Architecting SOT-RAM Based GPU Register File

Sparsh Mittal[⋆], Rajendra Bishnoi, Fabian Oboril, Haonan Wang[†], Mehdi Tahoori, Adwait Jog[†], Jeffrey S. Vetter[§]
[⋆]*IIT Hyderabad, India; KIT, Germany;*[†]*College of William and Mary, USA;* [§]*ORNL, USA*
sparsh@iith.ac.in, {rajendra.bishnoi,fabian.oboril,mehdi.tahoori}@kit.edu, {hwang07,adwait}@cs.wm.edu, vetter@ornl.gov

***Abstract*—With increase in GPU register file (RF) size, its power consumption has also increased . Since RF exists at the highest level in cache hierarchy, designing it with memories with high write latency/energy (e.g., spin transfer torque RAM) can lead to large energy loss. In this paper, we present an spin orbit torque RAM (SOT-RAM) based RF design which provides higher energy efficiency than SRAM and STT-RAM RFs while maintaining performance same as that of SRAM RF. To further improve energy efficiency of SOT-RAM based RF, we propose avoiding redundant bit-writes to RF. Compared to SRAM RF, SOT-RAM RF saves 18.6% energy and by using our technique for avoiding redundant writes, the energy saving can be increased to 44.3%, without harming performance.**

I. Introduction

To support their highly multi-threaded architecture and provide efficient context switching between threads, GPUs (graphics processing units) use very large register file (RF). The capacity of GPU RF is much larger than that of CPU RF and is even larger than that of L2 cache on GPU. This fact is evident from Table I and with increasing demand for throughput, RF size per compute unit (or streaming multiprocessor, SM) and per chip is likely to grow even further. Traditionally, RF is designed with SRAM cells which consumes large leakage power [1]. This, along with the large size of GPU RF makes RF power consumption a significant fraction of overall power consumption of GPU, for example, RF leakage and dynamic power consumption contribute nearly 17% and 44% (respectively) of the total core power in GTX 470 GPU [2].

Table I
L2 and RF size in KB in recent NVIDIA GPUs [3]

	G80	GT200	GF100	GK110	GM204	GP100
Architecture	Tesla	Tesla	Fermi	Kepler	Maxwell	Pascal
Compute Capability	1.0	1.3	2	3.5	5.2	6.0
Number of SMs	16	30	16	15	16	56
L2 size	N/A	N/A	768	1536	2048	4096
RF size per SM	32	64	128	256	256	256
Total RF size	512	1920	2048	3840	4096	14336

To manage GPU RF power consumption, researchers have explored emerging memory technologies for designing RF, for example, eDRAM [4, 5], STT-RAM (spin transfer torque RAM) [2, 6], hybrid SRAM-STTRAM design [7] and DWM (domain wall memory) [8]. However, these memory technologies have their own limitations, e.g., eDRAM requires frequent refresh operations due to very small (e.g., $40\mu s$) retention period, whereas DWM access requires shift operations which lead to large latency and energy penalty. For STT-RAM, high write latency/energy along with 'read-disturbance error' (RDE) issue present a crucial bottleneck in its use for designing fast memories, e.g., L1 cache and RF. It is clear that the state-of-art in RF design calls for novel approaches for meeting area and power budgets and ensuring adoption of GPU in today's power-constrained computing world.

In this paper, we propose an SOT-RAM based RF architecture for improving energy efficiency of GPU RF. SOT-RAM is the state-of-the-art spintronic technology which has very low switching delay and energy due to the absence of incubation delay in the switching process [9, 10]. Moreover, unlike in STT-RAM, the write current in SOT-RAM does not flow through the MTJ (magnetic tunnel junction) stacks and hence, SOT-RAM shows very high endurance and almost no RDE. SOT-RAM write latency is much lower than that in STT-RAM since there is no incubation delay to stimulate the switch of the magnetization [9]. Further, SOT-RAM has isolated read and write current paths, which allows independent optimization of read/write currents and latencies [11], whereas read/write paths are shared in STT-RAM. Hence, SOT-RAM shows close to SRAM read/write performance with much lower leakage energy consumption. We perform a comprehensive comparison of GPU RF designed with SRAM, STT-RAM and SOT-RAM, in terms of performance and energy efficiency. Based on this, we demonstrate that SOT-RAM is the best suited technology for designing GPU RF.

To further reduce the energy consumption of SOT-RAM based RF, we utilize the observation that a large fraction of bit-level writes in RF are redundant, i.e., they write the same value as originally stored. In fact, as shown in Figure 1, ~80% of bit-writes are redundant (refer Section IV for details on evaluation platform). We use two circuit-level techniques, namely *unnecessary write termination* (UWT) and *unnecessary write avoidance* (UWA) proposed in [11], to avoid such redundant writes to save RF dynamic energy. We design additional circuitry, e.g., comparators, control mechanism to integrate UWx (UWA or UWT) techniques in GPU RF framework, and also perform GPU system-

Support for this work was provided by SERB ECR/2017/000622 project (India), DOE ASCR, NSF grant CCF 1657336 and a start-up grant from College of William and Mary (USA).

2159-3477/17 $31.00 © 2017 IEEE
DOI 10.1109/ISVLSI.2017.17

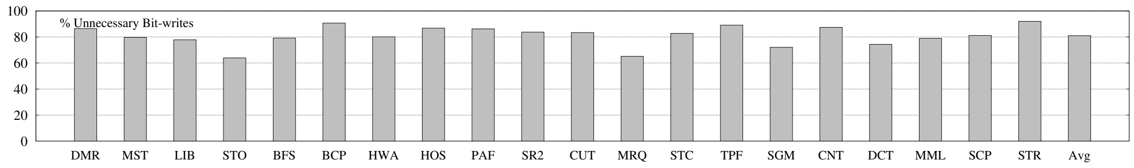

Figure 1. Percentage of unnecessary bit-writes in 128KB RF

level evaluation. Overall, compared to conventional SRAM RF, we propose saving leakage energy by using the non-volatile SOT-RAM memory and dynamic energy by avoiding redundant bit-writes. *To the best of our knowledge, we are the first to propose SOT-RAM based GPU RF architecture and circuit-level policies to manage its power consumption.*

Our contributions in this paper are as follows:

1. We propose a SOT-RAM based GPU RF design and show that it provides better performance than STT-RAM RF and better energy efficiency than SRAM and STT-RAM based RFs.

2. We leverage data access characteristics of GPU RF to further reduce the energy consumption of SOT-RAM based RF by employing redundant write avoidance/termination techniques.

3. Our microarchitectural simulations have shown that compared to SRAM RF, SOT-RAM RF saves 18.6% energy without harming performance, whereas STT-RAM saves only 0.8% energy with a relative performance of 0.99×. Further, on using UWT and UWA techniques with SOT-RAM RF, the energy saving can be increased to 41.0% and 44.3%, respectively. Further, SOT-RAM RF and UWx techniques provide large energy savings for different RF sizes, GPU frequencies and scheduling policies.

II. Background and Related work

A. SOT-RAM

SOT-RAM has several attractive features which make it suitable for designing RF. It has near-zero leakage power consumption since only peripheral circuitry contributes to the leakage. It has high density and write-endurance, is compatible with CMOS and immune to the radiation-induced soft-errors. Also, SOT-RAM promises ultra-fast switching speeds in the sub-ns region. As the metal electrode can be manufactured in an ultra-thin manner, the current density in SOT-RAM can be increased compared to STT-RAM without impairing endurance, and higher current density results in faster switching. However, SOT-RAM requires an additional terminal compared to STT-RAM for separating the read and write paths. Consequently, the area footprint of SOT-RAM cells is slightly larger than that of STT-RAM cells.

B. Related work

Li et al. [6] propose STT-RAM based RF design and use two write-buffers for coalescing writes and allowing simultaneous read/write accesses. Some other techniques also use write-back buffers to coalesce and reduce the writes to NVM [2, 6, 8]. Tan et al. [7] observe lifetime of register values using compiler and map long-lived and short-lived registers to STT-RAM and SRAM, respectively, to reduce soft-errors in SRAM. However, use of SRAM as a buffer or in hybrid SRAM-STTRAM RF leads to fabrication challenges due to its differences with STT-RAM. Also, the leakage power of SRAM may partially offset the energy saving due to use of STT-RAM. Further, use of a buffer and data-migration policies complicate the design and operation of RF.

For STT-RAM based RF, Goswami et al. [2] propose updating only the changed register arrays, instead of writing the whole register. Since using a per-bit write enable signal incurs area overhead due to extra circuitry, they propose redesigning the RF architecture such that N-bit register word is split into M arrays of K bits/array, such that $N = M \times K$. Then, even for a single bit modification in the array, all the K-bits of an array are written. They propose using K= 8 or 4. However, their technique still leads to redundant writes. By comparison, our technique does not require array-level redesign of RF and reduces writes at bit-level.

To address RDE in STT-RAM, after each read operation, a write needs to be performed to restore the data [**?**]. To reduce restore requirement, Zhang et al. [13] utilize compiler to identify last read and avoid restore operation for this. They also use an SRAM buffer to store read-intensive data. However, the first approach requires modifying the binary whereas the second approach complicates the read operations. Clearly, given the performance-criticality of read operations, RDE issue presents severe challenges in use of STT-RAM.

III. An SOT-RAM based GPU RF architecture

A. Motivation

Our work is guided by two observations and opportunities:

1. SRAM limitations and GPU characteristics: Figure 2 shows various components of energy consumption of SRAM RF (the experimentation platform is detailed in Section IV-A). In *SRAM-based GPU RF*, leakage energy contributes 25% of the total energy consumption, whereas in *SRAM-based last level caches (LLCs) in CPUs*, leakage energy contributes nearly 90% of the total energy [14]. Since RF is accessed more frequently than LLC, the contribution of leakage is smaller, however, leakage energy is still high for two reasons. First, most applications cannot make use of a large number of available registers and second, some other resource (e.g., number of threads, shared memory,

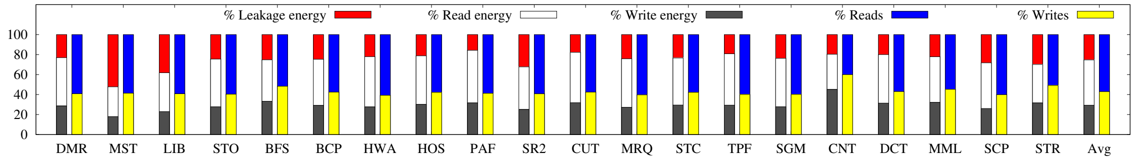

Figure 2. Percentage distribution of energy (leakage, read and write) and accesses (read and write) for a 128KB SRAM RF

limit on maximum number of warps, etc.) may restrict the total number of simultaneously running warps. Hence, many registers remain idle and dissipate energy. This motivates use of a low-leakage memory technology for designing RF. Further, given the performance-critical nature of GPUs and capacity requirement of GPU RF, the memory technology also needs to have low-latency.

Since memories such as SRAM, STT-RAM and eDRAM fail to meet one or more of these characteristics, we propose using SOT-RAM for designing GPU RF. Table II shows the parameters of SRAM, STT-RAM and SOT-RAM RFs obtained using NVSim for a 65nm technology node. We use perpendicular SOT-MRAM and the details of MTJ are provided in previous works [11, 15]. Clearly, the write latency and energy of SOT-RAM are lower than that of STT-RAM and are comparable to that of SRAM. Also, the leakage power of SOT-RAM is much lower than that of SRAM. Hence, SOT-RAM based RF can reduce the RF energy consumption compared to both SRAM and STT-RAM. Further, previous work has observed that the average time between consecutive accesses to RF ranges in hundreds of cycles [16]. Given this access characteristics of RF, use of SOT-RAM can be highly effective for enabling normally-off/instantly-on computing capability in RF. Additionally, the design parameters of SOT-RAM are improved with the increase of the RF capacity as depicted in the table. For instance, the total area of SOT-RAM for 256 KB RF is lower than that of the SRAM since with this RF capacity, the cell array area starts dominating in contrast to the periphery area for the contribution of the total area. Moreover, similar to RF area, the access latencies and energies are also improved with the increase of the RF capacity [10].

Table II
Parameters of SRAM, STT-RAM and SOT-RAM (area in mm^2, lat. = latency (in ns), energy in pJ, leakage in mW)

	128KB RF			256KB RF		
	SRAM	STT	SOT	SRAM	STT	SOT
Area	0.73	0.87	0.88	1.42	1.18	1.19
Read lat. (cycle)	1.15 (1)	1.07 (1)	1.06 (1)	1.66 (2)	1.11 (1)	1.1 (1)
Write lat. (cycle)	1.12 (1)	4.56 (4)	1.16 (1)	1.63 (2)	4.62 (4)	1.22 (1)
Read energy	404.7	340.66	317.06	504.85	378.67	355.24
Write energy	346.49	627.91	449.13	408.03	724.37	545.91
Leakage power	250.01	77.55	77.62	490.95	150.16	150.23

2. Temporal redundancy in RF writes: Unlike SRAM, the SOT-RAM bit-cells require a constant current value for a certain duration to switch their magnetization. Due to this, the write energy of SOT-RAM becomes higher than that of SRAM (refer Table II) and hence, write energy becomes a major contributor to the overall energy consumption in SOT-RAM RF. To reduce this, we leverage the observation that many successive bit-write operations to RF are redundant. This happens due to several reasons, e.g., use of same register for storing a variable or a constant in consecutive iterations of a loop or invocations of a function, small changes in value of a variable (e.g., $a \leftarrow a + 1$), operations on narrow values, etc. We exploit this phenomenon to avoid redundant bit-writes to RF for saving dynamic write energy.

B. Energy reduction techniques for SOT-RAM

In SOT-RAM, the *cell-level* read latency is nearly five times lower than the write latency (note that latency values in Table II are for *device-level* and not cell-level). Hence, by reading the contents of bit-cells at early stages of write operations, unnecessary (redundant) write operations can be avoided. Based on this, we employ two circuit-level techniques.

Unnecessary Write Avoidance (UWA): In this technique, a read operation is initiated before every write operation [11]. Once the content of each bit-cell is known, it is compared with the value to be written and then, unnecessary write operations are cancelled. In this way, the unnecessary write operations can be completely avoided at bit-cell level. Since the additional read operation is performed locally for each bit-cell, the addresses are already latched and hence, the latency and energy overheads due to this read operation and other control circuitry are negligible.

Unnecessary Write Termination (UWT): In this technique, a read operation is performed along with the write operation [11]. Here, the unique property of the SOT-RAM cell is exploited that it can perform read and write operations simultaneously. This is because, in SOT-RAM, the read and write currents flow through independent paths and hence, these currents can flow at the same time without affecting the functionality of each other. The circuit schematic to illustrate this simultaneous read and write behavior is shown in Figure 3. As shown in Figure 3, the read current flows from the sense amplifier through the transistor N3 and afterwards through the MTJ to the ground connection in the write block. Hence, depending on the ongoing write operation, the read current has to pass either through the source line or the write line. If only one access transistor controls the write operation, e.g., only N1 is used, the read current depends on the ongoing write operation, as on path has an additional resistance in form of the write access transistor. This, however, can disturb the read operation. To avoid this issue, the bit-cell architecture is modified by adding a transistor (N2) on the other write terminal as well, to balance the read current. Based on comparison of new and existing value, the write operation is terminated if unnecessary or continued if necessary.

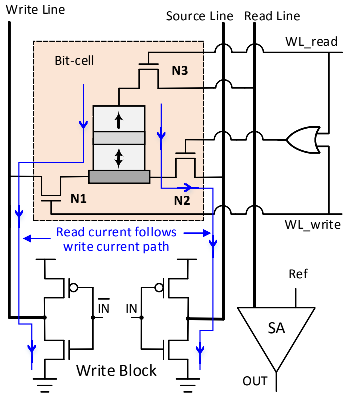

Figure 3. Circuit for simultaneous read-write used for UWT.

Since read is much faster than write, the concurrent read operation always reads the previous value at the very beginning stage of a write operation. The value read is compared with the value to be written, and the write operation is terminated if the value is same, otherwise the write operation is continued. On using UWT technique, for an unnecessary write operation, the write current flows for the duration of the read operation, still significant amount of energy can be saved after the write termination with no performance overhead. Compared to standard SOT-RAM design, implementation of UWT technique requires an additional routing row and column metal line.

Table III compares UWA and UWT techniques. Clearly, these techniques are complementary and allow the designer to exercise a tradeoff between performance and energy efficiency. UWT and UWA techniques are useful for high-performance and low-power applications, respectively.

Table III
Comparison of UWA and UWT

	UWA	UWT
Latency	Since read is performed before every write, there is a timing penalty equal to the bit-cell read latency	Read/write operations are performed simultaneously, thus there is no timing penalty.
Energy	Write is initiated only when read gets finished, hence, it achieves high energy savings.	Write operation continues till the read operation is finished. Hence, it saves less energy than UWA.

Comparison of UWx with previous works: . UWx techniques differ from previous works, e.g., early-write termination (EWT) [17] in the following ways:

(1) EWT technique is only applicable to STT-RAM (a two terminal device) and cannot work for SOT-RAM (a three terminal device). In STT-RAM, write current passes through the MTJ stack. Hence, any change during the write operation can be observed using a sensing mechanism as the resistance value is also altered immediately after the actual switching happens. However, in SOT-RAM the write current doesn't pass through the MTJ-stack, therefore the change in resistance cannot be observed during the write operation when the write current is flowing.

(2) EWT technique disturbs write operation since a load in the form of the sensing mechanism is placed on the write current path. The authors in [17] do not evaluate the impact of reduction of write current due to this effect. UWx techniques do not suffer from these effects since the reads and writes are completely isolated.

(3) EWT technique uses several additional circuit components for each bit e.g., several muxes, sense amplifier, conversion circuitries, etc. By comparison, UWx techniques require only one extra comparator since the remaining circuitries are already present in SOT-RAM design. Clearly, the area overhead of our approach is much lower than that of EWT.

For 128KB RF, the SOT-RAM write latencies for traditional scheme, UWT technique and UWA techniques are 1.156ns, 1.156ns and 1.165 ns, respectively. For a frequency of 700MHz, all these take 1 cycle only. Thus, due to the small capacity of RF, UWx techniques do not cause performance penalty, although we expect that for a larger capacity memory structure (e.g., a 4MB cache), UWA technique may add an extra cycle to the write latency. Further, UWx circuits do not affect the read latency/energy.

Both UWA and UWT add only a few XOR-gates to the memory design. The inputs of XOR-gates are connected to the read data and the incoming data to be written. The output of XOR-gate is connected to the write circuit, to deactivate it whenever both inputs are equal (i.e. XOR output is 0). Both techniques utilize existing read circuitry. The area overhead for a 128 KB RF for UWA and UWT is 0.9 % and 2.6 %, respectively. The area overhead for UWT is higher because it has to maintain the metal pitch in the layout for the additional column and row metal line. Since few additional control circuitries are involved, leakage contribution is also very low (~0.02 %). The write energy with UWx is computed as:

$$E_W = Bits_N \times E_N + Bits_U \times E_U + E_{overhead} \quad (1)$$

$Bits_N$ and $Bits_U$ show the number of necessary and unnecessary bit-writes (respectively). $Bits_N + Bits_U = 1024$ since each warp-register is 1024 bits (128B). E_N and E_U show the energy consumed in each necessary and unnecessary bit-write, respectively. $E_{overhead}$ shows the additional overhead energy. For 128KB RF bank, the energy values are as follows: $E_{overhead} = 90.15$ pJ for both UWA and UWT. For UWT, $E_N = 0.351$ pJ and $E_U = 0.052$ pJ, whereas for UWA, $E_N = 0.355$ pJ and $E_U = 0.004$ pJ. Thus, for unnecessary writes, UWA reduces write energy more than UWT and since the fraction of unnecessary writes is quite high (e.g., 80%), UWA saves more energy than UWT.

C. GPU RF architecture

Figure 4 shows the RF architecture assumed in this work [16, 18]. Under SIMT (single instruction multiple thread) execution model of GPUs, a group of multiple (e.g., 32 in CUDA and 64 in OpenCL) threads, called warp or wavefront, execute the same instruction. GPUs use a collector unit to avoid RF bank collisions among different warps. When all operands are read from RF, the instruction is issued to execution unit. After execution, the results are written-back to RF. The RF is divided into multiple uni-ported banks to reduce its access latency and power consumption. The RF has 128B width and it provides 32-bit operands to 32 threads of the warp.

When the input of an operation depends on the output of a previous operation (as indicated by scoreboard), RF write latency comes in critical path. An increase in write latency also delays future read operations since the arbiter prioritizes writes over reads issued to a same RF bank. However, in absence of such dependency, the write latency can be overlapped with read time or execution time. Since many instructions have two source operands and only one destination operand, the read latency has even higher impact on performance than the write latency. Clearly, since SOT-RAM is close to SRAM in read/write latency, it is more suitable for RF design than STT-RAM. Further, given the CMOS compatibility of SOT-RAM, it can easily replace SRAM. The circuit required for implementing UWT or UWA is simple, as explained before. UWx circuitry has negligible impact on write latency and no impact on read operations and hence, UWx techniques do not affect performance.

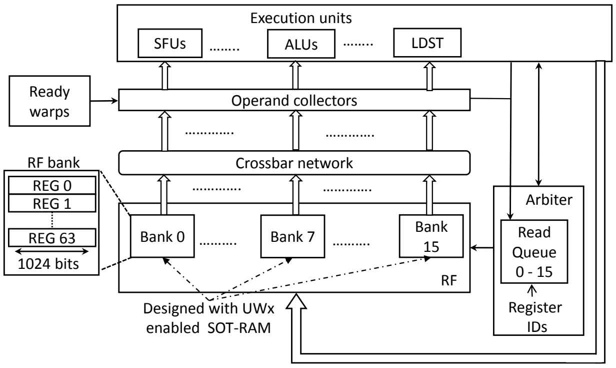

Figure 4. GPU RF architecture (UWx refers to UWT or UWA)

IV. Results and Analysis

A. Experimentation platform

Our overall evaluation approach is shown in Figure 5. For circuit-level results, we have employed STT and SOT based MTJ models as proposed in [19] and [15], respectively. For CMOS components, we have used 65 nm TSMC general purpose device models. We design a single bit-cell with read-write circuitry using these models and run SPICE simulation using Cadence Spectre tool to extract cell-level parameters such as current, energy, latency, etc. Then, these cell-level values are fed to NVSim tool. Additionally, RF configuration parameters such as capacity, array organization, optimization constraints, etc., are provided to the NVSim tool. We use subarray size of 256 rows x 256 columns, latency-optimized buffer design and current-sensing read. NVSim generates leakage, read/write latencies and energy values for a given RF configuration which are shown in Table II. Finally, these parameters are employed for modeling RF in GPGPUSim v3.2.2 cycle-accurate simulator [20] which is used for performing architectural simulations.

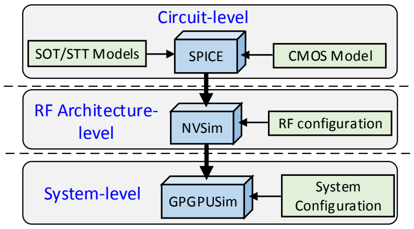

Figure 5. Overall evaluation approach.

The GPU configuration modeled is similar to NVIDIA Fermi GTX480 GPU. The SM frequency is 700 MHz. There are 15 SMs and each can run up to 48 warps. RF in each SM has 16 banks and 128KB capacity. GTO (greedy then oldest) warp scheduling policy is used. We simulate 20 workloads from Lonestar, ISPASS09, Rodinia, Parboil and CUDA SDK suites [20–23] which are shown in Table IV.

Table IV
Simulated workloads and their acronyms

dmr (DMR), mst (MST), LIB (LIB), STO (STO), bfs (BFS), backprop (BCP), heartwall (HWA), hotspot (HOS), pathfinder (PAF), srad-v2 (SR2), cutcp (CUT), mri-q (MRQ), stencil (STC), tpacf (TPF), sgemm (SGM), convolutionTexture (CNT), dct8x8 (DCT), matrixMul (MML), scalarProd (SCP), simpleStreams (STR)

B. System-level results

Figures 6(a) and 6(b) show the results on energy saving and relative performance, where UWT/UWA refers to use of UWT/UWA techniques with SOT-RAM RF. Note that we assume RDE-free STT-RAM, since accounting for RDE will further degrade the performance and energy behavior of STT-RAM. Compared to SRAM RF, STT-RAM RF saves 0.75% energy, whereas SOT-RAM RF saves 18.6% energy. Since the read/write latency with both SRAM and SOT-RAM are 1 cycle, the performance with SRAM and SOT-RAM are the same and thus, SOT-RAM does not harm performance. UWx techniques have the same performance as SOT-RAM and hence, we do not show the performance with them separately. STT-RAM, however, leads to performance loss due to its high write latency. Clearly, even without UWx techniques, SOT-RAM is more energy efficient than STT-RAM.

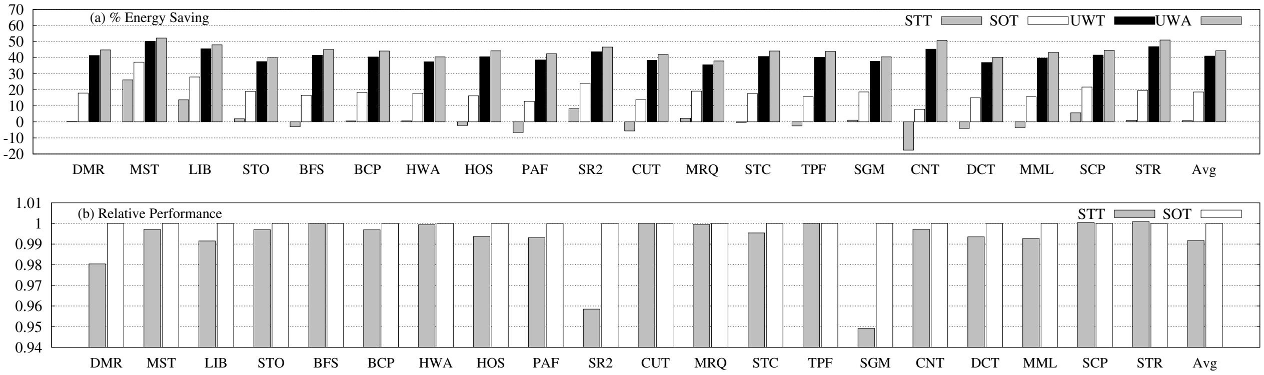

Figure 6. Results for 128KB RF: (a) energy saving over SRAM RF (b) performance relative to SRAM RF

The energy efficiency of SOT-RAM stems from two reasons: its lower write latency which leads to smaller execution time and its lower write energy. The energy saving with SOT-RAM RF depends on the fraction of leakage energy consumption in SRAM RF, e.g., from Figure 2, the leakage energy contributes 52.3% of energy in SRAM RF for MST, which is highest among all benchmarks. Hence, use of SOT-RAM saves largest amount of energy for MST (37.2%). Similarly, leakage energy accounts for more than 20% of total SRAM RF energy in LIB, SR2 and SCP and hence, SOT-RAM provides more than 20% energy saving for all these benchmarks.

The same reasoning also explains the poor energy efficiency of STT-RAM. From Figure 2, average contribution of leakage and write energy in SRAM RF energy consumption are 25.4% and 29.3%, respectively. Compared to SRAM, STT-RAM reduces leakage energy but increases write energy. Hence, STT-RAM saves energy for benchmarks with large leakage energy (e.g., MST, LIB, SR2, SCP) but loses energy for benchmarks with high write energy (e.g., CNT, PAF, CUT, DCT). In fact, STT-RAM leads to 17.6% energy loss for CNT. Clearly, due to high access frequency and dynamic energy of RF, STT-RAM is not suitable for designing RF.

The fraction of unnecessary writes are shown in Figure 1. On average, 81% of bit-writes are unnecessary and hence, UWT and UWA techniques save 41% and 44.3% energy, respectively (refer Figure 6(a)). Compared to SOT-RAM, the additional energy saved by UWx techniques depends on two factors: the fraction of unnecessary bit-writes and the contribution of write energy in total energy consumption of SRAM RF. The highest additional energy saving provided by UWT technique compared to SOT-RAM alone is for CNT (37.5%) and CNT is also the benchmark which has highest contribution of write energy in SRAM RF energy (45.2%). Also, the fraction of unnecessary writes in CNT is 87.4%. For some applications, write energy contributes more than 30% of SRAM RF energy and hence, UWx techniques save much higher energy than SOT-RAM alone. For all benchmarks, UWA saves more energy than UWT. This confirms our insight in Section III-B. Clearly, use of SOT-RAM RF with UWx techniques can reduce the energy consumption significantly.

C. Sensitivity Results

We now experiment with changing just one parameter from that used in main results.

Effect of RF capacity: Since the RF capacity is expected to increase in future GPUs, we experimented with increasing RF capacity per SM to 256KB and the results are shown in Figure 7. The results on unnecessary bit-writes are same as that in 128KB RF and hence, are omitted. Compared to SRAM RF, the energy saving with STT-RAM RF, SOT-RAM RF, UWT technique and UWA technique are 15.4%, 28.1%, 48.9% and 52.0%, respectively. Also, relative performance with STT-RAM RF and SOT-RAM RF are 1.015× and 1.021×, respectively. Clearly, with increasing RF capacity, a larger number of registers remain idle which is reflected in increased contribution of leakage energy in overall energy. Hence, the energy saving provided by SOT-RAM and UWx techniques also increase. Also, SOT-RAM provides better performance than SRAM and STT-RAM with increasing RF capacity. Clearly, our work will be even more useful for next-generation GPUs. On changing the RF capacity to 64KB, MML (matrixMul) fails to execute due to insufficient RF resources. Hence, we do not show results with 64KB RF.

Effect of GPU frequency: We now experiment with changing the frequency from 700 MHz to 1GHz, 1.5GHz and 2GHz (Table V). Results on performance are omitted since it remains close to SRAM. With increasing frequency, contribution of leakage energy in overall SRAM energy reduces and that of write energy increases. Hence, STT-RAM RF consumes even higher energy than SRAM RF. By contrast, SOT-RAM RF provides higher energy efficiency than SRAM RF for all frequencies, although the percentage savings are reduced. Compared to SRAM RF, use of SOT-RAM RF with UWx techniques reduces both leakage and write energy and hence, UWx techniques continue to provide large energy saving with increasing frequency. Clearly, SOT-RAM and UWx techniques will be highly effective for future high-frequency GPUs.

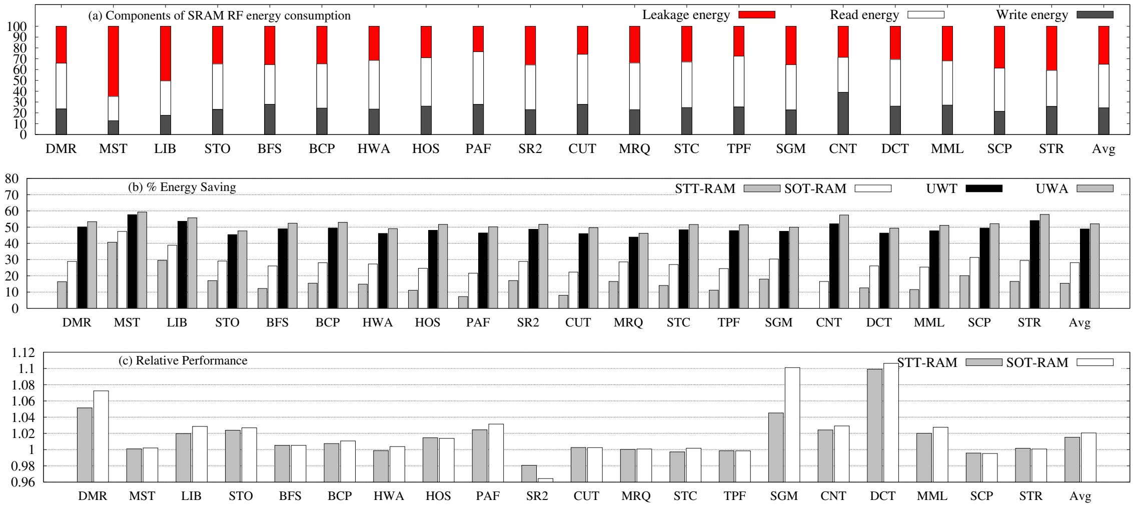

Figure 7. Results for 256KB RF: (a) components of energy consumption in SRAM RF (b) energy saving over SRAM RF (c) performance relative to SRAM RF

Table V
Effect of GPU frequency

	Energy saving (%)				SRAM Energy contribution (%)		
	STT	SOT	UWT	UWA	Leakage	Read	Write
700MHz	0.75	18.60	40.97	44.28	25.36	45.29	29.35
1GHz	-3.31	15.55	39.25	42.76	20.83	48.06	31.11
1.5GHz	-6.74	13.00	37.81	41.48	17.03	50.38	32.59
2GHz	-8.29	11.86	37.15	40.89	15.33	51.43	33.24

Effect of scheduling policy: A change in warp scheduling policy can alter the execution sequence of warps which can affect performance. Hence, we experiment with three other scheduling policies: LRR (loose round robin), two-level (which maintains separate list of ready and waiting warps and uses LRR to schedule from ready warps) and static-warp limiting (which limits the number of concurrently running warps). Table VI shows the results. It is clear that SOT-RAM and UWx techniques are also effective with different scheduling policies and provide high energy savings.

Table VI
Effect of scheduling policy

	Energy saving (%)				Relative Perf. with SOT	Unnecessary Bit-writes
	STT	SOT	UWT	UWA		
LRR	1.38	19.02	41.20	44.48	1.00	81.04%
Two-level	1.53	19.14	41.26	44.53	1.00	81.04%
warp-limiting	1.03	18.79	41.08	44.37	0.99	81.04%

V. Conclusions

We proposed SOT-RAM based GPU RF architecture along with circuit-level techniques to reduce its energy consumption. Experimental results have shown that our approach provides higher energy efficiency than SRAM and STT-RAM while maintaining performance same as that of SRAM.

References

[1] S. Mittal *et al.*, “Design and Analysis of Soft-Error Resilience Mechanisms for GPU Register File,” in *IEEE VLSID*, 2017.

[2] N. Goswami *et al.*, “Power-performance co-optimization of throughput core architecture using resistive memory,” in *HPCA*, 2013, pp. 342–353.

[3] S. Mittal, “A Survey of Techniques for Architecting and Managing GPU Register File,” *IEEE TPDS*, 2016.

[4] N. Jing *et al.*, “An energy-efficient and scalable eDRAM-based register file architecture for GPGPU,” *ISCA*, pp. 344–355, 2013.

[5] N. Jing *et al.*, “Compiler assisted dynamic register file in GPGPU,” in *ISLPED*, 2013, pp. 3–8.

[6] G. Li *et al.*, “A STT-RAM-based Low-power Hybrid Register File for GPGPUs,” *Design Automation Conference*, pp. 103:1–103:6, 2015.

[7] J. Tan *et al.*, “Soft-error reliability and power co-optimization for GPGPUS register file using resistive memory,” in *DATE*, 2015.

[8] M. Mao *et al.*, “Exploration of GPGPU register file architecture using domain-wall-shift-write based racetrack memory,” in *DAC*, 2014, pp. 1–6.

[9] K. Garello *et al.*, “Ultrafast magnetization switching by spin-orbit torques,” *Applied Physics Letters*, vol. 105, no. 21, 2014.

[10] F. Oboril *et al.*, “Evaluation of Hybrid Memory Technologies using SOT-MRAM for On-Chip Cache Hierarchy,” *TCAD*, vol. 34, no. 3, pp. 367–380, 2015.

[11] R. Bishnoi *et al.*, “Low-Power Multi-Port Memory Architecture based on Spin Orbit Torque Magnetic Devices,” in *GLSVLSI*, 2016, pp. 409–414.

[12] S. Mittal, “A survey of soft-error mitigation techniques for non-volatile memories,” *Computers*, vol. 6, no. 8, 2017.

[13] H. Zhang *et al.*, “Red-Shield: Shielding Read Disturbance for STT-RAM Based Register Files on GPUs,” in *GLSVLSI*, 2016, pp. 389–392.

[14] S. Mittal, “A survey of architectural techniques for improving cache power efficiency,” *SUSCOM*, vol. 4, no. 1, pp. 33–43, 2014.

[15] K. Jabeur *et al.*, “Compact Modeling of a Magnetic Tunnel Junction Based on Spin Orbit Torque,” *Transactions on Magnetics*, vol. 50, no. 7, pp. 1–8, 2014.

[16] M. Abdel-Majeed *et al.*, “Warped register file: A power efficient register file for GPGPUs,” in *HPCA*, 2013, pp. 412–423.

[17] P. Zhou *et al.*, “Energy reduction for STT-RAM using early write termination,” in *ICCAD*, 2009, pp. 264–268.

[18] S. Lee *et al.*, “Warped-compression: enabling power efficient GPUs through register compression,” *ISCA*, pp. 502–514, 2015.

[19] A. Mejdoubi *et al.*, “A compact model of precessional spin-transfer switching for MTJ with a perpendicular polarizer,” in *MIEL*, 2012, pp. 225–228.

[20] A. Bakhoda *et al.*, “Analyzing CUDA workloads using a detailed GPU simulator,” in *IEEE ISPASS*, 2009, pp. 163–174.

[21] M. Burtscher *et al.*, “A quantitative study of irregular programs on GPUs,” in *IISWC*, 2012.

[22] S. Che *et al.*, “Rodinia: A Benchmark Suite for Heterogeneous Computing,” in *IISWC*, 2009.

[23] J. Stratton *et al.*, “Parboil: A Revised Benchmark Suite for Scientific and Commercial Throughput Computing,” UIUC, Tech. Rep., 2012.