Minimizing Power Dissipation during Write Operation to Register Files

Kimish Patel
Department of Electrical Engineering
University of Southern California
Los Angeles CA 90089
(213) 821-4206
kimishpa@usc.edu

Wonbok Lee
Department of Electrical Engineering
University of Southern California
Los Angeles CA 90089
(213) 821-4206
wonbokle@usc.edu

Massoud Pedram
Department of Electrical Engineering
University of Southern California
Los Angeles CA 90089
(213) 740-4458
pedram@usc.edu

ABSTRACT

This paper presents a power reduction mechanism for the write operation in register files (RegFiles), which adds a conditional charge-sharing structure to the pair of complementary bit-lines in each column of the RegFile. Because the read and write ports for the RegFile are separately implemented, it is possible to avoid pre-charging the bit-line pair for consecutive writes. More precisely, when writing same values to some cells in the same column of the RegFile, it is possible to eliminate energy consumption due to precharging of the bit-line pair. At the same time, when writing opposite values to some cells in the same column of the RegFile, it is possible to reduce energy consumed in charging the bit-line pair thanks to charge-sharing. Motivated by these observations, we modify the bit-line structure of the write ports in the RegFile such that i) we remove per-cycle bit-line pre-charging and ii) we employ conditional data dependent charge-sharing. Experimental results on a set of SPEC2000INT / MediaBench benchmarks show an average of 61.5% energy savings with 5.1% area overhead and 16.2% increase in write access delay.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Design Aids

General Terms

Management, Design,

Keywords

Power, Register File, Write operation.

1. INTRODUCTION

In the state-of-the-art microprocessor design with very wide issue widths, power dissipation in the register file (RegFile) is becoming increasingly important. The operation and the structure of a RegFile are very similar to their memory counterparts (e.g. SRAM), except that the RegFile has *dedicated* and *separate* read and write ports i.e., a multiple number of bit-line pairs are connected to each of the 6-transistor (6-T) memory cells. Despite of these dissimilarities, the dynamic power consumption in the RegFile and memory can be similarly decomposed into several components: bit-line charging / discharging, word-line decoding, and differential sense amplification (SA).

It is known that, in the conventional 6-T SRAM structure the write operation dissipates more dynamic power than the read operation [1][2]. More precisely, during the write operation, power is dissipated by fully discharging either the bit-line or the bit-line-bar. In contrast, during the read operation either the bit-line or the bit-line-bar is partially discharged for the sense amplifier to read the value stored in the cell.

For example, consider that in some cycle we try to write a value of ‘1’ to a certain cell in some column. As a result, we have to drive the pair of bit-lines to (V_{dd}, 0). Suppose now that, in the next cycle, a cell in the same column (possibly a different one) must be written by value ‘1’, we must again set the pair of bit-lines to (V_{dd}, 0) because between these two write cycles the pair of bit-lines is typically pre-charged to a ‘1’ value. It is obvious that the bit-line pre-charging and the subsequent discharging become redundant. Next suppose that, in the next cycle, the value that must be written to some cell in that same column is ‘0’. As a result, the bit-line pair must be set to (0, V_{dd}) for the correct write operation. More precisely, ignoring the pre-charging step, the bit-line pair is switched from (V_{dd}, 0) during write ‘1’ to (0, V_{dd}) during write ‘0’. This provides us with an opportunity to save power/energy by employing the charge-sharing concept [3].

In this paper, we present a mechanism which avoids the aforementioned redundant bit-line pre-charging phase in the write operation cycles. In addition, a charge-sharing scheme is conditionally used to further increase the power/energy saving. The remainder of the paper is organized as follows. In section 2, we review some prior work on power reduction techniques in the SRAM structure and in-depth explanation of our conditional charge-sharing mechanism will be followed in section 3. The methodology and the experimental results will be in section 4. Finally, section 5 is the conclusion.

2. BACKGROUND AND PRIOR WORK

With an advent of increasingly fast, dense, and power hungry memory structures, energy has become major issues. In the half-swing (HS) scheme [4], 75% of the power reduction was achieved by restricting the bit-line swing to half of the V_{dd} combined with charge recycling. Similar swing voltage reduction technique was proposed by Yang et al. in [5]. Generally, reducing the swing voltage from the full V_{dd} swing is a powerful, cell data independent method, and its power saving is proportional to the swing voltage. However, the lower bound in swing voltage is limited since it is strongly dependent of the sensitivity of SA and the HS scheme innately has a problem with read operation since half V_{dd} of a bit-line pair in the read operation increases the erroneous cell data flip.

ISLPED’07, August 27-29, 2007, Portland, Oregon.
Copyright 2007 ACM 978-1-59593-709-4/07/0008...$5.00.

In [6], Kanda et al. reported that 90% of write power-saving was achieved in SRAM using sense-amplifying memory cell. Similar as above, their technique reduces the bit-line swing by $V_{dd}/6$ but amplifies the voltage swing by SA instead. However, their performance is bit-width dependent: They have reported to achieve 90% of write power savings under 256 bit-widths however their reduction ratio decreases when bit-width becomes smaller.

In [7], Cheng et al. devised a single bit-line driving technique for the write operation in SRAM to eliminate the excessive full charging on the bit-line pair. They force a strong ‘0’ signal in a single side of bit-lines while leaving the other side of bit-lines to be float.

In [8], Karandikar et al. introduced a hierarchical divided bit-line [9] concept in low power SRAM design. The division of bit-line into hierarchical sub bit-lines results in the reduction of bit-line capacitance, hence it reduces the dynamic power in SRAM. In this technique, however, the access of the memory is confined to a smaller sub-array and the area overhead for the extra decoding and control circuitries are not negligible.

On top of the above voltage swing based and the structural modification based approaches, some other approaches have taken advantage of the anticipated yet simultaneous charging / discharging situation and employ the charge-sharing concept to save power consumption. Basically, charging-sharing can be employed in a situation that we know, for sure, that two node voltages in a circuit change in a complementary manner. In [10], Pakbaznia et al. found that circuits above the NMOS sleep transistor and circuits below the PMOS sleep transistor in MTCMOS eventually flip their voltage levels in either active-to-sleep or sleep-to-active mode transition. By introducing a simple PMOS switch between the virtual ground and the virtual V_{dd}, charge-sharing is initiated ahead and behind the mode transitions to reduce the mode transition energy. The bit-line pair in the memory structure has the similar upward/downward node voltage situation as well.

To our knowledge, none of the previous approaches have observed that bit-line data independent pre-charging phase in the write operation might be redundant, hence have tried to pre-charge bit-line pair on the need basis. Theoretically, data awareness in the pair of a bit-line alone can avoid half of the bit-line power consumption and the additional charge-sharing mechanism can increase this ratio up to 75%, if we disregard the overheads. Section 3.4 will cover this theoretical case.

3. CONDITIONAL CHARGE-SHARING: CONCEPTS

3.1 Register File Structure and the Write Operation

Figure. 1 shows a 6-T storage cell in the conventional RegFile structure. Compared with the 6-T memory cell in the conventional SRAM, each storage cell in the RegFile is connected with multiple bit-lines to incorporate multiple read and write ports. In Figure. 1, for example, the cell has two pairs of read bit-lines (RBL1, $\overline{\text{RBL1}}$, and RBL2, $\overline{\text{RBL2}}$) and a pair of write bit-line (WBL and $\overline{\text{WBL}}$), which corresponds to the issue width of 1. Every increase of one in issue width is accompanied by an increase of two in read ports and an increase of one in write ports. Different wordlines, e.g. RWL1 and RWL2 (read word-lines) and WWL (write wordline), are selectively chosen for the specific operation of the cells.

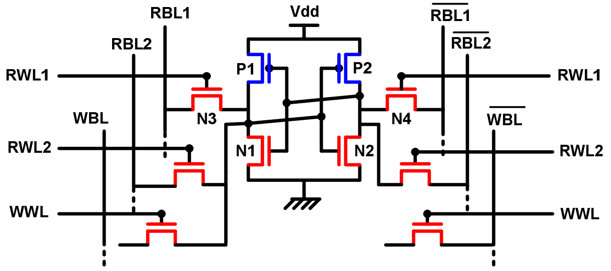


Figure. 1 A Typical 6-T cell in RegFile with issue width of one.

Figure. 2 shows one such write port to illustrate the basic write operation. As shown, WBL and complementary $\overline{\text{WBL}}$ are pre-charged to V_{dd} in every write operation. When the new cell data comes in, depending on the new cell data, one of the bit-lines will be discharged by the discharging circuitry at the bottom. This discharging circuitry is enabled by write enable (WEN) signal. The goal of the equalization transistor is to speed up equalization of the WBL and $\overline{\text{WBL}}$, during the pre-charge phase, by allowing the capacitance and the pull-up transistor of the non-discharged bit-line to assist in pre-charging the discharged bit-line. Note that each write operation is independent of the previous write operation; no matter what data we are getting for the next write operation, we fully pre-charge both of the bit-lines to V_{dd} and fully discharge one of them, therefore the write operation consumes the same amount of power on every write operation.

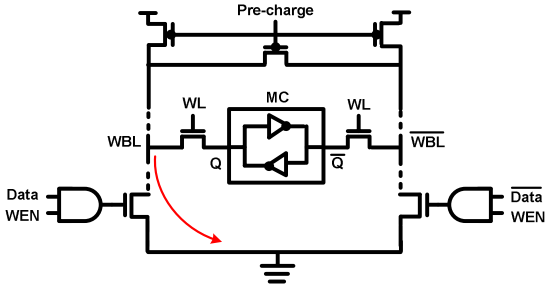


Figure. 2 Conventional write operation in the RegFile.

3.2 Motivation for the Conditional Charge-Sharing

Our conditional charge-sharing idea starts from the following observation: When a new cell data comes in during the write operation, either both of bit-lines will be flipped in the opposite direction or remain the same depending on the previously written data and the current data being written. If the new cell data is the same as the current data on bit-lines (which may be targeted to a different cell in the same column), then we do not need to unnecessarily charge one of the bit-lines to V_{dd} and then subsequently discharge it to GND. Only if the new cell data is different from the current data on the bit-lines, we will need to flip both of the bit-lines in the opposite direction, i.e., the bit-line that was previously V_{dd} has to be discharged to GND and the bit-line that was previously GND has to be charged to V_{dd}. (Here we are assuming that we are not pre-charging both of the bit-lines to V_{dd} after every write operation, which will be explained later). The latter situation provides an opportunity to apply charge-sharing between the bit-line pair so as to transfer some of the charge from the bit-line that is going to be discharged to the bit-line that is going to be charged to V_{dd} as explained below.

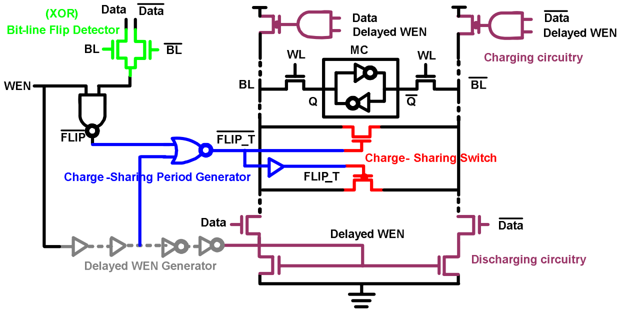


Figure. 3 Circuit design for the conditional charge-sharing scheme.

3.3 Conditional Charge-Sharing Circuitry

Based on this observation, we add a small circuit to facilitate the conditional charge-sharing operation. Our circuit design is depicted in Figure. 3. It includes a bit-line flip detector, charge-sharing period generator, a delayed write-enable (WEN) generator, and a pair of charge-sharing switches. The pre-charging and discharging circuitries are modified as well. Note that these circuit elements are added for each bit-line pair.

3.3.1 Bit-line Flip Detector

At any cycle, bit-line pair, BL and $\overline{BL}$, has a set of complementary values, e.g., (BL, $\overline{BL}$) = (V_{dd}, 0). If the same pair of data, e.g., (V_{dd}, 0) are being written to this column, then none of these bit-lines must be flipped. If, however, different values, e.g., (0, V_{dd}) are to be written to this column, then both bit-lines must be flipped, which charges a bit-line from 0 to V_{dd} and discharges the other bit-line from V_{dd} to 0. The bit-line flip detector in Figure. 3 detects this bit flipping situation and generates the FLIP and $\overline{FLIP}$ signals, indicating whether the bit-line flip will actually occur in this column.

The bit-line flip detector does not follow the conventional XOR gate design, which generally needs 6 transistors. Our XOR gate is designed to have only two NMOS transistors which reduce the area overhead. Such a 2-T XOR gate design, in turn, requires both positive and negative signals, which are fortunately available in memory structure of the RegFile; hence, no additional inverters are needed for logic value complementation.

3.3.2 Charge-Sharing Period Generator and the Charge-Sharing Switch

In our proposed design, it is crucial to allow a reasonable switching period that ensures complete charge-sharing between the bit-line pair. The charge-sharing period generator in Figure. 3, generates '0' at the output of the NOR gate whenever WEN is '0' since the NAND gate produces a '1' ($\overline{FLIP}$=1), and hence, the NOR gate will produce a '0' ($\overline{FLIP\_T} = 0$)). However, when the flip is detected and WEN is '1', the NAND gate produces a '0' ($\overline{FLIP}$=0). At this time, the other input of the NOR gate (which is the delayed WEN through the two buffers in the delayed WEN generator) is still '0', the NOR gate momentarily produces a '1' (which turns the switch ON). Shortly thereafter, when the delayed WEN becomes '1' the NOR gate produces a '0' (which turns the switch OFF). This momentary '1' value at the output of the NOR gate enables the charge-sharing switch to connect the two bit-lines. Clearly, the amount of time $\overline{FLIP\_T}$ stays at '1' is dependent on the two buffers in the delayed WEN generator. We size these two buffers such that the pulse widths of $\overline{FLIP\_T}$ and FLIP_T are large enough for the switches to perform full charge-sharing.

Clearly, design of the delay element, i.e. two buffers in the delayed WEN generator, is critical. If the delay element is designed to generate a small duty cycle pulse for driving the gate of the charge-sharing transistors, full charge-sharing will not take place, and hence, the power savings will be reduced. On the other hand, if it is designed to generate a long duty cycle pulse, then it can cause timing violation by elongating the write cycle, thereby, missing the memory access clock cycle.

The size of the charge-sharing switch also determines the time period that is spent in carrying out the charge-transfer and bringing the two bit-lines to voltage equalization. We design the switch such that it generates charge-sharing curves similar to the bit-line charging / discharging curves of the conventional write operation, which basically considers the bit-line pull-up and pull-down transistor sizes. A larger charge-sharing switch will shorten the charge-sharing period but will also result in high power consumption in its driving path.

3.3.3 Charging/Discharging Circuitry & Delay Generator

Our conditional charge-sharing mechanism does not pre-charge the bit-line pairs on every cycle. As such, pre-charging (this is also true for the discharging) of bit-lines occurs according to the actual occurrence of charge-sharing. At the end of charge sharing, BL and $\overline{BL}$ voltage levels will be equalized. Then, a bit-line will be charged to V_{dd} (by the charging circuitry shown in Figure. 3) while the other bit-line will be discharged to GND (by the discharging circuitry shown in Figure. 3). In this case, the full-swings on BL and $\overline{BL}$ are avoided because both bit-lines start from an equal voltage of $V_{dd}/2$ and move in opposite directions to V_{dd} and GND levels. As a consequence, we save power as explained in the next section.

The delayed WEN generator in Figure. 3 is designed to provide two features: i) produce a suitable turn-on time period of the charge-sharing switch (owing to the two buffers of delay element), and ii) control the delay of bit-line charging / discharging transistors such that it prohibits any of the bit-lines from being connected to V_{dd} or GND during the charge-sharing period, thereby, avoiding a short-circuit path. Notice that the delayed WEN generator is shared among all the columns, and hence, we can reduce the area and power overhead.

3.4 Performance Estimation

To estimate the achievable power savings in our charge-sharing scheme, we provide a pictorial explanation. Figure. 4 shows how the charge on the bit-line pair changes in consecutive write operations both in the conventional scheme and in the charge-sharing scheme. The color of each of bar-graph pairs represents the charge status of the bit-lines, i.e., blue means 'charged' and white means 'discharged'. Moreover, the transition from cycle n to n+1 corresponds to the bit-line flip whereas the transition from cycle n+1 to n+2 corresponds to the bit-line non-flip. By showing these two cases, we estimate the average energy savings in both flip and non-flip situations.

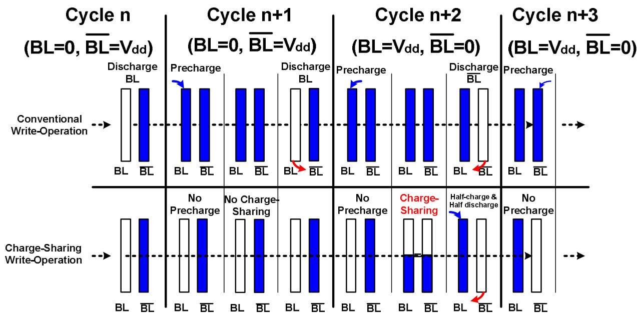

Cycle n
(BL=0, $\overline{BL}$=Vdd)
Cycle n+1
(BL=0, $\overline{BL}$=Vdd)
Cycle n+2
(BL=Vdd, $\overline{BL}$=0)
Cycle n+3
(BL=Vdd, $\overline{BL}$=0)
Discharge BL
Precharge
Discharge BL
Precharge
Discharge $\overline{BL}$
Precharge
Conventional Write-Operation
BL $\overline{BL}$
No Precharge
No Charge-Sharing
No Precharge
Charge-Sharing
Half-charge & Half discharge
No Precharge
Charge-Sharing Write-Operation

Figure. 4 Pictorial comparison between the conditional charge-sharing vs. conventional techniques.

We assume that the initial bit-line state of (BL, $\overline{BL}$) = (0, V_{dd}) in cycle n. In cycle n+1, in the conventional scheme, we assume the same data value, i.e., (BL, $\overline{BL}$) = (0, V_{dd}), comes in. Since the bit-line pair needs to be fully pre-charged before write operation, the BL has been fully charged at this time. Next, the new data set discharges BL. In cycle n+2, we assume that data value (BL, $\overline{BL}$) = (V_{dd}, 0) comes in. Again, BL will have been fully pre-charged, but this time $\overline{BL}$ is fully discharged by the new data. To sum up, a total of ***four*** bit-line charge/discharge operations occur. In contrast, in the charge-sharing scheme, bit-lines have not been pre-charged at cycle n and the same data value, (BL, $\overline{BL}$) = (0, V_{dd}), comes in at cycle n+1. During the write operation at cycle n+1, none of the bit-lines are discharged since currently discharged BL matches the new data set. When a different data value (BL, $\overline{BL}$) = (V_{dd}, 0) comes in at cycle n+2, the bit-line flip detector is triggered and the charge-sharing switch is turned on. As a result, the charge stored in $\overline{BL}$ is transferred to BL. After the bit-line pair reaches voltage equilibrium, the (delayed) write circuitry charges BL to V_{dd} from $V_{dd}/2$ while discharges $\overline{BL}$ to GND from $V_{dd}/2$. To sum up, equivalent of only ***one*** bit-line charge/discharge operation occurs. Therefore, in the ideal case as described above, the charge-sharing solution can save up to 75% of the bit-line power dissipation for consecutive write of a same (e.g. between n and n+1 cycles) and an opposite (e.g. between n+1 and n+2 cycles) values into any cell(s) on the same column of the RegFile. The power consumed in the cell data flipping and the extra power consumed in the additional circuitry was not considered in this example.

We point out that the conditional charge-sharing scheme may not be applied to the conventional SRAM array which has a single bit-line pair per-column that is shared between the read and write operations. Since the read operation needs unconditional pre-charging and our proposed technique, if applied to the SRAM array, will modify the pre-charging logic with conditional charging, which will increase the time required for performing the read operation. The read access delay, however, sets the overall access time of the SRAM array, and hence, the conditional charge-sharing scheme will result in performance degradation. In contrast, the RegFile has dedicated and separate bit-lines for read and write operations into a register and therefore it is amenable to the application of the proposed charge-sharing scheme.

3.5 Overhead Estimation

3.5.1 Area Overhead

For brevity, we estimate the area overhead for the conditional charge-sharing scheme by calculating the ratio of the number of additional transistors to the transistor count of the conventional design. The additional parts in each column are: a) bit-line flip detector, b) charge-sharing switch, c) charge-sharing period generator, d) delayed WEN generator, e) the modified pre-charging/discharging circuitry. Hence, the number of added transistors in a pair of bit-line column is 34 transistors: 6-T (flip detector) + 2-T (switch) + 6-T (period generator) + 12-T (delay generator) + 8-T (modified pre-charging / discharging).

For a conventional RegFile with two read ports and one write port, which is 32 bit wide and has an issue width of 1, the number of transistors in each column is 653: 64 (rows) x 10 (4 transistors in the cross coupled inverter pair, 4 access transistors for two read ports and 2 access transistors for the write port) + 3-T (two pre-charge PMOS transistors and a PMOS equalizer) + 10-T (two-input NAND which is 4-T and an NMOS transistor which is 1-T, in each side of bit-line). As a result, the proposed technique needs a total of 34-T+653-T= 687 transistors per column, i.e., the resulting estimated overhead is 34/653 = 0.051 (i.e. 5.1%). Note that the transistors in the delayed WEN generator can be shared over all columns hence the actual area overhead is indeed smaller. However, we included these transistors in the area penalty estimate to be conservative.

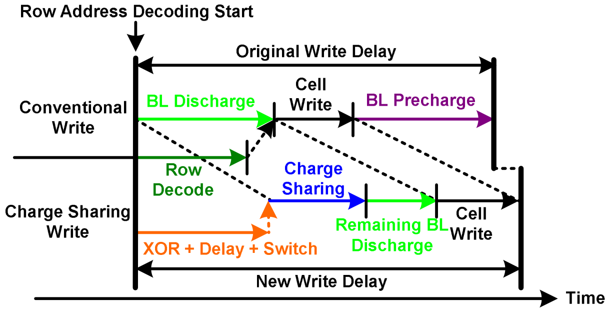


Figure. 5 Decomposition of a write delay in both designs.

3.5.2 Delay Overhead

In Figure. 5, we show a cycle period during the write operation. Unlike the large SRAM arrays, the row decoding time is not critical since the number of rows in the RegFile is relatively small. Moreover, there is no need to wait for column decoding in the RegFile, i.e., discharging of the bit-lines can start at the beginning of the cycle. During write to a conventional RegFile, either bit-line or bit-line-bar starts getting discharged based on the new cell data. Subsequently, the cell writing and the bit-line pre-charging take place. In contrast, during write to a conditional charge-sharing RegFile, some amount of time is needed to turn on the data dependent charge-sharing switch so that the conditional charge-sharing can subsequently take place. When the charge-sharing is completed, charging / discharging circuitry perform the remaining bit-line charging / discharging starting from the equalized voltage level of $V_{dd}/2$. The cell writing occurs last.

In the conditional charging-sharing scheme, we save some amount of time by avoiding redundant bit-line pre-charging; however, we spend extra time to share the charge between the bit-line pair to bring these lines into an equilibrium voltage. There is no delay increase when the new data is the same as before. On the other hand, when the data is flipped we need extra circuitry to facilitate the charge sharing which increases the write delay.

4. EXPERIMENTAL RESULTS

We use Hspice for the power and delay calculation. The proposed technique is implemented on a 64 x 32-bit and a 128 x 32-bit RegFiles. Conventional RegFiles (with two pre-charge and an equalization PMOS transistor configuration of Figure. 2) were also simulated for comparison purposes. For the technology file, we used the 65nm PTM from [11]. The temperature was set to 75 ℃ and the V_{dd} was set to 1.0V.

Figure. 7 shows the waveforms of the consecutive write operations with both schemes in the 64 x 32-bit RegFile. In each waveform, we show two cycles: the first cycle corresponds to the actual cell data flip while the second cycle corresponds to the non-flipping case. Here the initial status of BL and $\overline{BL}$ is assumed to be 0 and V_{dd}. During the first write cycle in the conventional scheme, the order of operation is: pre-charge BL, discharge one of the BL's, and perform the cell flip. During the second cycle, pre-charge and discharge of BL exist, which consumes redundant power. In contrast, during the first write cycle in the conditional charge-sharing scheme, the order of operation is: charge sharing between BL's, charging / discharging remaining BL, and performing the cell flip. Notice that charge-sharing does not consume power and the remaining BL charging/discharging consumes less power compared to the conventional counterpart. In the second cycle, there is no bit-line status changes, which ideally (ignoring bit-line leakage) does not consume any power. One important point is that bit-line flips do not necessarily result in the occurrence of cell flip, and that those two outcomes are independent of one another.

Figure. 7 also shows the current waveform out of the V_{dd} line that powers up all circuitry for a single column in the RegFile, including the memory cells attached to the column, the pre-charge logic for the conventional scheme plus the additional charge-sharing logic for the proposed scheme. Notice that the current waveforms for the case of writing '1' into a cell storing '0' and the case of writing '1' into a cell storing '1' is nearly identical. This confirms the data-independent power consumption of the write operation in the conventional RegFile design. In contrast, the current waveform of the conditional charge-sharing RegFile design is strongly dependent on whether or not a bit-line pair flip occurs. During the first write cycle (cell flip case), we dissipate some amount of power due to switching activity inside the circuitry that is responsible for charge sharing. This overhead reduces the theoretical energy saving in the bit-flip case from 50% to average of 39.2%. During the second write cycle (no cell flip case), there is some amount of power consumption due to activity which is caused by WEN in the added charge-sharing circuitry.

TABLE 1 THE ENERGY SAVINGS AND DELAY PENALTY

RegFile Size	Bit-line status	Energy savings in Charge-sharing design over conventional design	Delay penalty in write operation
64	Flip	40.4	16.2%
	Non-flip	90.3	16.2%
128	Flip	38.0	16.2%
	Non-flip	90.1	16.2%

TABLE 1 shows another set of experimental results for power dissipation and delay. Compared to the conventional RegFiles, the proposed technique achieves an average of 39.2% and 90.2% energy savings in the two RegFiles for the flipping and non-flipping writes, respectively. The energy savings come from i) reduced bit-line charge/discharge swing due to charge-sharing (in the 'Flip' case), and ii) elimination of unnecessary pre-charging (in the 'Non-flip' case). In both cases, the delay increase is 16.2%.

TABLE 2 NORMALIZED POWER DECOMPOSITION

Functional Block	RegFile Size	
	64	128
Bit-line flip detector & Switching period	49.7	49.0
Delayed WEN generator	3.9	3.8
Charge-sharing switches	0.9	1.0
Bit-line charging	45.5	46.2

In TABLE 2, we decompose the overall power consumption of the conditional charge-sharing design into its constituent parts. As shown, the bit-line flip detector and switching period generator consume almost half of the power because i) switching period generator needs to drive the large charge-sharing switches in case of the bit-line flip and ii) due to charge sharing, the portion of bit-line charging in a write operation power is dramatically reduced.

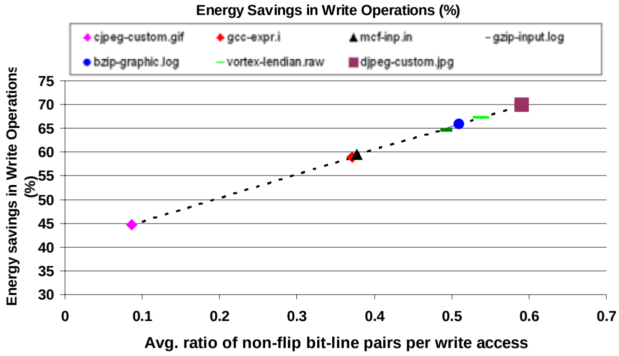


Figure. 6 Average ratio of non-flip bit-line pair per write access to RegFile and the energy savings.

To estimate the energy savings, we used SimpleScalar [12], and ran applications from SPEC2000INT benchmark suite [13] with respective default input file, and two applications from the MediaBench suite [14] with custom input files. The architectural simulator was configured to have an issue width of 4 and was modified to generate information about the number of bits flipped during RegFile write operations over a complete run of each program. With this information and cycle-level energy saving values (for the 64 x 32-bit RegFile) in TABLE 1, we computed and report the energy savings in Figure. 6. Clearly, energy saving of the proposed scheme varies linearly as a function of the average ratio of bit-line non-flips per write operation to the RegFile. Our experimental results show that, on average, we achieve 61.5% energy savings per write operation in these programs.

5. CONCLUSION

In this paper, we propose a write power reduction mechanism in the RegFile, which exploits a conditional charge-sharing between the complementary bit-line pair in each column. The data similarity on the bit-line pair due to the previous write and the data being written in the current write operations provides a chance to avoid redundant pre-charging in every cycle, and the data dissimilarity in the same context gives a chance to employ a charge-sharing scheme. Experimental results show an average of 61.5% of energy savings with 5.1% area overhead and 16.2% increase in write access delay.

6. REFERENCES

[1] J-H. Kim et al., “Fixed-Load Energy Recovery Memory for Low Power,” *Proc. of IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI System Design*, pp. 145-150, Feb. 2004.

[2] J-H. Kim et al., “Constant-Load Energy Recovery Memory for Efficient High-Speed Operation,” *Int’l Symposium on Low Power Electronics and Design*, pp. 240-243, Aug. 2004.

[3] B-D. Yang et al., “A Low-Power ROM Using Charge Recycling and Charge Sharing Techniques,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 641-653, Apr. 2003.

[4] K. W. Mai et al., “Low-Power SRAM Design Using Half-Swing Pulse-modulation Techniques,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11. pp. 1659-1671, 1998.

[5] B-D. Yang and L-S. Kim, “A Low-Power Charge-Recycling ROM Architecture,” *IEEE Trans. on Very Large Scale Integration Systems*, vol. 11, no. 4, pp. 590-598, Aug. 2003.

[6] K. Kanda et al., “90% Write Power-Saving SRAM Using Sense-Amplifying Memory Cell,” *IEEE Journal of Solid State Circuits*, vol. 39, no. 6, pp. 46-47, Jun. 2004.

[7] S-P. Cheng et al., “A Low-Power SRAM Design Using Quiet-Bitline Architecture,” *Int’l Workshop on Memory Technology Design and Testing*, pp. 135-139, Aug. 2005.

[8] A. Karandikar and K. K. Parhi, “Low Power SRAM Design Using Hierarchical Divided Bit-Line Approach,” *Proc. of the Int’l Conf. on Computer Design*, pp. 82-88, Oct. 1998.

[9] B-D. Yang and L-S. Kim, “A Low-Power ROM Using Single Charge-Sharing Capacitor and Hierarchical Bitline,” *IEEE Trans. on Very Large Scale Integration Systems*, vol. 14, no. 4, pp. 313-322, Apr. 2006.

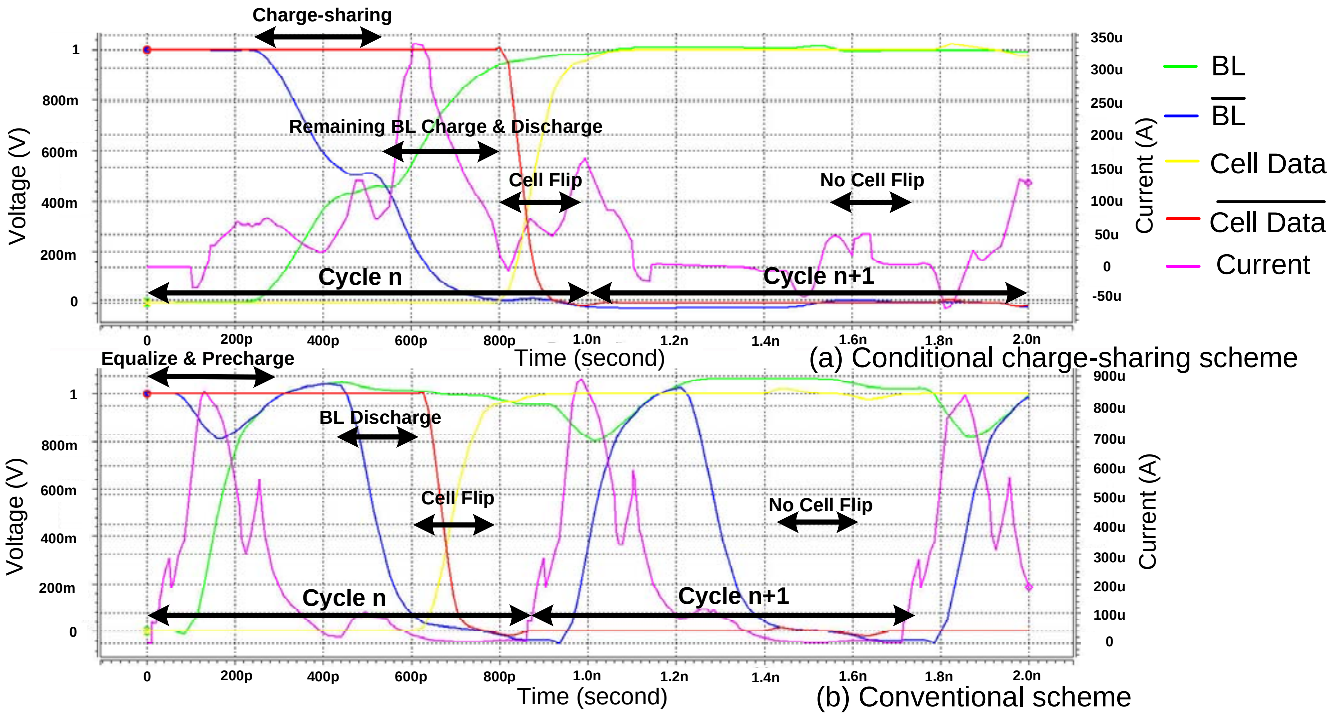
[10] E. Pakbaznia, F. Fallah, and M. Pedram, “Charge Recycling in MTCMOS Circuits: Concept and Analysis,” *Proc. of Design Automation Conf.*, pp. 97-102, Jul. 2006.

[11] http://www.eas.asu.edu/~ptm/

[12] http://www.simplescalar.com

[13] http://www.spec.org

[14] C. Lee, et al., “MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems,” *Proc. of Int’l Symposium on Microarchitecture*, 1997.



(a) Conditional charge-sharing scheme

(b) Conventional scheme

Figure. 7 Electrical waveforms for various signals when writing a ‘11’ sequence into a cell that had initially stored a value of ‘0’. (a) Conditional charge-sharing scheme (b) Conventional scheme.