Minimizing Power Dissipation during Write Operation to Register Files

Kimish Patel, Wonbok Lee, **Massoud Pedram**

University of Southern California
Los Angeles CA

August 28th, 2007

Outline

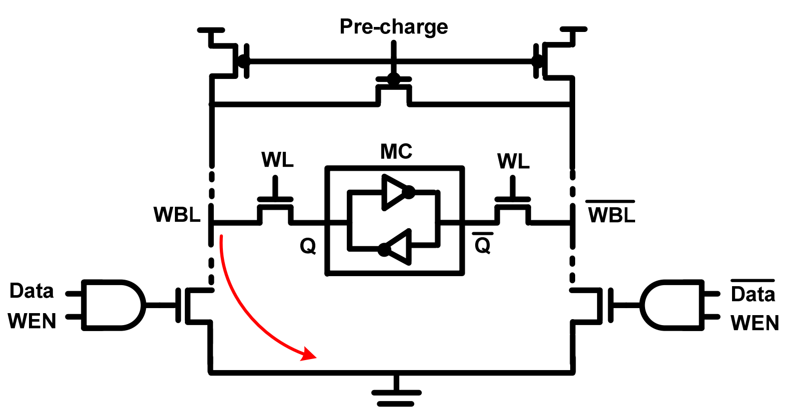
- Introduction
- Conditional Charge-Sharing Architecture
  - Bit-line Flip Detector
  - Charge-Sharing Period Generator
  - Charge-Sharing Switch
- Energy Savings Estimation
- Delay Overhead
- Experimental Results
- Conclusion

Background

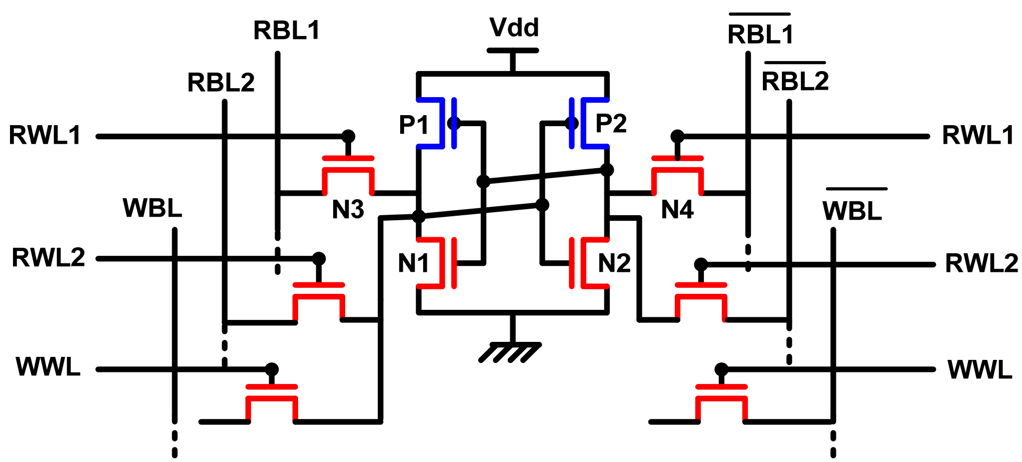
- Current state-of-the-art microprocessors tend to have very wide issue-width.
  - Increased number of read/write ports & word-line/bit-line capacitance make the Register Files a power hungry block on the chip.
    - Register Files are also known to be the hottest blocks on the chip.
- Typically, the write operation in register files consumes more power over the read operation.
  - The read operation partially discharges one bit-line within each bit-line pair whereas the write operation fully discharges one bit-line within each pair.
- Generally, the word-line power consumption is fixed for both read and write operations.
  - A bit-line charging/discharging consumes a significant portion of the register file power, making it a good target for low power design.

Introduction

- Conventional memory structure vs. register files.
  - Single vs. multiple read/write ports per column
  - For every read/write operation, all the bit-line pairs in the row are accessed in register files.
- Why is the write operation targeted for power optimization?
  - For the write operation, we know the new data value to be stored and the old data is still present on each of the bit-line pairs.
    - In this scenario, comparison of the original and new data is feasible.
    - This is not the case for the read operation,.



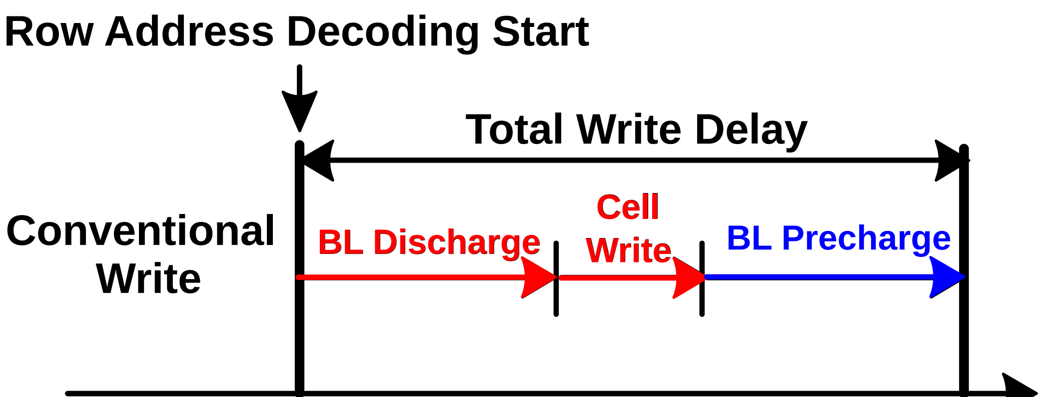
< Memory Structure >

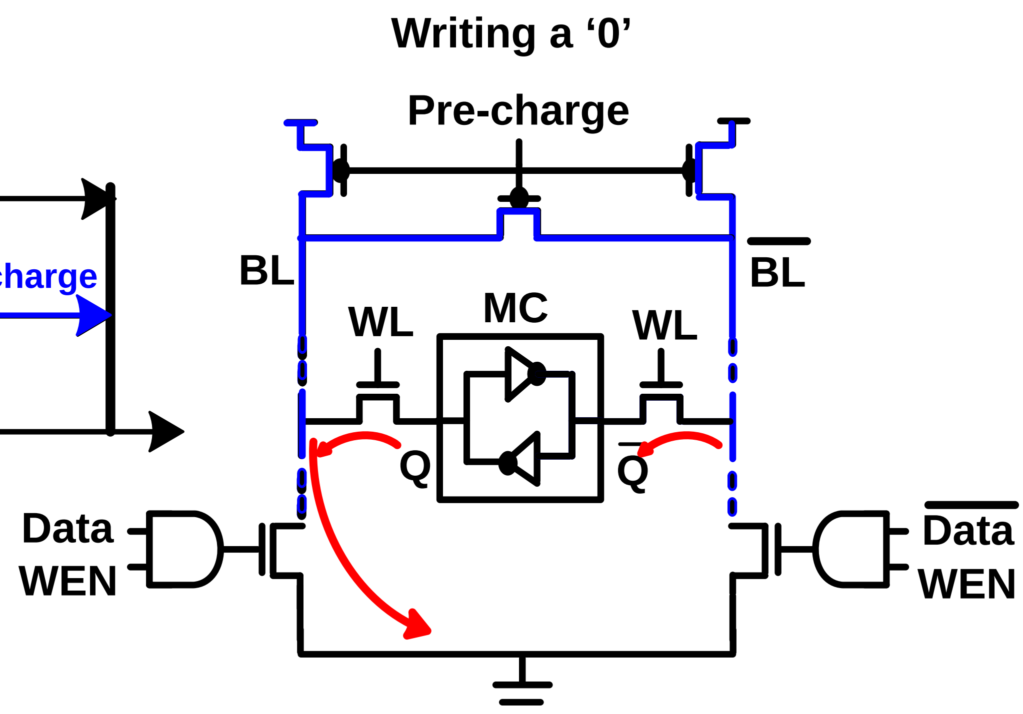


< Register File Structure >

Introduction (Cont'd)

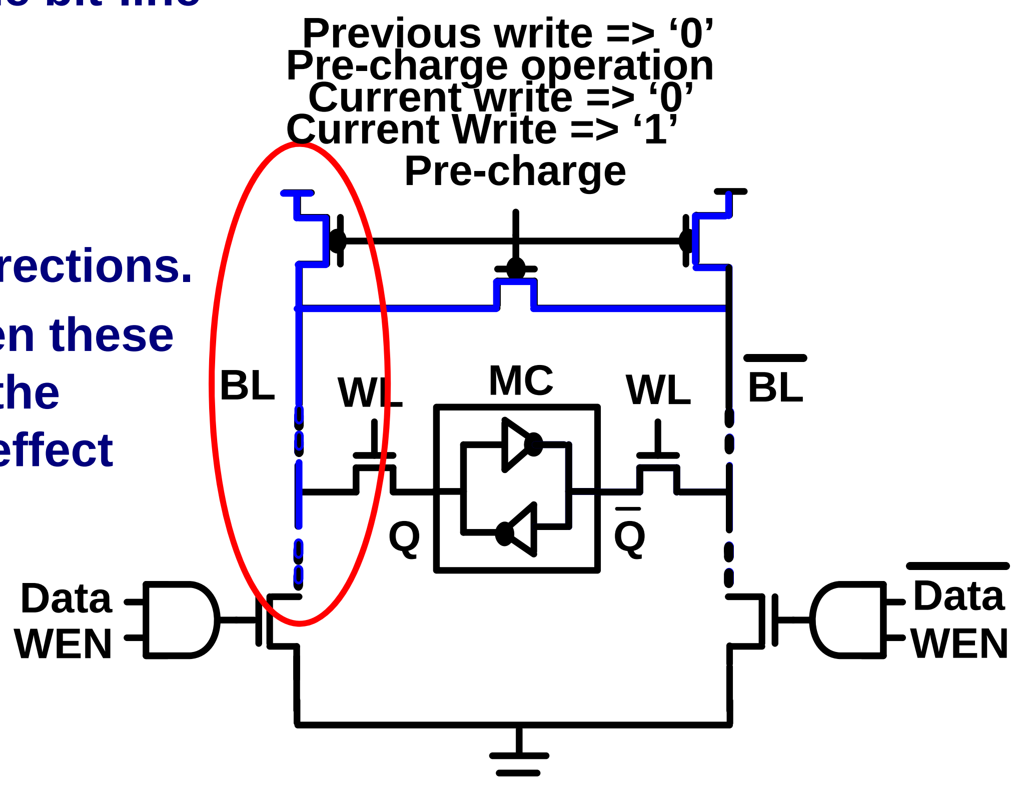
- **Conventional write operation to register files comprises of three steps;**
  - **Fully discharge one of the bit-lines in a column**
  - **Write into the target cell**
  - **Pre-charge the discharged bit-line to V_{dd} again**
- **These three steps are independent of the value being written.**



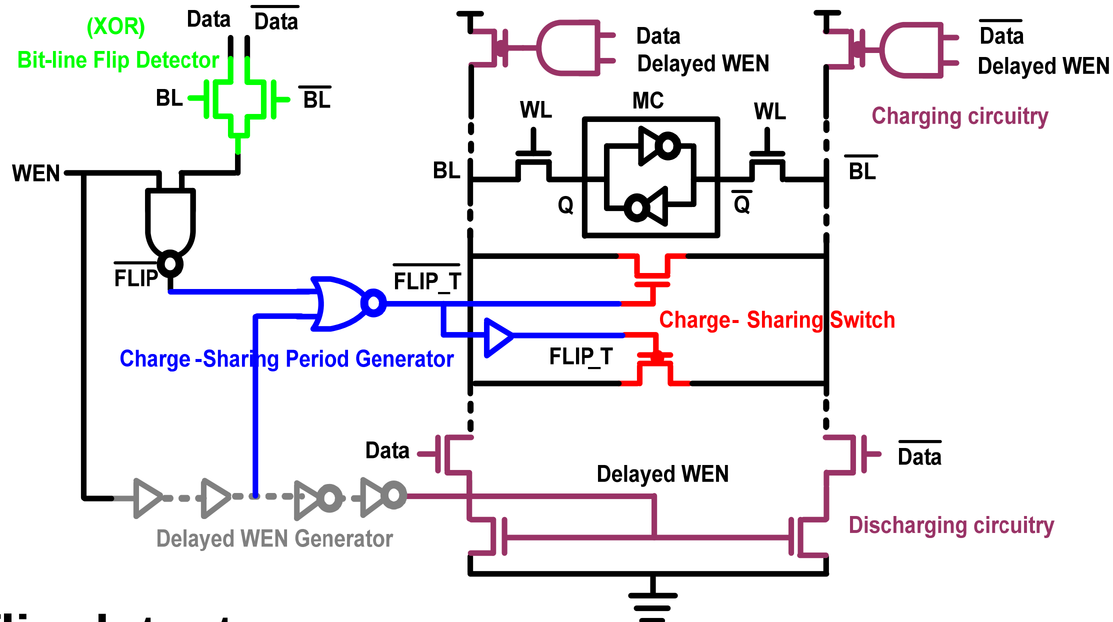


Motivation

- We compare
  - Current data value on the bit-line pair and new data being written to the memory cell.
- If they are the same for a write operation,
  - Redundant pre-charging of the bit-line follows the full discharging.
  - This step can be avoided.
- If they are different,
  - Bit-lines swing in opposite directions.
  - Charge can be shared between these two bit-lines so as to reduce the amount of charge needed to effect the change.



Conditional Charge-Sharing Architecture



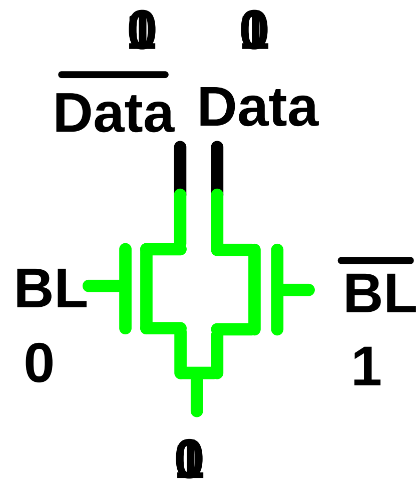
- Bit-line flip detector
  - Detects if the current value being written is different from the previous write.
- Charge-Sharing (CS) period and delayed WEN generators
  - Control charge-sharing time period.
- Charge-Sharing switch
  - Facilitates charge-sharing between BL and $\overline{\text{BL}}$.

Bit-line Flip Detector

- **Bit-line flip detector**
  - **Assume the current data on bit-lines is '0' and '1'.**
  - **If the new data being written is '0'**
    - **Output is '0'.**
  - **If the new data being written is '1'**
    - **Output is '1'.**
- **Non-conventional XOR gate for power and area savings.**
  - **For this type of XOR design, we need signal and its complement.**
    - **Fortunately they are available in memories .**

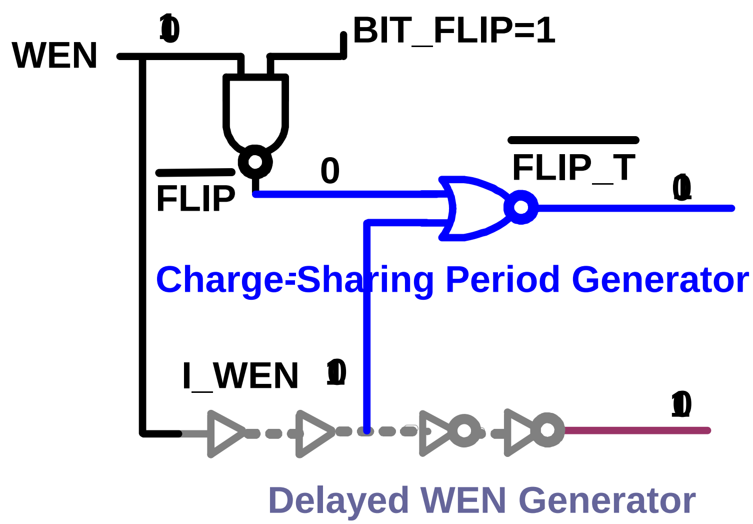
(XOR)

Bit-line Flip Detector

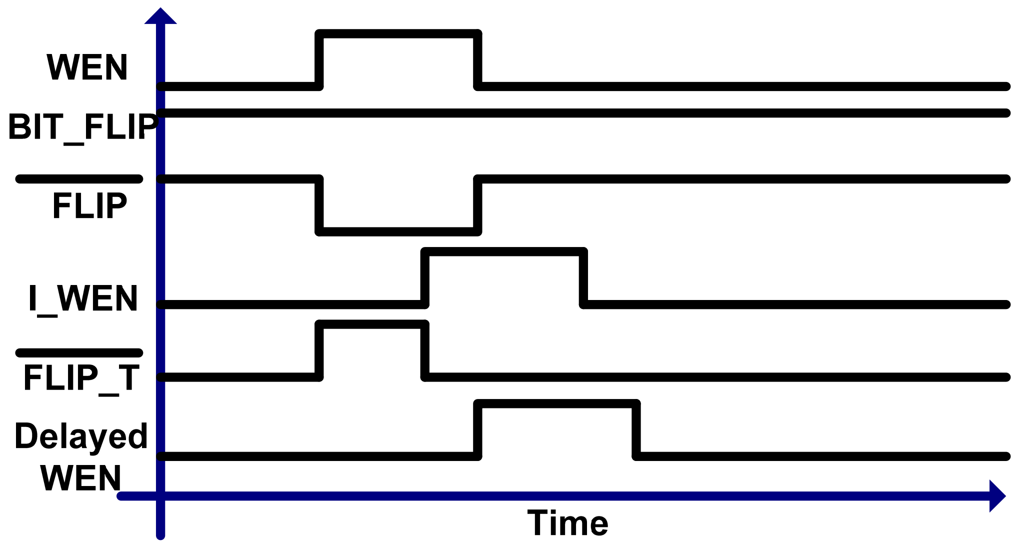


CS Period Generator

- **When WEN = 0**
  - **$\overline{\text{FLIP\_T}}$ is 0 hence the charge-sharing switch is disabled.**
- **When WEN = 1 and BIT_FLIP = 1**
  - **NAND gate produces $\overline{\text{FLIP}}$ = 0 and other input is still ‘0’ due to buffer chain.**
    - **Hence $\overline{\text{FLIP\_T}}$ is ‘1’ which in turn enables the charge-sharing switch.**
  - **After a while when WEN propagates through the buffer chain as ‘1’ at the other input of the NOR gate.**
    - **It makes $\overline{\text{FLIP\_T}}$ = ‘0’, thereby disabling the charge-sharing switch.**
  - **WEN further propagates to Delayed WEN, the charge-sharing switch is disabled.**



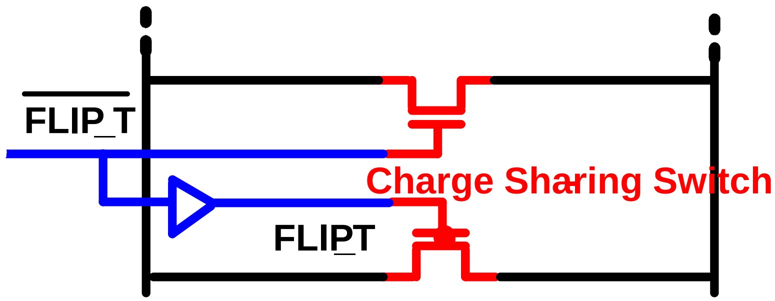
CS Period Generator: Timing Diagram



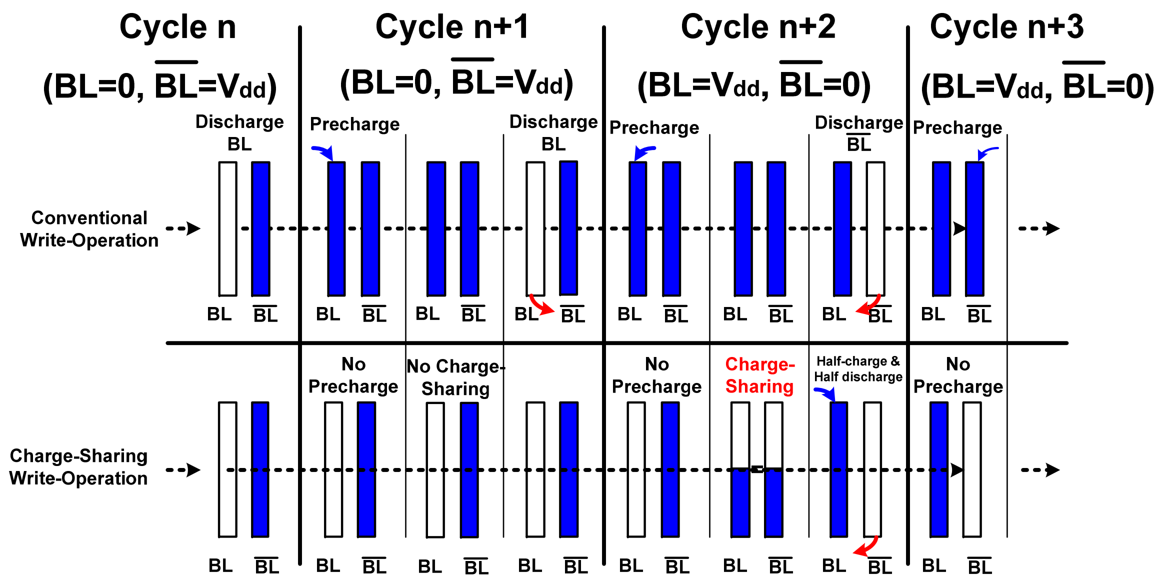
- Delayed WEN and charge-sharing period generator circuits must
  - Generate long enough time period to exploit full charge-sharing
  - Make sure that the CS operation ($\overline{\text{FLIP\_T}}$) does not overlap with the actual write operation (Delayed WEN) to avoid the direct current path.
- Delayed WEN (buffer chain) is shared among all the columns.
  - Area and power overheads can be reduced.

Charge-Sharing (CS) Switch

- Charge-Sharing Switch
  - Enables charge-sharing between bit-line pair if a bit flip is detected.
- Charge-sharing is enabled only for the period of time generated by the CS period generator.
  - CS switch should be large enough so as to allow full charge-sharing during the time period generated.
    - In our design, it is sized such that the current dissipation curve during charge-sharing is same as that of the conventional pre-charge operation.
  - Similarly, the CS period generator must be sized to create a long enough charge-sharing period.

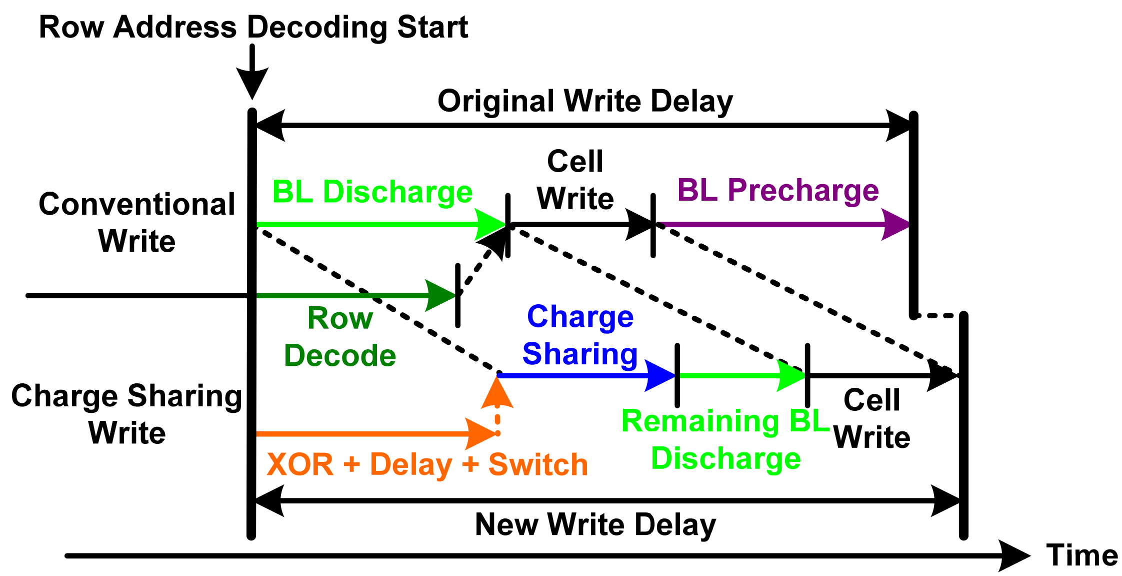


Ideal Energy Saving



- Write operation pattern on this estimation
  - Initial data at cycle n is ‘0’ (w.r.t BL).
  - At cycle n+1, same data is written.
  - At cycle n+2, different data is written.
- According to this estimation, ideal energy saving is 75%, without the consideration of additional circuitry.
- This scheme cannot be applied to the conventional SRAM since the read and write operations share the same pair of bit-lines.

Delay Overhead



- Conditional charge-sharing based write operation
  - BL pre-charging is not necessary.
  - Additional delay is incurred to perform the remaining charge/discharge of the BL pair .
- Experimental results show a delay increase of 16.2%.
  - In general, read operation is the critical operation which determines the cycle period.

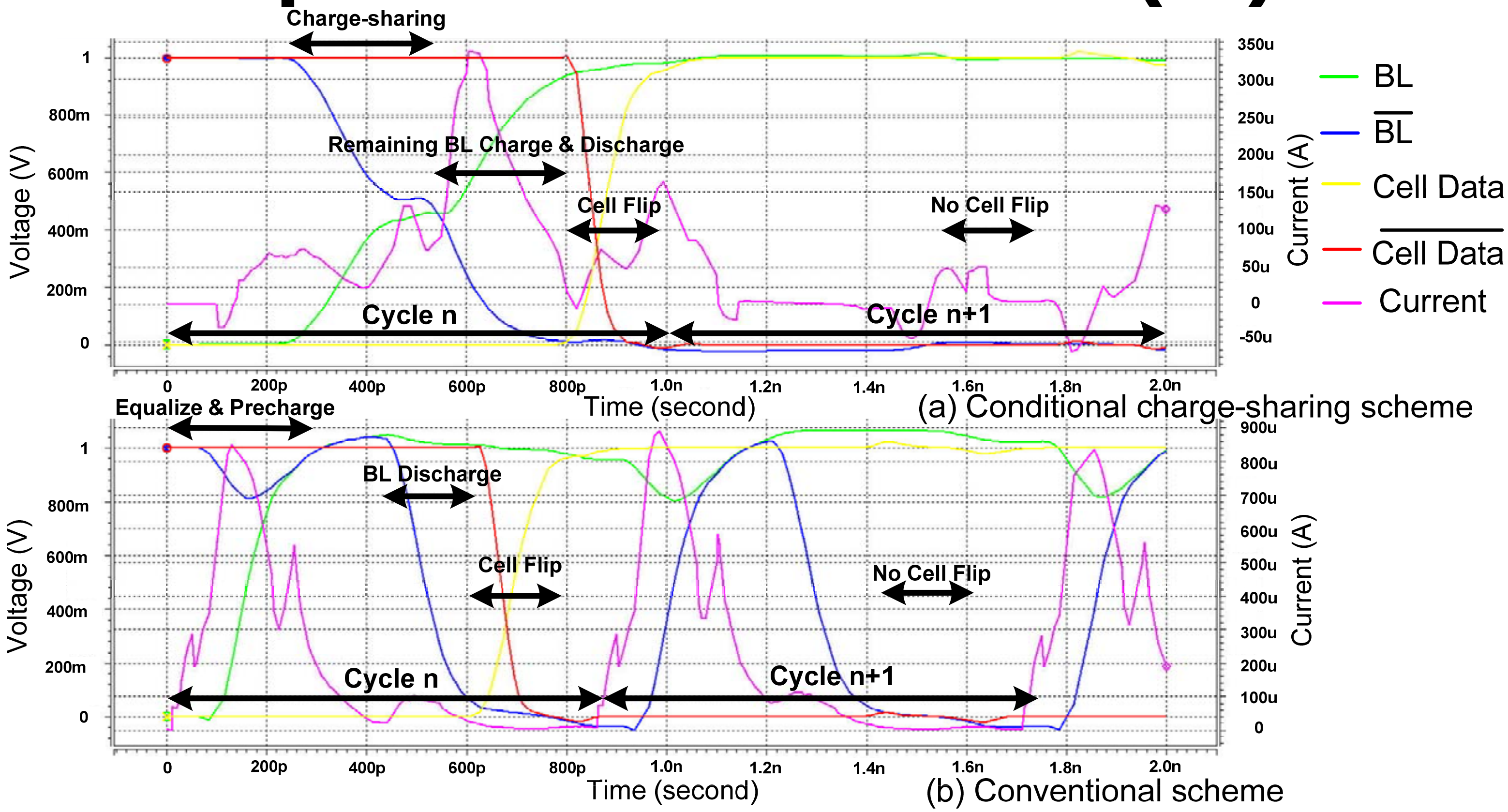
Experimental Results (1/3)

- Experimental Setup
  - A conventional register file and the proposed architecture were designed and simulated in Hspice.
  - Two configurations of the register files
    - 64 (H) X 32 (W)
    - 128 (H) X 32 (W)
  - Used 65nm PTM, 75℃, and V_{dd} of 1.0V.
- Experimental results show average energy savings of 39.2% and 90.2% for the data flip and non-flip cases, respectively.

Register file size	Bit-line status	Energy Saving(%)
64	Flip	40.4
	Non-flip	90.3
128	Flip	38.0
	Non-flip	90.1

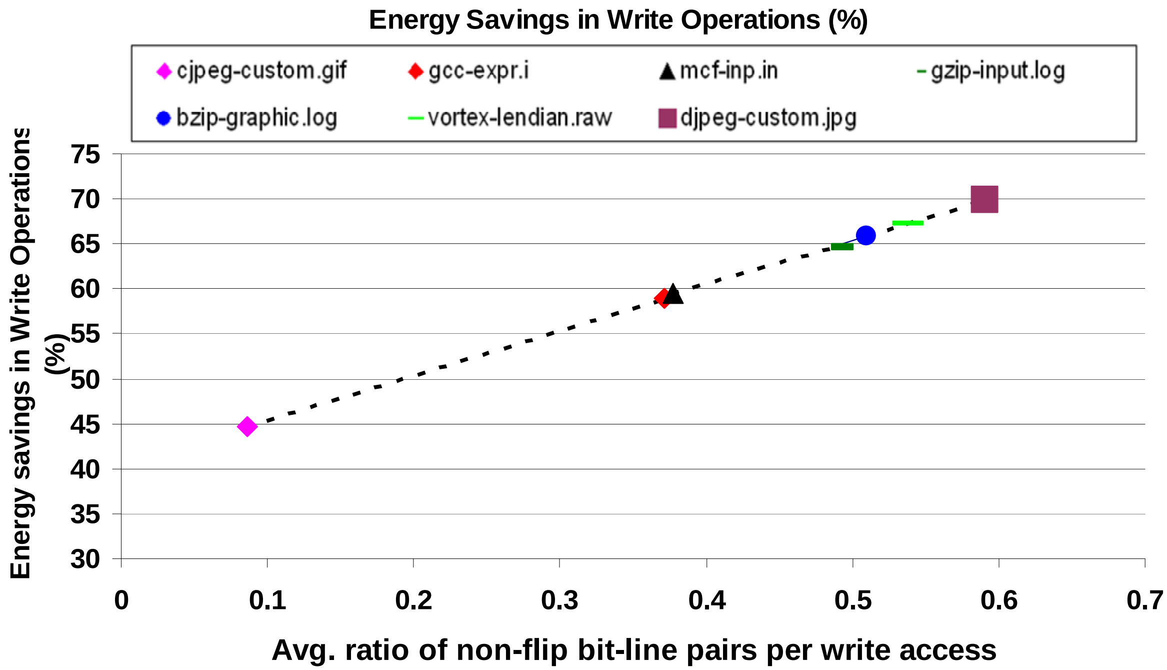
- Energy for the two cases is saved differently:
  - Reduced bit-line charge/discharge swing for the flip case
  - Elimination of bit-line (pre-)charging for the non-flip case.

Experimental Results (2/3)



- For conventional write operation (bottom)
  - Dissipated current remains the same independent of the new & old data.
- For charge-sharing based write operation (top)
  - Dissipated current varies depending on the flip/non-flip case.
  - During cell-flip, current is high due to the activity in charge-sharing block.

Experimental Results (3/3)



- **In practice, actual energy savings is a function of the ratio of non-flipped to flipped bit-lines.**
  - **We used *simplescalar* along with some SPEC2000INT benchmarks programs to estimate the ratios and then calculate the power savings.**
  - **Average 61.5% energy savings per write operation.**
- **As the ratio increases, i.e., more writes end up with non-flips, and thus energy savings increases.**

Conclusion

- We presented a charge-sharing based energy reduction scheme for write operation to register files. The technique
  - Exploits the data similarity between consecutive writes
  - Cuts the bit-line swing in half, i.e. when different data is written.

- Experimental results show that
  - Average energy savings of 39% and 90% for the cases of data flip and non-flip, respectively. The delay penalty is16.2%. Area penalty is negligible.
  - For the SPEC2000INT benchmark suite, the average energy saving is 61.5% per write operation.