	1->1
IC	tag	85.77%	3.07%	5.41%	5.74%
(32K 2-way)	data	79.23%	6.35%	10.88%	3.55%
DC	tag	79.72%	2.33%	14.75%	3.19%
(32K 4-way)	data	92.40%	2.41%	4.25%	0.94%

TABLE V
THE POWER DISSIPATED IN ONE COLUMN FOR VARIOUS WRITE PATTERNS

Column Power (mW)		$CP_{0->0}$	$CP_{1->0}$	$CP_{0->1}$	$CP_{1->1}$
Conv.	IC	4.06E-01	4.32E-01	4.57E-01	4.05E-01
	DC	2.44E-01	2.46E-01	2.38E-01	2.43E-01
ZA	IC	5.09E-03	1.27E-02	4.32E-01	3.92E-01
	DC	2.45E-03	6.15E-03	2.20E-01	1.99E-01

TABLE VI
THE IMPACT OF ZA CELL ON THE ACWP FOR BOTH IC AND DC

(a) SPEC2000 benchmarks.

ACWP (mW)		Conv.	ZA	***Reduction***
IC	tag	4.12E-01	7.93E-02	*80.74%*
(32K 2-way)	data	4.13E-01	6.73E-02	*83.71%*
DC	tag	2.43E-01	6.70E-02	*72.44%*
(32K 4-way)	data	2.44E-01	1.82E-02	*92.55%*

(b) MediaBench benchmarks.

ACWP (mW)		Conv.	ZA	***Reduction***
IC	tag	4.10E-01	5.07E-02	*87.64%*
(32K 2-way)	data	4.13E-01	6.57E-02	*84.10%*
DC	tag	2.43E-01	4.10E-02	*83.14%*
(32K 4-way)	data	2.44E-01	1.37E-02	*94.39%*

TABLE VII
SUMMARY OF CACHE WRITE POWER REDUCTION FOR THE USE OF ZA CELL

	SPEC2000	*MediaBench*
IC (32K 2-way)	60.18%	60.66%
DC (32K 4-way)	66.03%	67.63%

$CP_{0->0}$ is the power dissipated in one column for 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 power dissipated in one column for various write patterns are listed in Table V. We assume the tag array is implemented with the same SRAM cell as the data array. Applying the data shown in Tables IV and V to (4), the ACWP for each configuration are obtained and listed in Table VI. Because the column write power of the conventional cell is independent of write pattern, the results show that the ACWP of the conventional cell is almost equal to any column write power. Compared to the conventional cell, by minimizing the power dissipated in writing "0" (including 0- $>$ 0 and 1- $>$ 0), the ZA cell can reduce the ACWP by $72\% \sim 92\%$ for SPEC2000 benchmarks, and by $83\% \sim 94\%$ for MediaBench benchmarks. In particular, for DC data, the ZA cell even reduces the ACWP by 92.71% and 94.51% for SPEC2000 and MediaBench, respectively.

Note that the power dissipated in bitlines is only a part of the total cache power consumption. According to the results shown in [5], in which the authors presented a breakdown of the cache power consumption for writes. The tag bitlines and data bitlines contribute about 3% and 69% to the total cache power consumption during a write, respectively. To obtain the total cache power reduction during a write, in our method the cache write power reduction in tag bitlines and data bitlines must be multiplied by 3% and 69%, and then add these two parts. As summarized in Table VII, in the baseline IC the proposed ZA cell can result in roughly 60% reduction in total cache power consumption during a write for both SPEC2000 and MediaBench. In the baseline DC, the ZA cell reduces the total cache power consumption during a write by 66% and 68% for SPEC2000 and MediaBench, respectively.

VI. CONCLUSION

Most low-power SRAM techniques only reduce read power, but generally write power is larger than read power. In this paper, we concentrate on reducing the cache write power. Based on over 85% and 90% of the values written to the instruction cache

and data cache are "0," we propose a novel *zero-aware asymmetric (ZA) cell* to minimize the cache power consumption in writing "0," which is orthogonal and synergistic to other low-power techniques. While exploiting the prevalence of "0" to reduce the average write power, the ZA cell can retain the same stability and performance as the conventional cell with a cache area increase of 8.8%. The experimental results show that the proposed ZA cell can reduce the average cache write power consumption up to 61% and 68% for the baseline instruction and data caches, respectively. The register files, on-chip level-two (L2) caches, TLB and BTB are usually implemented with the SRAM cell. Because all of them are used for caching data temporarily, they also exhibit the 0/1 distribution is highly skewed toward zero. Consequently, we can directly apply the proposed ZA cell to these array structures to reduce the average write power but with a 12.6% cell area penalty.

REFERENCES

[1] J. F. Edmondson *et al.*, "Internal organization of the Alpha 21164, a 300-MHz 64-bit quad-issue CMOS RISC microprocessor," *Digital Tech. J.*, vol. 7, no. 1, pp. 119–135, 1995.

[2] J. Montanaro *et al.*, "A 160 MHz, 32 b 0.5 W CMOS RISC microprocessor," in *IEEE ISSCC 1996 Dig. Papers*, 1996.

[3] M. Ukita *et al.*, "A single-bit-line cross-point cell activation (SCPA) architecture for ultra-low-power SRAM's," *IEEE J. Solid-State Circuits*, vol. 28, pp. 1114–1118, Nov. 1993.

[4] G. Reinman and N. P. Jouppi, "CACTI 2.0: An Integrated Cache Timing and Power Model, COMPAQ WRL Res. Report,", 2000.

[5] L. Villa, M. Zhang, and K. Asanovic, "Dynamic zero compression for cache energy reduction," in *Proc. 33rd Int. Symp. Microarchitecture Micro-33*, 2000, pp. 214–220.

[6] B. Amrutur and M. Horowitz, "Techniques to reduce power in fast wide memories," in *Proc. Symp. Low-Power Electronics*, Oct. 1994, pp. 92–93.

[7] K. W. Mai *et al.*, "Low-power SRAM design using half-swing pulse-mode techniques," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1659–1671, Nov. 1998.

[8] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 1995.

[9] J. Tseng and K. Asanovic, "Energy-efficient register access," in *Proc. Symp. Integrated Circuits and Systems Design*, Manaus, Brazil, Sept. 2000, pp. 377–382.

[10] B. K. Park, Y. S. Chang, and C. M. Kyung, "Confirming inverted data store for low power memory," in *Proc. Int. Symp. Low-Power Electronics and Design (ISLPED)*, Aug. 1999, pp. 91–93.

[11] N. Azizi, A. Moshovos, F. N. Najm, and B. Falsafi, "Asymmetric-Cell Caches: Exploiting Bit Value Biases to Reduce Leakage Power in Deep-Submicron, High-Performance Caches," Univ. of Toronto, Toronto, ON, Canada, Tech. Rep. TR-01–01-02.

[12] N. Azizi, A. Moshovos, and F. N. Najm, "Low-leakage asymmetri-cell SRAM," in *Proc. Int. Symp. Low-Power Electronics and Design (ISLPED)*, Aug. 2002, pp. 48–51.

[13] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 748–754, Oct. 1987.

[14] P. Shivakumar and N. P. Jouppi, "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," COMPAQ WRL Res. Report, 2001/2.

[15] D. C. Burger and T. M. Austin, "The simplescalar tool set, version 2.0," *Comput. Arch. News*, vol. 25, no. 3, pp. 13–25, June 1997.

[16] SPEC Benchmark Suite [Online]. Available: http://www.spec.org

[17] C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "MediaBench: A tool for evaluating and synthesizing multimedia and communications systems," in *Proc. 13th Int. Symp. Microarchitecture*, Dec. 1997, pp. 330–335.

Yen-Jen Chang (M'02) received the M.S. degree in computer science and information engineering from Chung-Yuan Christian University, Taiwan, R.O.C., in 1997, and the Ph.D. degree in computer science and information engineering from the National Taiwan University, Taipei,Taiwan, R.O.C. in 2003.

In 2004, he joined the Department of Computer Science at National Chung-Hsing University , where he is currently an Assistant Professor. His research interests include computer architecture, low-power VLSI design, and embedded system SoC design.

Feipei Lai (M'87–SM'94) received the B.S.E.E. degree from National Taiwan University, Taiwan, Taiwan, R.O.C., in 1980, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1984 and 1987, respectively.

Currently, he is a professor in the Department of Computer Science and Information Engineering and the Department of Electrical Engineering at National Taiwan University. He is also the Director of the Computer and Information Network Center at National Taiwan University. He was a Visiting Professor in the Department of Computer Science and Engineering at the University of Minnesota, MN. He was also a Guest Professor at University of Dortmund, Germany, and a Visiting Senior Computer Systems Engineer in the Center for Supercomputing Research and Development at the University of Illinois at Urbana-Champaign. Currently, he holds four Taiwan patents and two U.S. patents. He served as a consultant at ERSO, ITRI during 1988 and at Faraday Technology Corp. from August, 1994 to July 1995. He is one of the founders of the Institute of Information and Computing Machinery, Taipei, Taiwan. His current research interests are SOC low-power computing, computer architecture systems, and VLSI SOC design.

Prof. Lai is a Member of Phi Kappa Phi, Phi Tau Phi, ACM, and the Chinese Institute of Engineers. He received the Acer awards five times in 1989, 1991, 1992, 1993, and 1995 and the Taiwan Fuji Xerox Research Award in 1991. He is also a member in the "Who's Who in Science and Engineering" and "Who's Who in the World."

Chia-Lin Yang (M'00) received the B.S. degree from the National Taiwan Normal University, Taipei, Taiwan, R.O.C., in 1989, the M.S. degree from the University of Texas at Austin in 1992, and the Ph.D degree from the Department of Computer Science at Duke University, Durham, NC, 2001.

In 1993, she joined VLSI Technology Inc., San Jose, CA, (now Philips Semiconductors) as a Software Engineer. She is currently an Assistant Professor in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. Her research interests include energy-efficient microarchitectures, memory hierarchy design, and multimedia workload characterization.

Dr. Yang was the recipient of a 2000–2001 Intel Foundation Graduate Fellowship Award and she is a Member of the ACM and IEEE.