A Scalable Design Methodology for Energy Minimization of STTRAM: A Circuit and Architecture Perspective

Subhó Chatterjee, Student Member, IEEE, Mitchell Rasquinha, Sudhakar Yalamanchili, Senior Member, IEEE, and Saibal Mukhopadhyay, Member, IEEE

Abstract—In this paper, we analyze the energy dissipation in spin-torque-transfer random access memory array (STTRAM). We present a methodology for exploring the design space to minimize the energy dissipation of the array while maintaining required read and write quality for a given magnetic tunnel junction technology. The proposed method shows the need for proper choice of the silicon transistor width and array operating voltage to minimize the energy dissipation of the STTRAM array. The write energy is found to be $10 \times$ greater than read energy. Hence, read-write ratio becomes a crucial factor that determines energy for STTRAM last level caches (L2). An exploration is performed across several architectural benchmarks including shared and non-shared caches for detailed energy analysis.

Index Terms—L2 caches, magnetic tunneling junction (MTJ), spin torque transfer RAM (STTRAM), tunneling magneto-resistance (TMR).

I. INTRODUCTION

MAGNETIC RAM (MRAM) has surfaced as a promising candidate in the quest for a universal memory. It has distinct advantages over SRAM, as the cell size of SRAM is relatively large (for a 6T structure) thereby limiting the amount of memory that can be integrated within a die. DRAMs have small cell size (1T) but have the undesirable characteristic of requiring periodic refresh. Flash memories, suffer from slow-write operations with limited read/write endurance. In contrast, the conventional MRAM structure has the desired properties of high endurance, fast read/write speed, non volatility and most importantly, zero standby power. However, it suffers from high write power consumption and poor scalability. The use of spin-torque-based magnetization has been shown to improve the preceding shortcomings of the conventional MRAM. This makes STTRAM a more promising prospect for becoming a universal memory [1], [2].

The structure and working principles for magnetic tunneling junction (MTJ) have been well studied [3]–[5]. The cell structure of a STTRAM consists of a MTJ connected in series with a transistor. This cell is connected between the bit lines and the select lines whereas word line is responsible for switching of the transistor. The MTJ consists of two ferromagnetic layers separated by a dielectric layer referred to the spacer (usually MgO) (see Fig. 1). The magnetization of one ferromagnetic layer is fixed. The magnetization of the other layer can be controlled by the injection of spin polarized electrons. Switching occurs if current greater than the critical value flows through the structure in the proper direction. The MTJ offers different resistances in the two modes of magnetization (parallel and anti-parallel).

New MTJ structure and materials are being constantly explored for storage options parallel is taken to correspond to “0” and anti-parallel corresponds to “1”. The resistance of the MTJ in the anti-parallel state is higher than that in the parallel state. The difference between the resistances in parallel and anti-parallel states with respect to resistance of the anti-parallel state is called tunneling magneto-resistance (TMR). A high TMR indicates a larger difference between resistances in the parallel and anti-parallel modes and facilitates reading. Hence we conclude that TMR and switching current are the two major MTJ parameters determining MTJ characteristics.

Operation of STTRAM cells, and memory arrays have been demonstrated experimentally [6], [7]. Research in the STTRAM domain has primarily been focused on reducing the switching current requirement of the MTJ device and development of robust design strategies. The prior work on STTRAM can be categorized into work on MTJ device structure and work on circuits. New MTJ structure and materials are being constantly explored to reduce the switching current and improve the TMR [8], [9]. From the circuit side, there have been efforts to develop behavioral models for MTJ [10]. Given a MTJ device, the characteristics of the memory cell strongly depend on the access transistor in the cell (see Fig. 1). Proper design of the access transistor is crucial for array performance, robustness, and energy dissipation. The effects of the access transistor on performance and robustness of a cell have recently been studied [11]. Chen, et al. has presented circuit models to capture the dynamic response of the cell for STTRAM design margin exploration [12].
Due to its non-volatile nature and very low standby power, STTRAM arrays are suitable for last level caches (LLC), e.g., L2, in microprocessors. Further, very small cell size can help to improve the integration density and hence, allow a larger size L2 cache relative to the SRAM equivalent. A larger size reduces the L2 miss-rate and consequently off-chip memory accesses leading to better performance and lower energy consumption. One of the primary challenges in using STTRAM as an L2 cell is the energy dissipation during read and write operations. In particular, due to the requirement of high-write current density, write energy can dominate the total array energy for L2 motivating architectural optimizations that minimize write operations. Hence, detail analysis and cooptimization at both the circuit and architecture level are necessary to understand how to design the STTRAM cell (given a MTJ), array, and LLC to minimize the energy dissipation of the memory array. While STTRAM was evaluated against existing memory technologies and its architectural impact and evaluation for 3-D microprocessor stacking was explored [13], [14], the architectural implications for an energy optimal STTRAM solution and the impact of scalability remain to be studied.

In this paper, we present a detail analysis of the energy dissipation of an STTRAM array and explore the cell and architecture design space to minimize the array energy dissipation in 180-nm TSMC technology. In particular, this paper makes the following contributions.

1) A comprehensive analysis of the array energy is performed using a detail dynamic model of the STTRAM array.
2) A methodology to co-design the cell access transistor and operating voltage to minimize the energy dissipation of the array while maintaining cell quality.
3) Exploration of architecture design space for L2 caches (shared and non-shared) considering read to write access ratio.

II. CELL METRICS AND OPERATIONS

The two important quality metrics for an MTJ are: 1) the TMR and 2) the switching current ($I_C$). At a particular operating temperature, both of these factors depend on the structure and material of the MTJ device. TMR is the readability metric and is defined as

$$TMR = \frac{R_H - R_L}{R_H} \quad (1)$$

where $R_H$ and $R_L$ denote the resistance offered by the MTJ in anti-parallel and parallel orientations [10]. A high TMR indicates a larger difference between resistances in the parallel and anti-parallel modes and facilitates reading. A high TMR is hence desirable.

However, for an STTRAM cell, equivalent quality metrics need to consider the effect of the access device. In this section, we analyze the effect of access device on these qualities and define the regions in the array operating voltage ($V_{WL}$) and access device width ($W$) plane that will satisfy the cell quality requirements. In this analysis, we assume that, the word-line voltage during read/write operations and bit-line/source-line voltage (for parallel/anti parallel writes) during write operations are same (defined collectively as the array operating voltage, $V_{WL}$). The bit-line voltage during read operation ($V_{read}$) is different. This ensures that, only two voltage levels are required for the array.

A. Read Operation—Cell-TMR

During read the bitline is precharged with a small voltage ($V_{read}$) and current flows in the direction of bitline (BL) to source line (SL) (see Fig. 1). The errors in read operation are classified under two heads: 1) false read and 2) read disturb. False read occurs when the output sensed in different from the state of the MTJ. Read disturb occurs when in trying to read the state of the MTJ is flipped. In this subsection, we study the read operation metrics, their relation to the read errors and validate our analysis with simulation results from TSMC 180 nm.

False read is related to the quality of read operation. The quality of read operation of the cell depends on the difference between the current flowing through the cell while reading “1” (i.e., anti-parallel state) and reading “0”, (i.e., parallel state). The higher this difference, the more robust is the circuit against false reads. The cell TMR (CTMR) can be defined as

$$CTMR = \frac{I_{\text{read}0} - I_{\text{read}1}}{I_{\text{read}0}} \quad (2)$$

where $I_{\text{read}0}$ and $I_{\text{read}1}$ are the read currents for the parallel or “0” and anti-parallel or “1” cases, respectively. For a given access transistor, since the resistance in the anti-parallel (“1”) state is higher than the resistance in the parallel state $I_{\text{read}0} > I_{\text{read}1}$. From simple circuit analysis (2) can be interpreted as

$$CTMR = \frac{R_H - R_L}{R_H + \frac{1}{I_{\text{read}}}(V_{WL} - V_{th})} \quad (3)$$

where $R_H$ and $R_L$ are the resistances of the MTJ in the anti-parallel and parallel states; $W$ is the width of the access device; $V_{th}$ is the threshold voltage of the access device; $V_{WL}$ is the
word-line voltage (i.e., array operating voltage). For (3), the device length is incorporated in the constant $k$ (assuming minimum device length for the technology). It is important to note that $R_H$ and $R_L$ actually vary with the voltage across MTJ for considerable voltage swing. In our case the voltage swing across MTJ permits variation in $R_H \sim 2\%-5\%$. Hence it is logical to assume constant $R_H$ and $R_L$ values. In order to distinguish the currents for reading “0” and “1” the CTMR needs to satisfy a minimum bound (say, $\eta_{CTMR}$) governed primarily by the robustness of the sensing circuit. From (3), we can obtain the region in the $V_{WL} - W$ plane that will satisfy this minimum CTMR requirement

$$W(V_{WL} - V_{th}) \geq \frac{1}{k(R_H - R_L)} \left(\frac{1}{\eta_{CTMR}} - \frac{1}{TMR_0}\right)^{-1}$$

(4)

where TMR$_0$ represents the TMR of the MTJ given by (1) and is a property of the MTJ. From (3) we can clearly observe that, a higher word-line voltage and larger access device help to improve CTMR. Fig. 2 shows the effects of $W$ and $V_{WL}$ on CTMR considering TMR value obtained from using $R_H = 4.5$ K, $R_L = 2$ K at 180-nm technology.

The read disturb occurs when the read current is higher than the MTJ switching current which cause the cell data to flip during reading. We observe that flipping of a “0” during read is not a possibility. This is because the read current flows from BL to SL. Hence read disturb is limited to flipping of a “1”. The read current needs to be sufficiently smaller than the write switching current for the MTJ to reduce the probability of the read disturb (i.e., switching of MTJ state while reading it). On the other hand, a minimum value of read-current (say, $I_{min}$) is required to ensure that it can be sensed properly even under noise in the sensor. We consider these effects using the following criteria:

$$I_{min} < I_{read1} = \frac{V_{read} \left(V_{WL} - V_{th}\right)}{R_H + \frac{1}{\eta_{RM}}(V_{WL} - V_{th})} < \eta_{RM}I_C$$

(5)

where $\eta_{RM}$ is the minimum ratio between switching current and read current. The ratio $\eta_{RM}$ quantifies a design margin for read such that even under variation (which might increase $I_{read1}$ beyond its expected value) read disturbs do not occur. Hence, we refer to $\eta_{RM}$ as the read margin. Equation (5) helps to determine the $V_{read}$ voltage that should be used while reading a cell.

B. Write Operation—Cell-Switching Current

For an MTJ to switch the current through the MTJ must be greater than a critical current $I_C$ and a higher switching current is required for faster switching. Given the MTJ switching current, the cell switching current ($I_{cell}$) strongly depends on the access device and the direction of switching. During this operation $V_{WL}$ and $V_{BL}/V_{SL}$ are tied to the same voltage (depending on whether it is parallel/anti-parallel write).

1) Parallel to Parallel Switching—Write “0”: During anti-parallel to parallel switching, the current flows from bit-line to source-line. During this operation, the MTJ acts as a resistive load to the transistor. Hence, $V_{ds}$ of the access device is low and the device remains in the linear region. Further, during this operation, the initial cell resistance is $R_{AP}$ or $R_H$. We obtain the region in $V_{WL} - W$ plane that allows $I_{cell} > I_C$.

2) Parallel to Anti-Parallel Switching Write “1”: During parallel to anti-parallel switching, the current flows from source-line to bit-line. The MTJ acts as a resistor at the source of the transistor thereby reducing its effective $V_{gs}$. The transistor remains in the saturation region (as $V_g = V_D = V_{WL}$). Further, during this operation, the initial cell resistance is $R_P$ or $R_L$. We can obtain the region in the $V_{WL} - W$ plane that will allow $I_{cell} > I_C$ for writing “1”.

Fig. 4(a) and (b) shows the current through the cell for different $W$ and $V_{WL}$ for write “1” and write “0” conditions. Note, for write “0”, increasing the width beyond a certain point does not have strong impact on as the device acts as a resistor and current is limited by $V_{WL}/R_H$. But, for write “1” increasing $W$ always increases the current as the device is in saturation. For reasonable voltage ranges, for a given $V_{WL}$, the minimum device width required for write “1” is larger. This is because the increase in the source voltage reduces the effective $V_{gs}$ of the transistor. However, for small $V_{WL}$ (i.e., when $V_{WL}/R_H > I_C$) the higher width is required for write “0”. The operating region in the $V_{WL} - W$ plane to ensure a required cell switching current can be obtained by simultaneously considering the two write operations [see Fig. 4(c)]. Hence, for a given $V_{WL}$, the required device width is determined by the higher of widths required for write “1” and write “0”.

C. Design Space for Read and Write Operations

Proper choice of $V_{WL}$ and $W$ are required to simultaneously satisfy both the cell TMR and switching current requirements. Fig. 5(a) shows the acceptable regions in $V_{WL} - W$ plane for a given $\eta_{CTMR} (= 0.5)$. Essentially, CTMR target provides a lower bound on the allowable $V_{WL}$ and $W$ values. From Fig. 2 we observe that a higher CTMR targets requires a higher $V_{WL}$ and/or $W$ values. Hence, a higher CTMR target reduces the acceptable $V_{WL} - W$ plane as illustrated in Fig. 5(b). Fig. 5(a)
also shows the acceptable $V_{WL} - W$ plane satisfying a critical current target ($I_C \approx 0.3$ mA). The device width for a given $V_{WL}$ is determined considering the method discussed earlier [i.e., Fig. 4(c)].

As expected a higher critical current requires a higher $V_{WL}$ and/or $W$ value and hence reduces the acceptable $V_{WL} - W$ region. This is illustrated in Fig. 5(c). In summary, the $V_{WL}$ and $W$ values for a STTRAM cell need to be chosen to ensure that CTMR and switching current requirements are satisfied. A trivial solution is to increase both $W$ and $V_{WL}$ until read margin requirements are violated. But that will significantly increase the energy-dissipation of the array. Hence, the cell design (i.e., $V_{WL}$ and $W$ choice) need to consider their effect on the array energy. To analyze this effect in the next section we present the overall array energy model.

III. ENERGY DISSIPATION OF STTRAM ARRAY

The energy dissipation of the array is contributed by the total energy dissipation during read and write operation. Note, since STTRAM is a non-volatile memory all the voltage levels can be at 0 for a standby array. Hence, standby energy dissipation can be neglected. Let us now consider the energy dissipation during write and read operation for an array with $N_{\text{row}}$ number of rows and $N_{\text{col}}$ number of columns.

A. Dynamic Array Model for Energy Estimation

The dynamic energy of the array depends on the bitline and wordline capacitances. To compute these capacitances, we need to construct the dynamic model of a column while switching. The dynamic model of a STTRAM cell consists of per cell bitline ($C_{BL}$), the word-line metal capacitance ($C_{WL}$) and parasitic capacitance at the intermediate node ($C_{\text{node}}$) as shown in Fig. 6. $C_{\text{node}}$ is contributed by the overlap and junction capacitance of the transistor as well as interconnect capacitance due to the connection of MTJ and transistor. The parasitic capacitance also exists at the node connected to the source-line and increases linearly with an increase in the width of the cell transistor. The MTJ structure also consists of two ferromagnetic layers separated by a thin dielectric layer and hence acts as a capacitance between bit line and the intermediate node. If we consider the bit line is switching from 0 to $V_{WL}$, intermediate nodes of all the cells (selected and unselected) will be charged to $V_{WL}$. Hence, the total energy dissipated during the bit line switching need to consider intermediate capacitance of all the nodes. Similarly for source-line, source parasitic capacitance ($C_{\text{source}}$) of the all the cells connected to the source-line need to be considered. We evaluate the per cell bit line (i.e., $C_{BL}$) and word line (i.e., $C_{WL}$) capacitance from layout considerations (see Fig. 7). We estimate the per cell bit line length $L_{BL}$ and per cell word line length $L_{WL}$. Bit line capacitance is per cell is $C_{BL} = L_{BL} \times C_0$ and word line capacitance per cell is $C_{WL} = L_{WL} \times C_0$, where $L_{WL} = W + L$, $L_{BL} = 7L$. Here $W$, $L$ and $C_0$ are the width, length and capacitance of the metal line per unit length (taken to be 0.2 fF/um). Based on this model the total switching capacitance for bit-line ($C_{BL-\text{switch}}$), source-line ($C_{SL-\text{switch}}$) and word-line ($C_{WL-\text{switch}}$) are estimated as

$$C_{BL-\text{switch}} = N_{\text{row}} \left( \frac{L_{BL} C_0 + C_{\text{node}}}{C_{BL}} \right) = N_{\text{row}} \left( 7LC_0 + C_{\text{node}} \right)$$

$$C_{SL-\text{switch}} = N_{\text{row}} \left( \frac{L_{BL} C_0 + C_{\text{source}}}{C_{BL}} \right) = N_{\text{row}} \left( 7LC_0 + C_{\text{source}} \right)$$

$$C_{WL-\text{switch}} = N_{\text{col}} \left( \frac{L_{WL} C_0 + C_{\text{gate}}}{C_{WL}} \right) = N_{\text{col}} \left( (W + L) C_0 + C_g \right)$$

(6)

where $N_{\text{row}}$ is the number of rows per column; $N_{\text{col}}$ is the number of columns per row; $C_g$ is the gate capacitance of the cell transistor which depends linearly on the cell width and $W$ is the cell transistor width. The array consists of $N_{\text{row}} \times N_{\text{col}}$ number of individual cells.

B. Write Energy of the Array

The write energy of the array is contributed by three components, namely: 1) bit-line switching energy ($E_{BL}$); 2) wordline...
switching energy ($E_{WL}$); and 3) energy dissipated in the cell due to the flow of write current ($E_{cell}$).

1) **Bitline Switching Energy**: We consider the worst case switching of the bitline and source lines. This occurs when a write “1” is followed by a write “0” or vice versa. Under the valid assumption that interconnect capacitance for source line and bit-line are similar, the total bit line switching energy is given by

$$E_{BL} = (C_{BL-switch} + C_{SL-switch}) V_{BL}^2 = N_{row} \left(2L_{BL}C_0 + C_{node} + C_{source}\right) V_{WL}^2. \quad (7)$$

2) **Wordline Switching Energy**: Considering the array dynamic model, the word-line switching energy is given by

$$E_{WL} = (C_{WL-switch}) V_{WL}^2 = N_{col} \left(L_{WL}C_0 + C_g\right) V_{WL}^2. \quad (8)$$

3) **Energy Due to Write Current**: During writing we need to ensure the current flowing through the cell is equal to (or higher than) the required (worst-case) MTJ switching current for both write “0” and write “1” operations. As already mentioned we consider the higher of the MTJ switching current for write “0” and write “1” as our target MTJ switching current ($I_C$). The average cell switching current ($I_{sw}$) for write energy estimation (see Fig. 8) is estimated as follows:

$$I_{sw} = 0.25 \left[ I_{sw1 \to 0} + I_{sw1 \to 1} + I_{sw0 \to 1} + I_{sw0 \to 0} \right]$$

$$E_{cell} = V_{WL} I_{sw} T_{write}. \quad (9)$$

4) **Array Write Energy**: For a memory array with word-size of $N_{bit}$ (i.e., $N_{bit}$ number of columns are selected in each access) the total array energy is given by

$$E_{write} = N_{bit} \left( E_{BL} + E_{cell} \right) + E_{WL}. \quad (10)$$

C. **Read Energy of the Array**

Following the previous discussion, we can also obtain the read energy of the array which is given by

$$E_{BL} = N_{bit} \left( C_{BL-switch} + C_{SL-switch} \right) V_{read}^2 + (C_{WL-switch}) V_{WL}^2$$

$$+ N_{bit} V_{read} 0.5 \left(I_{read0} + I_{read1}\right) T_{read}$$

$$= N_{bit} N_{row} \left(2L_{BL}C_0 + C_{node} + C_{source}\right) V_{read}^2$$

$$+ N_{col} \left( L_{WL}C_0 + C_g \right) V_{WL}^2$$

$$+ N_{bit} V_{read} 0.5 \left(I_{read0} + I_{read1}\right) T_{read}. \quad (11)$$

In order to compute the energy model we consider equal word-line turn on time for both read and write. From [6] for a switching current of about 300 uA we can expect a write pulse width of 10 ns ($T_{write} = 10$ ns). In this work we assume equal cycles for read and writes. However it is possible that writes are accomplished in multiple cycles. To capture that effect an extension of the method with proper read-write probability considerations and weights attached to $T_{read}$ and $T_{write}$ can be used.

D. **Leakage Energy**

The STTRAM leakage has zero standby leakage. However active leakage needs to be considered. This leakage arises from the unselected cells in the selected column while read/write operations (see Fig. 6). The final leakage model is given by

$$E_{leakage-write} = N_{bit} \left(N_{row} - 1 \right) V_{WL} T_{write} \times 0.25 \left[ I_{J_{leak0}}^{(0 \to 0)} + I_{J_{leak0}}^{(0 \to 1)} + I_{J_{leak1}}^{(1 \to 0)} + I_{J_{leak1}}^{(1 \to 1)} \right]$$

$$E_{leakage-read} = N_{bit} \left(N_{row} - 1 \right) V_{read} T_{read} \times 0.5 \left[ I_{J_{leak0}}^{(0 \to 0)} + I_{J_{leak1}}^{(1 \to 1)} \right] \quad (12)$$

where $I_{J_{leak}}^{(i \to j)}$ represents the leakage energy when writing $i$ (0 or 1) to the selected cell and unselected cell store $j$ (0 and 1) and $I_{J_{leak}}^{(0 \to 0, 1 \to 1)}$ is the read energy when the unselected cell store $j$ (0 or 1). The leakage is primarily dominated by the leakage of the unselected cell while writing “0” to the selected cell.

IV. **ENERGY-AWARE DESIGN SPACE EXPLORATION**

Let $I_{sw_{write0}}$ is the switching current while writing 0 to a cell storing 1; $I_{sw_{write1}}$ is the switching current while writing 1 to a cell storing 0; $I_C$ is the MTJ switching current; $\eta_{CTMR}$ is the minimum CTMR target; $TMR_0$ is the TMR of the MTJ device; $\eta_{RM}$ is the minimum read margin target (i.e., determines maximum read current value).
First, we create a lookup table (LUT) that provides the cell write current, read current and CTMR for \( V_{WL} \) and \( W \) values satisfying the constraint. Next, given a TMR \( TMR_0 \) and \( \eta_{CTMR} \), we obtain the design space (given by \( U_{TMR} \)) for \( V_{WL} \) and \( W \) to satisfy constraint (2) using the LUT as shown in the following:

\[
U_{TMR} \equiv \{(V_{WL}, W) : CTMR > \eta_{CTMR}\}.
\]

Next, given \( I_C \), we obtain the design space for \( V_{WL} \) and \( W \) (given by \( U_{IC} \)) to satisfy constraint (1) using the LUT as shown in the following:

\[
U_{IC} \equiv \{(V_{WL}, W) : I_{sw,\text{write}0} and I_{sw,\text{write}1} > I_C\}.
\]

Next, we obtain the design space (\( U_{rd,\text{distrb}} \)) that satisfies the read disturb condition using the LUT:

\[
U_{rd,\text{distrb}} \equiv \{(V_{WL}, W) : I_{\text{min}} < I_{\text{read}} < \eta_{RM}I_C\}.
\]

Next, we obtain the feasible design space (\( U_{rd,\text{sat}} \)), i.e., \( V_{WL} \) and \( W \) that satisfy both constraints as shown in the following:

\[
U_{rd,\text{sat}} \equiv \{(V_{WL}, W) : (V_{WL}, W) \in (U_{TMR} \cap U_{IC} \cap U_{rd,\text{distrb}})\}.
\]

Finally, we explore the \( V_{WL} - W \) space in \( U_{rd,\text{sat}} \) to obtain the minimum energy solution [using (6)–(11)] for \( V_{WL} \) and \( W \).

The transistor width and the word line voltage are thus fixed giving us an energy efficient solution for the array meeting the criteria of TMR and write current. The methodology is shown in Fig. 9. From the energy surface for the allowed design space for particular set of constraints (the acceptance bin population in Fig. 9), the minimum energy solution is chosen. The factor \( \eta_{RM} \) is assumed to be 0.5. An exact estimation of \( \eta_{RM} \) for a reliability target can be performed using the statistical modeling technique proposed in [11]. Next, we consider different switching current and CTMR targets to obtain the minimum energy condition. Since a tighter constraints (i.e., higher CTMR and higher switching current) results in higher values for \( V_{WL} \) and \( W \) (see Fig. 5), minimum achievable energy increases at tighter requirements (see Fig. 10).

### A. Performance Impact of the Proposed Methodology

The performance of the STTRAM cell is primarily determined by the MTJ switching time which depends on the write current target. Since the proposed energy optimization method uses write current as a constraint it does not impact the MTJ switching time. However, the optimal solution calls for a new combination of operating voltages and transistor widths. Corresponding to each such combination the word line length changes and consequently there is a change in word line capacitance.

The bit line length is not dependent on the transistor width and is not influenced by these changes. Hence, it is expected that the optimization of the energy will also impact the word-line delay and hence overall cell performance. For worst-case performance overhead estimation we consider the “1” to “0” (i.e., high resistance to low resistance transition) which requires a less write time for the same switching current as discussed in [6]. Fig. 11 shows that with increasing write current, the optimal transistor sizes increase, write pulse sizes decrease and the word line delay becomes a more significant part of the write time requirement. However even this is restricted to 6% which is not a significant contribution. For a read access requirement of 10 ns, word line delay is also a constant 6% of the access time across switching currents.
is the probability of writing to the STTRAM cache.

The total energy contribution of the array can be evaluated as

$$E_{\text{total}} = \alpha E_{\text{write}} + (1 - \alpha) E_{\text{read}}$$ (17)

where $\alpha$ is the probability of writing to the STTRAM cache. Fig. 12 shows that at very low write probabilities ($< 0.2$) read energy is more than write energy. However, beyond write probability of 0.4 write energy starts dominating by and large. This is due to the fact that the write current and the bit line voltages are larger than in the corresponding read case. Another interesting aspect is the total energy is $4 \times$ lower for read intensive operations compared to write intensive ones. However for programs in the same benchmarks the read-write ratio varies appreciably. Hence we do not consider optimization based on the read-write probability. However, there is a scope for minimizing the total energy if we can modify the read-write ratio in L2 caches using STTRAM. In that case, (17) can be treated as cost function to extend the proposed methodology to read/write biased applications. Hence a detailed study of the architectural methods of varying read-write probability becomes crucial.

B. Architectural Evaluation

We present our experimental setup for capturing LLC access behavior and computing the read/write statistics. Fig. 13 illustrates a high level representation of a two level cache hierarchy with a single processor core and the accesses that correspond to read/write behaviors. For our model we assume that the tag arrays of the caches do not use an STTRAM-based cell design and hence the energy from accesses to the tag array is not accounted for. All instruction read, data read, and data prefetch hits in the LLC correspond to cell read operations (18) while all instruction, prefetch, and data misses as well as data write hits correspond to cell write operations (19)

$$\text{Total}_{\text{reads}} = \text{IRH} + \text{DRH} + \text{IPH} + \text{DPH}$$

$$\text{Total}_{\text{writes}} = \text{IRM} + \text{IPM} + \text{DWH} + \text{DRM} + \text{DWM} + \text{DPM}$$

where IRH is the instruction read hit; IPH is the instruction prefetch hit; DRH is the data read hit; DWH is the data write hit; DPH is the data prefetch hit; DRM is the data read miss; DWM is the data write miss and DPM is the data prefetch miss.

Both expressions are multiplied by the corresponding array read and write energies for a 32 bit word. A write occurs when there is a write from a program perspective or from a write due to a miss in the cache. Hence the raw miss rate amplifies the number of write operations. A second parameter that can adversely impact the number of write operations is the cache prefetch policy. The result of a prefetch operation is a number of writes with the further impact that in some cases a prefetch may pollute the cache and be the cause for a cache miss, further increasing the number of writes. Traditionally hit rates are known to improve when the cache size is increased. With the use of STTRAM, which takes significantly less area than an SRAM equivalent, we have the option of realizing a much larger cache in the same area. Other design decisions or optimizations that reduce the number of write operations, even at the expense of increasing the raw number of read operations can prove to be beneficial when considering the energy consumption with an STTRAM-based cache.

C. Choice of LLC Configuration

We first analyze the read-write statistics of a set of benchmark applications. Fig. 13 shows the variation in read/write statistics for four SPEC CPU 2006 floating point applications and two SPEC CPU 2006 integer benchmark sets. All simulations were executed for 3 billion instructions (1 billion warm up period) using L2 cache sizes varying from 512KB to 8MB. The simulation environment for this section involved a multicore version of zesto [15] compiled with gcc 4.1. The L2 is modeled as a 16 way set associative cache of varying sizes with an 8 cycle latency of access. We note that the relative probabilities of read versus write operations are fairly stable for the MILC and the cactus ADM benchmarks and some variability can be seen for the bzip2 benchmarks reducing the write probability by approximately 0.1. From Fig. 14 it can be seen that a variation of 0.1 in the write probability is a reduction of approximately 10 pj/access. As an alternative to changing the cache size, we may choose between a shared versus private L2 organization. Shared caches are typically much larger in size and consume a significant percentage of the total energy of the core. The degree of sharing is the primary determinant in the choice of the private versus shared caches. However, a high degree of sharing enables a single copy to be maintained in the LLC rather than being replicated in multiple private LLCs increasing the number of reads per cell with the consequent energy cost. In designs with private caches sharing however can cause increased coherence.

V. DEPENDENCE OF ENERGY ON READ-WRITE RATIO: AN ARCHITECTURAL STUDY

A. Energy Variation With Read-Write Ratio

In this section, we consider the effect that different read and write ratios on the energy consumption of the STTRAM array. The total energy contribution of the array can be evaluated as

$$E_{\text{total}} = \alpha E_{\text{write}} + (1 - \alpha) E_{\text{read}}$$ (17)

where $\alpha$ is the probability of writing to the STTRAM cache. Fig. 12 shows that at very low write probabilities ($< 0.2$) read energy is more than write energy. However, beyond write probability of 0.4 write energy starts dominating by and large. This is due to the fact that the write current and the bit line voltages are larger than in the corresponding read case. Another interesting aspect is the total energy is $4 \times$ lower for read intensive operations compared to write intensive ones. However for programs in the same benchmarks the read-write ratio varies appreciably. Hence we do not consider optimization based on the read-write probability. However, there is a scope for minimizing the total energy if we can modify the read-write ratio in L2 caches using STTRAM. In that case, (17) can be treated as cost function to extend the proposed methodology to read/write biased applications. Hence a detailed study of the architectural methods of varying read-write probability becomes crucial.

B. Architectural Evaluation

We present our experimental setup for capturing LLC access behavior and computing the read/write statistics. Fig. 13 illustrates a high level representation of a two level cache hierarchy with a single processor core and the accesses that correspond to read/write behaviors. For our model we assume that the tag arrays of the caches do not use an STTRAM-based cell design and hence the energy from accesses to the tag array is not accounted for. All instruction read, data read, and data prefetch hits in the LLC correspond to cell read operations (18) while all instruction, prefetch, and data misses as well as data write hits correspond to cell write operations (19)

$$\text{Total}_{\text{reads}} = \text{IRH} + \text{DRH} + \text{IPH} + \text{DPH}$$

$$\text{Total}_{\text{writes}} = \text{IRM} + \text{IPM} + \text{DWH} + \text{DRM} + \text{DWM} + \text{DPM}$$ (19)

where IRH is the instruction read hit; IPH is the instruction prefetch hit; DRH is the data read hit; DWH is the data write hit; DPH is the data prefetch hit; DRM is the data read miss; DWM is the data write miss and DPM is the data prefetch miss. Both expressions are multiplied by the corresponding array read and write energies for a 32 bit word. A write occurs when there is a write from a program perspective or from a write due to a miss in the cache. Hence the raw miss rate amplifies the number of write operations. A second parameter that can adversely impact the number of write operations is the cache prefetch policy. The result of a prefetch operation is a number of writes with the further impact that in some cases a prefetch may pollute the cache and be the cause for a cache miss, further increasing the number of writes. Traditionally hit rates are known to improve when the cache size is increased. With the use of STTRAM, which takes significantly less area than an SRAM equivalent, we have the option of realizing a much larger cache in the same area. Other design decisions or optimizations that reduce the number of write operations, even at the expense of increasing the raw number of read operations can prove to be beneficial when considering the energy consumption with an STTRAM-based cache.

C. Choice of LLC Configuration

We first analyze the read-write statistics of a set of benchmark applications. Fig. 13 shows the variation in read/write statistics for four SPEC CPU 2006 floating point applications and two SPEC CPU 2006 integer benchmark sets. All simulations were executed for 3 billion instructions (1 billion warm up period) using L2 cache sizes varying from 512KB to 8MB. The simulation environment for this section involved a multicore version of zesto [15] compiled with gcc 4.1. The L2 is modeled as a 16 way set associative cache of varying sizes with an 8 cycle latency of access. We note that the relative probabilities of read versus write operations are fairly stable for the MILC and the cactus ADM benchmarks and some variability can be seen for the bzip2 benchmarks reducing the write probability by approximately 0.1. From Fig. 14 it can be seen that a variation of 0.1 in the write probability is a reduction of approximately 10 pj/access. As an alternative to changing the cache size, we may choose between a shared versus private L2 organization. Shared caches are typically much larger in size and consume a significant percentage of the total energy of the core. The degree of sharing is the primary determinant in the choice of the private versus shared caches. However, a high degree of sharing enables a single copy to be maintained in the LLC rather than being replicated in multiple private LLCs increasing the number of reads per cell with the consequent energy cost. In designs with private caches sharing however can cause increased coherence.
A single cache line is 256 bits wide and 512 kB for the private caches). In simulating private caches, embedded applications are simulated on 2 MB shared caches. The first four bars denote the write probability and energy/access from a single private cache of 1MB and the last bar indicates the energy/access from a single shared cache of 4 MB (embedded applications are simulated on 2 MB shared caches and 512 kB for the private caches). In simulating private caches, each benchmark group used a private L2 of one-fourth the size. The first set comprises of phylip, clustal, tigr, and fasta from the biobench benchmark suite. The second two sets are from the SPEC CPU2006 benchmark suite. 436.CactusADM, 447.dealII, 437.gromacs, and 470.lbm from the floating point applications and 473.astar, 401.bzip2, 400.perlbench, 403.gcc from the integer applications. The last two sets are applications from MIBench [16] and MediaBench. In both the private and shared cases, each core is executing a single program from the set. In all cases the LLCs are designed using the write statistics as detailed earlier. One write intensive benchmark can be seen from the plots in Fig. 14(a.ii) is basic math which is a benchmark from the automation and industrial control category of MIBench and performs simple mathematical calculations that often don’t have dedicated hardware support in embedded processors. Fig. 14(b) further shows how the absolute number of writes is reduced with the use of a shared cache for the lbm benchmark. Reducing the number of writes will further reduce the energy per access. For write intensive workloads the saving in terms of energy may be significant as can be seen for the lbm workload in Fig. 14(a.i).

While STTRAM presents opportunities for energy minimization it affects several traditional cache optimizations. For example, pre-fetching is a common optimization that has the negative consequence of amplifying the higher write energy of STTRAMs. Consequently, pre-fetch is no longer an obviously desirable feature. Turning off pre-fetch optimizations can increase the miss rate and therefore the execution time. However, this can be compensated for by increasing STTRAM cache size without a net increase in the number of write operations. Finally the asymmetric read-write energy costs motivate compiler and architectural optimizations that will minimize write operations even at the expense of increased read operations. For example, replacement policies may bias the choice of line to evict based on the probability of this line be written rather than read.

VI. CONCLUSION

In this paper, we analyze the energy dissipation in an STTRAM array and develop a methodology for co-design of the cell transistor width and supply voltage to minimize the array energy dissipation. Our analysis shows, for a given MTJ, the quality of an STTRAM cell strongly depends on the cell transistor. Co-design of the transistor width and supply voltage is necessary to minimize the energy dissipation while maintaining requirements for successful read and write operations. We show that for STTRAM write energy dominates over read energy and conclude that minimizing write access to the array could help to reduce the total energy. Motivated by the above observation, we explore the use of program statistics to drive the design of memory cells to provide a more energy efficient LLC.

ACKNOWLEDGMENT

The authors would like to thank Dr. A. Raychowdhury of Circuit Research Labs, Intel Corporation, for many helpful discussions. The authors would like to acknowledge Prof. G. H. Loh and the STING Research Group for the base simulation framework (zesto) that has been used in all architectural simulations in this work.

REFERENCES


Mitchelle Rasquinha received the Bachelor’s degree in electrical engineering from the University of Texas at Austin, in 1984. Upon graduation, she joined Honeywell’s Systems and Research Center, Minneapolis, MN, where she worked as a Senior, and then Principal Research Scientist from 1984 to 1989, served as an Adjunct Faculty and taught in the Department of Electrical Engineering at the University of Minnesota, and served on Honeywell’s Program Technical Advisory Committee to the Microelectronics Technology Corporation (MSS). She joined the ECE faculty at the Georgia Institute of Technology, Atlanta, in 1989, where she is currently a Joseph M. Pettit Professor of Computer Engineering. She has served as the PI and Co-PI on sponsored projects in the areas of reconfigurable computing, high performance interconnection networks, resource management for parallel architectures, and modeling and simulation of parallel architectures. She is currently a Co-Director of the NSF Industry University Research Center on Experimental Computer Systems (CERC) and is a member of the Research Advisory Group to the HyperTransport Consortium. She is the author of VHDL Starters Guide, 2nd ed. (Prentice-Hall, 2004), VHDL: From Simulation to Synthesis (Prentice-Hall, 2000), and co-author with J. Duato and L. Ni, of Interconnection Networks: An Engineering Approach (Morgan Kaufman, 2003).

Dr. Yalamanchili has served as a Distinguished Visitor of the IEEE and Associate Editor for the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED PROCESSING and IEEE TRANSACTIONS ON COMPUTERS. He contributes professionally through service on conference and workshop program committees in the areas of high performance computing systems, interconnect networks, and computer architecture.

Sudhakar Yalamanchili (SM’91) received the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, in 1984. Upon graduation, he joined Honeywell’s Systems and Research Center, Minneapolis, MN, where he worked as a Senior, and then Principal Research Scientist from 1984 to 1989, served as an Adjunct Faculty and taught in the Department of Electrical Engineering at the University of Minnesota, and served on Honeywell’s Program Technical Advisory Committee to the Microelectronics Technology Corporation (MSS). He joined the ECE faculty at the Georgia Institute of Technology, Atlanta, in 1989, where he is currently a Joseph M. Pettit Professor of Computer Engineering. He has served as the PI and Co-PI on sponsored projects in the areas of reconconfigurable computing, high performance interconnection networks, resource management for parallel architectures, and modeling and simulation of parallel architectures. He is currently a Co-Director of the NSF Industry University Research Center on Experimental Computer Systems (CERC) and is a member of the Research Advisory Group to the HyperTransport Consortium. He is the author of VHDL Starters Guide, 2nd ed. (Prentice-Hall, 2004), VHDL: From Simulation to Synthesis (Prentice-Hall, 2000), and co-author with J. Duato and L. Ni, of Interconnection Networks: An Engineering Approach (Morgan Kaufman, 2003).

Dr. Yalamanchili has served as a Distinguished Visitor of the IEEE and Associate Editor for the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED PROCESSING and IEEE TRANSACTIONS ON COMPUTERS. He contributes professionally through service on conference and workshop program committees in the areas of high performance computing systems, interconnect networks, and computer architecture.

Saibal Mukhopadhyay (S’99–M’07) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 2000 and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2006.

He is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, as an Assistant Professor. Prior to joining Georgia Tech, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member and worked on high-performance circuit design and technology-circuit co-design focusing primarily on Static Random Access Memories. His research interests include analysis and design of low-power and robust circuits in nanometer technologies. He has (co-)authored over 90 papers in refereed journals and conferences.

Dr. Mukhopadhyay was a recipient of the IBM Faculty Partnership Award for 2009, SRC Inventor Recognition Award 2009, SRC Technical Excellence Award in 2005, IBM PhD Fellowship Award for 2004–2005, the Best in Session Award at 2005 SRC TECNCON, Best Paper Awards at 2003 IEEE Nano and 2004 International Conference on Computer Design.

Subho Chatterjee (S’08) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 2004 and the M.S. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 2006. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA.

His research interests include non-volatile memories, low power and variation tolerant design techniques, thermal reliability analysis, and 3-D integrated circuit design.