High Density, Low Energy, Magnetic Tunnel Junction Based Block RAMs for Memory-rich FPGAs

Kosuke Tatsumura*†, Sadegh Yazdanshenas*, and Vaughn Betz*
*Department of Electrical and Computer Engineering, University of Toronto, Canada
†Advanced LSI Technology Laboratory, Corporate R&D Center, Toshiba Corporation, Japan

***Abstract*—Many important applications demand large amounts of on-chip memory both to fully utilize an FPGA's computational capacity and to minimize energy-consuming off-chip memory accesses, leading some recent commercial FPGAs to add higher-capacity on-chip block RAMs (BRAMs). While memory is becoming more important to FPGA designs, SRAM scaling is becoming more difficult because of increasing device variation. An alternative is to build FPGA BRAM from magnetic tunnel junction (MTJ) cells as this emerging embedded memory features a small cell size, low energy usage, and good scalability. In this work, we conduct a detailed comparison study of SRAM and MTJ BRAMs that includes cell designs that are robust with device variation, transistor-level design and optimization of all the required BRAM-specific circuits, and variation-aware simulation at the 22nm node. We find that as the capacity of a BRAM increases, the MTJ benefits of high-density and low-energy increase and its drawback of lower speed is mitigated. At a 256 Kb block size, MTJ-BRAM is 3.06× denser and 55% more energy efficient and its F_{max} is 274 MHz, which is adequate for most FPGA system clock domains. We detail how the non-volatility of an MTJ-BRAM saves energy, especially for narrow write operations which are common for the width-configurable BRAMs of FPGAs. For a RAM architecture similar to the latest commercial FPGAs, MTJ-based block RAMs reduce the FPGA fabric area by 28%, or alternatively could expand FPGA memory capacity by 2.95× with no die size increase.**

I. Introduction

Configurable on-chip memories (block RAMs or BRAMs) are a key FPGA component and recent FPGAs include thousands of BRAMs that in total provide up to 455 Mbits of embedded memory [1]. These BRAMs can be configured to different widths and depths to meet the needs of different applications, and connect to the general FPGA routing to allow the FPGA logic to access memory at much higher bandwidth, lower latency, and lower energy than would be possible with off-chip memory.

System throughput can be either computation-bound or memory-bound [2]. When throughput is computation-bound, we can use all the available computation resources (logic blocks and DSP blocks) on the FPGA. On the other hand, if throughput is memory-bound we are limited by the external memory bandwidth of the chip and will underutilize its computational capacity. An important part of FPGA application design is to re-use data using on-chip block RAMs in order to reduce the required external memory bandwidth and fully utilize the computational capacity. Often this requires increasing the amount of computation by re-organizing operations or re-computing intermediate results rather than fetching them from external memory. In [3] for example, the authors detail a flow that optimizes convolutional neural network implementations on an FPGA by balancing computation and the required external memory bandwidth; generally the most efficient implementations must increase the computation required in order to cache enough data to stay below the external memory bandwidth limit. The work in [4] makes a similar trade-off – the authors increase the computational operations for matrix calculations in order to scale to very large FPGAs without exceeding the on-chip memory available for matrix block storage. In a commercial FPGA-accelerated search engine, BRAMs were the limiting resource for four of the seven FPGA designs in the computational pipeline [5] and in newer and more optimized versions of these designs BRAMs have become the limiting factor in six of the seven designs [6]. Higher-capacity BRAMs would provide application designers with a larger design space of caching and memory scratchpad solutions, enabling implementations that make more full use of the FPGA's computational resources and do not require extra computation to reduce external memory bandwidth.

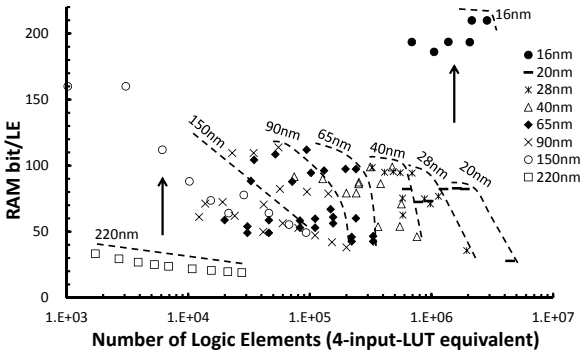

Fig. 1. Trend of RAM bit per LE for Xilinx high-end FPGAs [1].

Another way to utilize all of an FPGA's computational units is to increase the external memory bandwidth by using through-Si-via and Si-interposer technology. By combining die stacking and a sophisticated banking architecture, 3D-stacked DRAMs improve not only bandwidth but also access energy, for example by ∼3 times [7], compared to conventional DRAMs. However, access energy to off-chip memory (ex. 6–7 pJ/bit in [7]) is still much larger than that to on-chip memory (∼0.1 pJ/bit), and even interposer-based DRAM provides only a fraction of the bandwidth of on-chip BRAM. Consequently the latest 14/16nm generation FPGAs from Xilinx [1] and Altera [8] have introduced both high-capacity coarse-grain BRAMs and 3D-stacked DRAMs such as high bandwidth memory (HBM).

Figure 1 shows the trend of FPGA memory richness, RAM bits (including LUTRAM and BRAM) per logic element (LE, a 4-input LUT equivalent), for Xilinx high-end products [1]. After the appearance of BRAM at the 220nm node with a 4 Kb block size and ∼24 bit/LE, BRAM size and density jumped to 18 Kb blocks and ∼78 bit/LE at the 150nm node, stayed there to the 20nm node, and then again jumped at the 16nm node to ∼198 bit/LE with the introduction of a 288 Kb block in addition to the 18 Kb block (∼72% of bits are in 288 Kb blocks). Altera devices show a similar trend [8]. BRAM occupies ∼25% of FPGA fabric area with traditional compositions (∼78 bit/LE) in estimations based on [9], [10] and consumes dynamic energy roughly in proportion to the area [11]. We expect this fraction would increase for memory-rich FPGAs such as 16nm Xilinx Ultrascale+ [1]. Given the importance of adequate BRAM to applications and the already significant fraction of an FPGA's area and power budget devoted to BRAM, a more efficient way to integrate larger amounts of BRAM on FPGAs would be very useful.

To date, BRAMs have relied exclusively on static RAM (SRAM) as their memory element. While application needs motivate more SRAM bits on FPGAs, SRAM scaling is becoming more difficult because of device variation (see Fig. 7). Magnetic tunnel junctions (MTJs) are an emerging alternative memory technology [12], [13] with many attractive features: a small cell size, low energy, good scalability, and non-volatility. Unlike many other non-volatile technologies (e.g. PCM), MTJ has essentially infinite endurance, which is crucial for BRAMs. Replacing SRAM cells in BRAM with MTJ cells could enhance the memory density of FPGAs and the gains would grow as FPGAs continue to become more memory-rich. However, to our

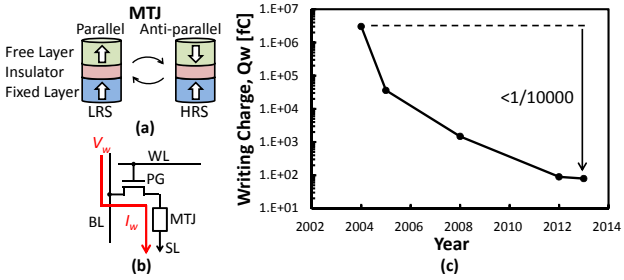

Fig. 2. (a) Magnetic Tunnel Junction (MTJ), (b) single-port 1T-1R cell, and (c) trend of MTJ writing charge (Q_w) [21].

knowledge, quantitative evaluations of the benefits of MTJ-BRAM have not been carried out.

In this work, we assess the benefits of using cutting-edge MTJs in FPGAs through a detailed comparison study of SRAM and MTJ BRAMs including complete cell designs, transistor-level circuit designs for all portions of the BRAM, and analysis of various BRAM size options. We also include the effect of device variation in our BRAM optimization and analysis; this is crucial in both SRAM and MTJ designs, as the weakest bits set the speed and functionality parameters. As a flexible FPGA component, BRAMs have higher functionality than conventional RAM macros for custom ICs: width and port (single/dual) configurability, and programmable block I/Os to the routing. The area overhead to realize this functionality degrades cell efficiency, the ratio of cell area to total block area, and decreases the benefits of introducing small-cell memory, especially for small blocks. We include all the effects of configurability in our circuit design and analysis.

Our contributions include:

- Detailed designs of width- and port-configurable BRAM with a logic-block-compatible routing interface and SRAM/MTJ-RAM cores featuring dual-port access.
- An area, energy, and delay comparison between SRAM and MTJ BRAMs for various block sizes.
- Discussion of the BRAM architecture impact on MTJ benefits.

The methodology is as follows. We design BRAM circuits and memory cells at the 22nm node in Sec. III. In Sec. IV, we implement these circuits in COFFE [14], [15], a fully-automated transistor sizing tool for FPGAs, and evaluate SRAM and MTJ BRAMs after optimization. In Sec. V, we discuss the MTJ benefits and how they are impacted by the FPGA architecture.

II. Background

Despite the importance of BRAMs, the literature on BRAM implementation and optimization is limited. A pioneering work [16], [17] describes the concept of a width-configurable memory block, but its implementation is based on a bidirectional data bus which is not compatible with the directional routing architectures of modern FPGAs. [11] shows BRAM power averages ∼18% of FPGA power on a suite of commercial benchmarks, and that power can be optimized by an intelligent RAM mapping algorithm that implements the user-requested RAMs with the available BRAMs. Work in [18] optimized the area-efficiency of a commercial FPGA by exploring different BRAM block sizes and BRAM to logic block ratios while [19] optimized an FPGA's BRAM architecture for energy-efficiency via block size, banking and BRAM to logic ratio investigations. The work in [18] did not describe how the BRAM areas were calculated, while [19] used the CACTI memory estimator [20] originally developed for processor caches to estimate BRAM metrics. CACTI does not include support for MTJ memories, and we have also found that its area, delay and power estimates differ significantly from the published data available on commercial FPGA block RAMs; for both these reasons we develop our own transistor-level design and optimization of SRAM and MTJ BRAMs in this work.

An MTJ is a 2-terminal resistance-switch non-volatile memory device, composed of two ferromagnetic layers separated by a thin insulating layer (Fig. 2 (a)). The magnetization direction of one layer is fixed (fixed layer), while that of the other (free layer) can be switched by writing currents (I_w) through the insulating layer; the free layer magnetization direction depends on the last write current

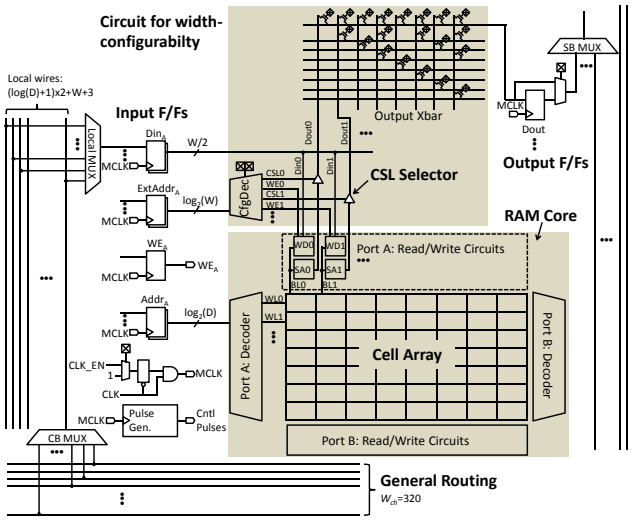

Fig. 3. An architecture of dual-port, width-configurable BRAM.

TABLE I
BRAM SUBCIRCUIT COUNT PER BLOCK

Subcircuit	Size	Count
CB MUX	64:1	(log(D)+1)x2+W+3
Local MUX	25:1	(log(D)+log(W))x2+W+3
Input F/F	-	(log(D)+log(W))x2+W+2
RAM core	DxW	1
Cfg Dec	log(W)toW	2
Output Xbar	2W:W	1
Output F/F	-	W
SB MUX	10:1	Wx8

direction. If the magnetization directions of the layers are parallel, the MTJ is in a low resistance state (LRS) and, if they are anti-parallel, the MTJ is in a high resistance state (HRS). Memorized data stored as the relative magnetic direction is read out by comparing the sense read current (I_{read}) and reference current (I_{ref}). A single-port MTJ cell consists of 1 pass-gate (PG) and 1 MTJ, and is called a 1T-1R cell (Fig. 2 (b)).

Figure 2 (c) [21] shows the trend over time for a crucial MTJ parameter: the required writing charge (Q_w). Q_w determines the write time ($t_w = Q_w/I_w$), energy ($E_w = I_w V_w t_w$) and area ($W_{PG} \propto I_w$) of an MTJ. MTJ development has progressed rapidly in the past decade and Q_w for the state-of-the-art, spin-transfer-torque writing, perpendicular magnetization MTJ (STT-pMTJ) is around 100 fC, enabling its application to embedded RAMs [12], [13] including last level caches for CPUs [22].

The sense read currents for MTJs must be well separated (smaller) from the write currents to avoid changing the MTJ state during a read (read disturb). [23] describes the clamped-reference read scheme which achieves this by using voltage-clamper nFETs to produce controlled read voltages of a bit line (BL) and a reference-BL. [24] shows a dual-port 2T-1R MTJ cell (see Fig. 5 (c)). Our MTJ RAM core is built by combining the robust read scheme in [23] and the dual-port cell in [24] as dual-port functionality is crucial in BRAMs; a set of commercial FPGA benchmarks shows over 80% of RAM instances use some form of dual-port functionality [25].

III. Circuit Design

We design practical BRAM circuits with functionality similar to commercial BRAMs: they have width/port-configurability and interfaces to general routing with routability (multiplexer sizes, etc.) comparable to logic blocks (LBs). Their operation is synchronous with the supplied clocks, and all timing signals are derived from these clocks and self-timed circuits. The SRAM/MTJ cells are designed to have failure rates low enough to support a block size of 256 Kb. The SRAM/MTJ RAM cores support a BRAM-specific operation called *narrow write* in addition to conventional read and write.

The BRAM can be organized to provide a word width of $\times 1$, $\times 2$, $\times 4$, $\cdots$, up to the maximum width (W) supported by the RAM core.

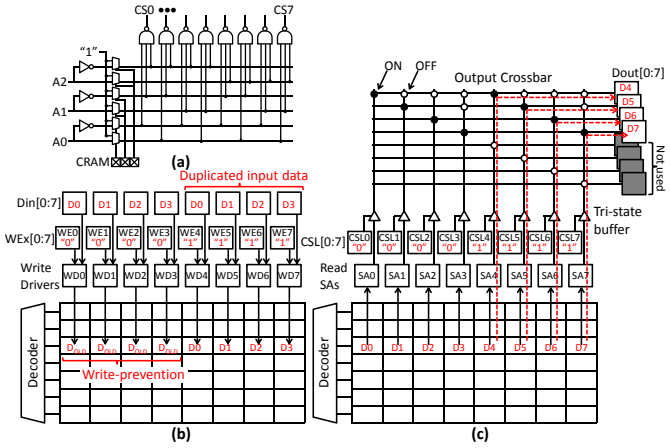

Fig. 4. An implementation of width configurability: (a) configurable column decoder, (b) narrow write, and (c) narrow read.

As the word width decreases, the number of words increases from the native depth (D) of the RAM core (at $\times W$ mode) up to DW (at $\times 1$ mode). We refer to the address supplied to the RAM core in any mode as $addr$ ($log(D)$ bits), and additional address needed for modes other than the widest width mode is referred to as $Extaddr$ ($log(W)$ bits at minimum width and 0 bit at maximum width). The RAM core is dual-port and the BRAM supports single-port (1rw), ROM (1r), simple-dual-port (1r/1w), and true-dual-port (2rw) modes. The widest width mode is not allowed for the 2rw mode to save I/O port area, as otherwise it would roughly double the number of signals going to and from the programmable routing over any other mode.

We follow the FPGA architecture (logic block and general routing) described in [14] (pass transistor, $V_{boost} = +0.2$ V). Also, the same transistor (PTM 22nm HP, V_{DD}=0.80 V [26]) and wire load models (ITRS11 [27]) as in [14] are assumed unless otherwise noted.

The following sub-sections detail our design for the width-configurable BRAM circuit, memory cell, and RAM core.

A. Width-configurable BRAM circuit

Figure 3 and Table I show the BRAM architecture and subcircuit count per block. The BRAM consists of logic-block-like input ports (CB MUX and local MUX), mandatory input F/Fs, the RAM core, a circuit for width configurability, optional output F/Fs, and output ports (SB MUX).

Input signals (address and write-in data) are routed via CB MUXes, local wires, and local MUXes to input F/Fs. The BRAM is synchronous, meaning the input signals must be registered in the input F/Fs, as this makes timing control of the RAM core easier. On the other hand, output F/Fs have bypass MUXes and are optional. Output signals (read-out data) are routed back to general routing via SB MUXes. The supplied clock is ANDed with clock enable (CLK_EN) to generate the memory clock (MCLK). All registers and control pulse generators are triggered by MCLK. The BRAM receives two write enables (WEs) for ports A/B. As shown in Table I, we assume the same size CB/local/SB MUXes as in the logic block [14] to provide similar routability to the routing CAD tool. The number of input ports is $((log(D)+1) \times 2 + W + 3)$, which is determined by the mode that needs the maximum input signals (i.e. $\times W/2$ 2rw mode). This number is the same as the number of local wires because there is no internal feedback, unlike a LB. The number of output F/Fs is W.

The RAM core has no configurability. Unlike a conventional RAM core, however, each write driver in the RAM core receives an *individual write enable* (WEx) to enable $narrow$ write ($D_{in}<W$).

Width configurability is realized by the configurable column decoder (CfgDec) and output crossbar (Xbar), which are programmed with configuration RAMs (CRAMs). $Extaddr$ is supplied to the CfgDec and the CfgDec switches dynamically with $Extaddr$ and outputs W-bit column select signals (CSs). Figure 4 (a) shows an implementation of CfgDec (W=8). It is configured to be either one 3to8, two 2to4, four 1to2, or eight 0to1 (no selection) decoders. The CSs are ANDed with other control signals to generate individual write enable (WEx) and logical column select (CSL) signals, which are used to realize $narrow$ write ($D_{in} < W$) and $narrow$ read ($D_{out} < W$) functions.

Figures 4 (b) and (c) show narrow write and read operations ($\times 4$ mode for $W = 8$ core). In a narrow write, the write-in data is first duplicated to fill the RAM's natural W-bit width using a combination of the local MUX configuration and a data broadcast enabled by the connectivity of the D_{in} wires. The proper data is now available to write in the appropriate cells, and the WEx signals for the appropriate columns are logic high causing the cells in the selected row and in the activated columns to be written. In the other columns, unselected write drivers do *write prevention* operations, which preserve the old data in the cell when the cell is accessed by the word line (WL). Unselected write drivers are disabled by WEx being low so the bit lines (BL) are left in their precharged state and cells in unselected columns retain their old data. Note that write prevention can consume dynamic energy as bit lines are still pre-charged and then partially discharged by the cells in the selected row.

In a narrow read (Fig. 4 (c)), a row of cells selected by a WL are read out simultaneously, but read-out data in unselected columns are cut off at the tri-state buffers controlled by the CSL signals and only the appropriate data is sent on to the output Xbar. The selected data is routed by the output Xbar to output ports. In the example of Fig. 4 (c), the output Xbar connects two sets of four CSL selectors to one set of four output ports and one of the two sets of four CSL selectors is exclusively selected by CSL signals. The size of the output Xbar is $2W$:W as the output pins of both RAM ports are connected to the input pins of the output Xbar; this is necessary to allow the BRAM to function as either a wider (up to $\times W$) 1r1w RAM or a narrower (up to $\times W/2$) 2rw RAM. The output Xbar is implemented as a pass transistor matrix with the minimum switch patterns required to realize all allowable modes.

We use PTM 22nm low power (LP) model ($V_{DD} = 0.95$ V) [26] for the RAM core to suppress SRAM leakage currents. Level conversions are performed at level-shifters located before the input F/Fs (not shown) (up-conversion) and at CSL tri-state buffers (down-conversion).

Read time (T_{read}) (/write time (T_{write})) is defined as the time required by signals starting from input F/Fs to reach output F/Fs (/cells). BRAM area includes all areas for cells, RAM peripherals, and the programmable routing associated with the BRAM tile. BRAM energy includes all dynamic energy consumed by the subcircuits between the input and output F/Fs.

B. Memory cell

We design SRAM/MTJ memory cells to achieve failure rates supporting a 256 Kb block, i.e., R_{fail}=4×10^{-6}, by using the Monte Carlo method while taking into account V_{TH} variation of transistors and MTJ resistance variation.

In this section, the cumulative probability of metric χ (P_χ) is represented by the standard normal deviate (Z), where Z=$(\chi-\mu)/\sigma$, and μ and σ are the average and standard deviation of the distribution of χ. Z is related with P_χ by $Z = CDF^{-1}(P_\chi)$, where CDF^{-1}

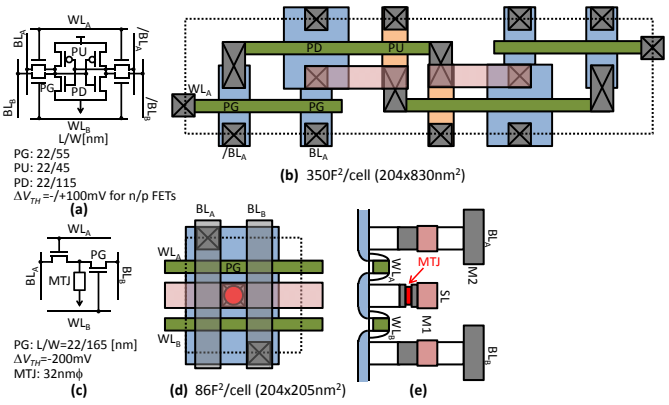

Fig. 5. SRAM and MTJ cells: (a) Dual-port 8T SRAM cell and (b) its layout. (c) Dual-port 2T-1R MTJ cell, (d) its layout and (e) cross-section.

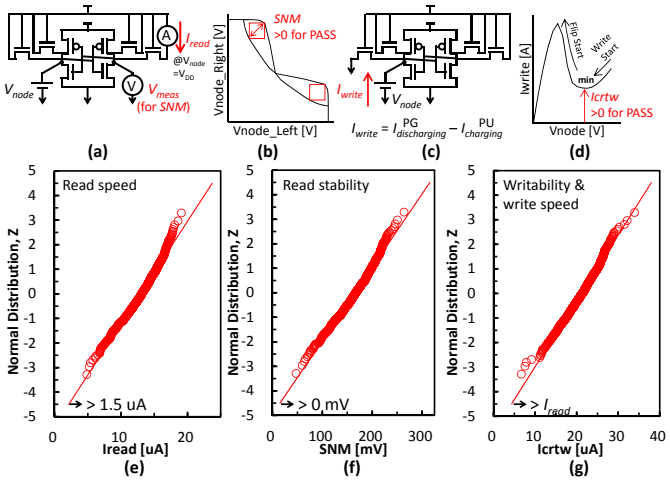

Fig. 6. Characteristics of SRAM cell: (a) Circuit configuration for I_{read} and SNM measurements. (b) Characteristic of cross-coupled inverters during read and definition of SNM. (c) Circuit configuration for I_{write} measurement. (d) Change of I_{write} during write (N-curve). Distributions of (e) I_{read}, (f) SNM, and (g) I_{crtw}.

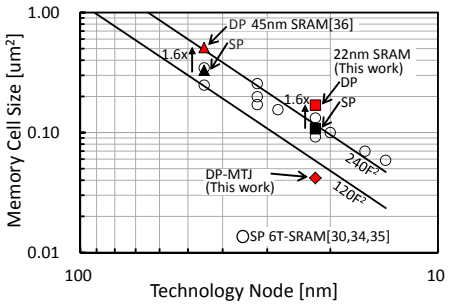

Fig. 7. SRAM cell size trend [30], [34], [35], [36].

is the inverse function of cumulative normal distribution function. $Z = CDF^{-1}(4\times10^{-6}) = -4.5$. $Z = CDF^{-1}(50\%) = 0$. The tail bit cell at $Z=-4.5$ and median cell at $Z=0$ can be quite different in characteristics, and are used to measure delay and energy, respectively. Transistor V_{TH} variation is modelled by Pelgrom plot, $\sigma(V_{TH}) = A_{VT}/\sqrt{LW}$, with $A_{VTn/p} = 2.0/1.7$ mV·μm [28], [29]. Here L and W represent the channel length and width of transistor and Pelgrom plot indicates that we have to increase the channel area of a transistor to suppress V_{TH} variation. MTJ resistance variation is expressed by percentage of σ/μ. In the Monte Carlo method, we generate deviations of V_{TH} and MTJ resistance as normal random numbers with the given standard deviations, assign them to devices in the cell, calculate metrics to be controlled (by SPICE simulations), and repeat this procedure N_{sample} times. The obtained populations of the metrics are ranked (sorted) and the cumulative probability of the N_ith sample is given by $(N_i-0.5)/N_{sample}$, and then converted to Z_i. Due to SPICE runtime, it is not realistic to increase N_{sample} to the bit size of the target block (256 Kb). Instead we characterize the relevant metrics at Z=-4.5 by linear extrapolation of the Z-χ plot.

The layout rules for the cell array were obtained by scaling high-density SRAM cell rules at the 45nm node [30] to the 22nm node. The metal pitch is 90 nm [31], [32] and the corresponding wire load is assumed for BL and WL metal wires.

1) SRAM cell: Figures 5 (a) and (b) show the designed dual-port 8T SRAM cell in terms of circuit, transistor sizing, V_{TH} adjustment (ΔV_{TH}), and layout area. V_{TH} of the (LP) transistor model is adjusted by ΔV_{TH}, as is common in SRAM design. Transistor sizes and V_{TH} of the SRAM cell were optimized to balance read speed, read stability and writability.

Read current (I_{read}), static-noise-margin (SNM), and critical write current (I_{crtw}) [33] are key metrics, and the distributions of those metrics for the optimized cell are shown in Figs. 6 (e), (f) and (g). I_{read} is measured with the circuit configuration in Fig. 6 (a) (PG: open, BL pair: V_{DD}), indicating current drive of the cell to pull down

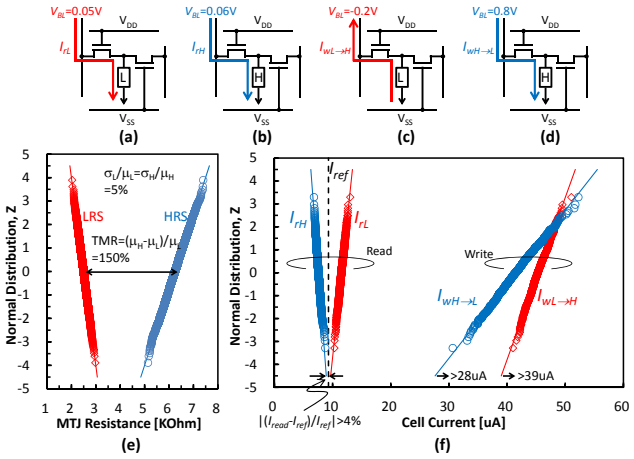

Fig. 8. Characteristics of MTJ cell: BL voltages during (a) L-read, (b) H-read, (c) L→H-write, and (d) H→L-write. Distributions of (e) MTJ resistance, and (f) cell currents.

one of the BLs. The worst-cell read current (I_{read}^{wrst}) and median-cell read current (I_{read}^{med}) determine the read delay and energy respectively. The design target of I_{read}^{wrst} was set to be 1.5 μA, leading to I_{read}^{med} of 12.9 μA. SNM indicates the data-preservation stability during read and must be greater than 0 to preserve bistable states during read; otherwise, read disturb occurs. SNM is defined as the diagonal length of the inscribed square (the smaller one of two) in the characteristic curves of cross-coupled inverters with PGs being open (Fig. 6 (b)). $SNM > 0$ at $Z=-4.5$ is achieved. The critical process during write is the discharge of the high-state data node by PG connected to V_{SS}-driven BL. The current drive of the PG must overcome that of the pull-up pFET (PU) throughout write process; that is, I_{write} defined by $I_{discharging}^{PG} - I_{charging}^{PU}$ must be greater than 0. Otherwise, write fail occurs. I_{write} is measured with the circuit configuration in Fig. 6 (c) and the change of I_{write} as a function of V_{node}, called N-curve, is shown in Fig. 6 (d). I_{crtw} is the minimum current of I_{write} before the cross-coupled inverters flip and it is an indicator of writability and write speed. In the optimized cell, I_{crtw}^{wrst} at $Z=-4.5$ is greater than I_{read}^{wrst}, meaning that the cell is writable and read-speed limited. As V_{TH} of the transistor model is lowered, read and write speed improve but read stability degrades. The optimal ΔV_{TH} is -/+100 mV for n/p FETs. The static leakage current at 85 °C averaged over $N_{sample}=4\times10^4$ is 1.1 nA/cell (1.0 nW/cell in power), so leakage is acceptable even with the V_{TH} adjustments, as we are adjusting LP transistors that start with very low leakage.

To validate that our SRAM design is in line with the industrial state-of-the art, Fig. 7 shows the SRAM cell size trend extracted from recent publications. The single-port (SP) SRAM cell area in the literature deviates from the traditional trend of 120 F^2 [30] after the 45nm node and exceeds 240 F^2 below the 22nm node. The area of a single-port 6T SRAM cell designed by our method described above is 222 F^2 (F=22 nm) and on the trend, lining up very well with the literature. The layout area of our dual-port (DP) 8T SRAM cell is 350 F^2 (F=22 nm). Fewer design points are available for dual-port SRAMs, but note that the area ratio of our dual-port to single-port SRAM cell is 1.6×, which lines up well with the ratio at the 45nm node (1.6×) in [36]. Overall we consider the alignment of our cell sizes, designed from first principles, with that of recent commercial SRAMs to be excellent.

2) MTJ cell: Figures 5 (c) and (d) show our designed dual-port 2T-1R MTJ cell in terms of circuit, transistor size, V_{TH} adjustment (ΔV_{TH}), and layout area. HRS/LRS read currents (I_{rL}/I_{rH}) and L→H/H→L write currents ($I_{wL\to H}/I_{wH\to L}$) are key metrics to be controlled and the distributions of them are shown in Fig. 8 (f).

As a cutting-edge MTJ, we assume $Q_{wL\to H/H\to L} = 116/85$ fC [37] with an diameter of 32 nmϕ, writing time of $t_w = 3.0$ ns [38], tunnelling magneto resistance ratio of $TMR=(\mu_H-\mu_L)/\mu_L=150\%$ [38] [23], and resistance variation of $\sigma_L/\mu_L=\sigma_H/\mu_H=5\%$ [23]. The MTJ resistance distribution is shown in Fig. 8 (e).

The dual-port 2T-1R MTJ cell (Fig. 5 (c)) [24] has two PGs connected to different BLs and enables single-BL access for each port with a bipolar writing scheme. L→H/H→L write are realized

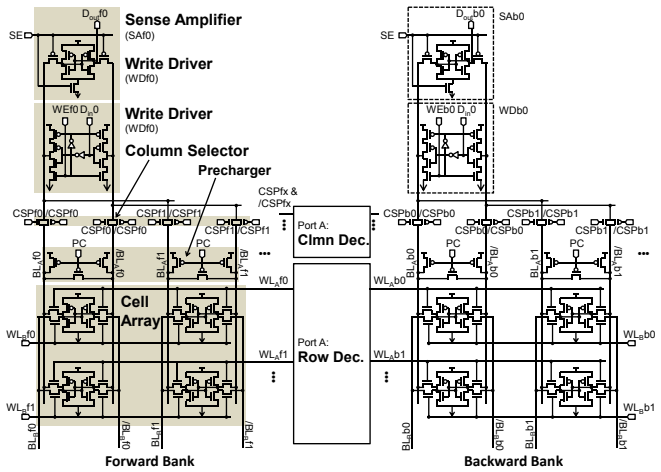

Fig. 9. Architecture of dual-port SRAM core (N_{bank}=2). Node names in forward and backward banks are given suffixes of f and b, respectively.

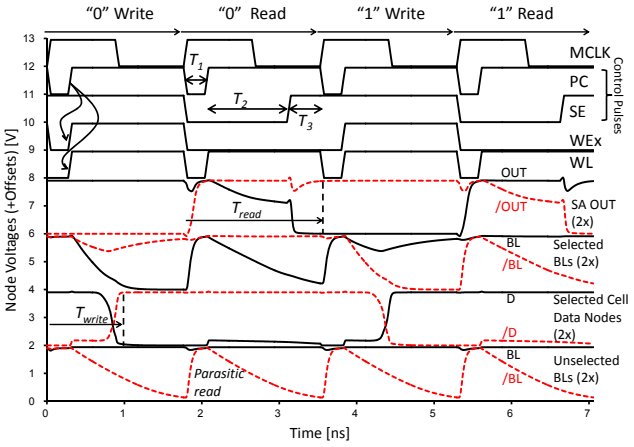

Fig. 10. Operation waveforms of dual-port 128 Kb SRAM with median cells.

by applying negative/positive voltages to the BL (Fig. 8 (c)/(d)). We expect that external negative DC-DC converters added in power management ICs for FPGAs supply the negative voltages (V_{NEG}).

The distributions of I_{rL}, I_{rH}, $I_{wL\to H}$, and $I_{wH\to L}$ were measured with V_{BL} = 0.05/0.06/−0.2/0.8 V. These V_{BL} are realized in the RAM core design (Fig. 11). Note the V_{BL} for HRS read is slightly larger than that for LRS read due to higher MTJ resistance. The assumptions of Q_w and t_w imply that $I_{wL\to H}/I_{wH\to L}$ must be greater than 39/28 μA (writability). Current sense margin ($|(I_{read}-I_{ref})/I_{ref}|$) must be greater than 0 (readability), where I_{ref}=($I_{rH}+I_{rL}$)/2. To prevent read disturb, I_{rH} must be well separated from $I_{wH\to L}$. These three conditions are achieved at Z=−4.5 (Fig. 8 (f)).

In negative write (Fig. 8 (c)), the current drive of the selected PG is enhanced because of the increased V_{gs} (effectively gate boosting) and V_{TH} lowering by forward body biasing. In addition, the increased gate overdrive improves current fluctuation from V_{TH} variation. To avoid undue gate oxide reliability risk [14], we chose V_{BL} of −0.2 V, corresponding to the same V_{boost} as that used for the routing pass transistors. Due to the non-volatile nature of the MTJ, there is no static leakage current, allowing the decrease in V_{TH} of PG. However, an overly low V_{TH} induces unacceptable leakage currents through the PGs in same-column unselected cells during a negative write; the unselected PGs are biased at V_{gs}=V_{ds}=V_{bs}=+0.2 V. The sum of the leakages of the unselected PGs, especially in the case that N_{row} is large, can be larger than the write current for the selected cell. The optimal ΔV_{TH} is -200 mV for PG.

As MTJ resistance increases, current sense margin increases, but the PG size required for the write current also increases. The MTJ resistance in LRS (μ_L) can be adjusted by varying the insulator thickness, and we determined that 2.5 $k\Omega$ is both physically realizable and detectable with a sense margin of $>4\%$. We designed the transistor

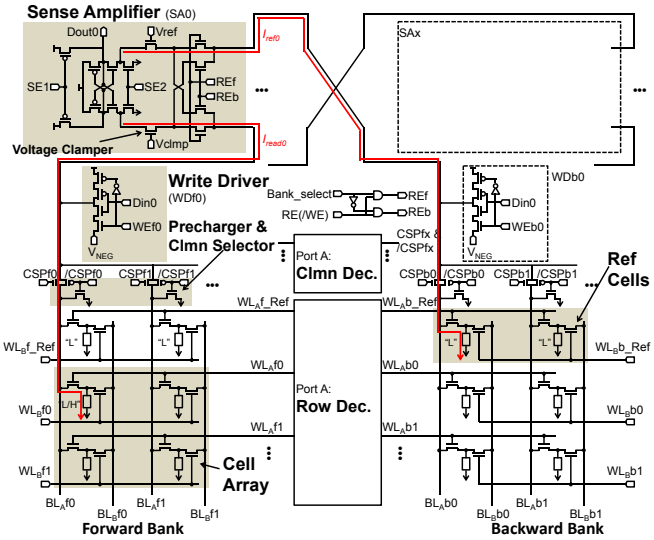

Fig. 11. Architecture of dual-port MTJ-RAM core (N_{bank}=2).

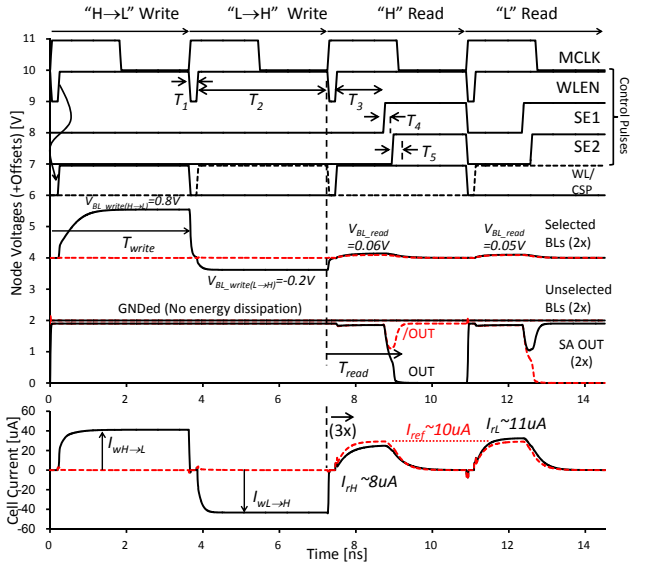

Fig. 12. Waveforms of dual-port 128 Kb MTJ-RAM with median cells.

sizes of the sense amplifier so that it can read the worst-case cell, using a Monte Carlo method that takes into account V_{TH} variation for the sense amplifier.

As shown in Fig. 5, the designed MTJ cell is 86 F^2, which is 4.05× more dense than the SRAM cell (350 F^2).

C. RAM core

Figures 9 and 11 show dual-port SRAM/MTJ-RAM cores, including cell arrays, sense amplifiers, write drivers, BL-precharge circuits, and row/column decoders. Each write driver receives an individual write enable (WEx). By introducing banks and column selectors, a RAM core with D words and W bits per word ($D\times W$) is organized to be $N_{bank}\times N_{row}\times N_{clmn}$, where N_{bank} is the number of banks, and N_{row} and N_{clmn} are the numbers of rows and columns of the cell array per bank. N_{row}=$D/(N_{bank}\times M_{clmn})$ and N_{clmn}=$W\times M_{clmn}$, where M_{clmn} is the column multiplicity – the number of columns that share one column driver through physical column selectors (CSPs). CSP selectors, which are different from the CSL selectors outside the RAM core (Fig. 3), are controlled by the column decoder. Both cores have differential sense amplifiers (SAs). The SRAM cell is self-reference type and connected to the SA with a BL pair. The MTJ cell is single-BL-access type and the reference cells are located in an extra row in the array; when one bank is accessed, the reference cell in the other bank is used as shown in Fig. 11. The SAs for the MTJ core are shared by the adjacent banks. In the following, we explain the four

operations of *read*, *write*, *write prevention*, and *parasitic read* and define each delay component.

1) SRAM core: Figure 10 shows operation waveforms of the SRAM core with median cells. The core follows a V_{DD}-precharge-before-read/write policy. During a read, the selected cell pulls down one wire of the precharged BL pair and the SA detects the voltage difference. During a write, the write driver fully pulls down one of the BL pair, making the selected cell flip.

Control pulses are precharge (PC) and sense enable (SE). WL and WEx signals are triggered (ANDed) with PC to realize the precharge policy. For the SRAM cell, stored data is not guaranteed when WL is asserted unless the BLs are fully precharged to V_{DD}.

The read operation consists of precharge (T_1), amplify (T_2), and latch (T_3) phases. In the precharge phase, all BLs and internal sense nodes in SAs are precharged by precharge pFETs. T_1 is the maximum of the precharging time (T_{PC}) and decoder delay (T_{DEC}). In the amplify phase, one wire of a BL pair is discharged by a WL-selected cell with current drive of I_{read}. We designed a latch-type SA with sensing ability (Δ_{sense}) of 30 mV under V_{TH} variation. BL delay (T_{BL}), the time to develop BL voltage difference of Δ_{sense} with the worst cell, is approximated by $T_{BL}{\sim}C_{BL}\Delta_{sense}/I_{read}^{wrst}$. T_2 is the sum of the WL wire delay (T_{WL}) and T_{BL}. The BL voltage swing is much larger than Δ_{sense} with the median cell, consuming larger energy. In the latch phase, the SA is activated and the read-out data is sensed and kept stable long enough to propagate to the output F/F. T_3 is the sum of the SA delay (T_{SA}), output Xbar delay (T_{Xbar}), and set-up time of the output F/Fs (T_{FF}). T_{read} is the sum of T_1, T_2 and T_3.

A cell write is performed by the write driver after the precharge phase if WEx is high. If WEx is low (write prevention), the write driver is not activated, but one of BL pair is pulled down by the WL-selected cell just as in a read operation, consuming energy.

As shown in Fig. 10, BLs in unselected columns are also discharged by the WL-selected cells during both read and write cycles and must be re-charged in the next precharge phase to avoid data disturb. This *parasitic-read* (PR) consumes energy comparable to the read energy for the selected column. The energy for a read, write or write prevention (WP) operation, E_{op}, is the sum of the energy required by the selected column, E_{op}^{SC}, and the energy from the parasitic read of all the columns not selected by the physical column selector: $E_{op}{=}E_{op}^{SC}{+}E_{PR}(M_{clmn}{-}1)$.

2) MTJ core: Figure 12 shows operation waveforms of the MTJ RAM core with median cells. The core follows a V_{SS}-precharge-before-read/write policy to prevent erroneous writes. In the bipolar write scheme [24], the write driver applies negative/positive voltages to the BL. In the clamped-reference read scheme [23], voltage clamper nFETs biased with V_{clmp}/V_{ref} produce controlled voltages of BL/reference-BL, which induces HRS/LRS read currents (I_{rH}/I_{rL}) for HRS/LRS cells and reference current ($I_{ref}{=}(I_{rH}{+}I_{rL})/2$) for LRS reference cell. The V_{clmp}/V_{ref} generator (not shown) is constructed from current mirrors and replica cells and can be shared by all blocks [23]. I_{ref} is adjusted to be 10 μA.

Control pulses are WL enable (WLEN) and two sense enables (SE1/2); WL and CSP signals are triggered with WLEN. BLs are GNDed by precharge nFETs attached to CSP selectors unless selected (normally-GND). Selected BLs are charged for read/write operations, but again discharged in the next precharge phase. There is no energy consumption for write prevention and parasitic read in the case of MTJ-RAM.

The write operation consists of precharge (T_1) and cell-write (T_2) phases. T_1 is the maximum of T_{PC} and T_{DEC}, where T_{PC} is determined by BL-discharging time after a positive write. T_2 is the sum of T_{WL} and the MTJ writing time (T_w). As shown in Fig. 5 (d), the WLs of MTJ-RAM are 22nm width gate lines with a sheet resistance of 5 $\Omega/\square$ [39]. T_{write} is the sum of T_1 and T_2.

The read operation consists of precharge (T_1), stabilize (T_3), sense (T_4) and latch (T_5) phases. T_3 is the sum of T_{WL} and T_{BL}, where T_{BL} is the BL-charging time by $I_{read/ref}$, i.e. the waiting time for $I_{read/ref}$ to be in a steady state to maximize sense margin. T_4 is the sense time (T_{SA}) for the current sense amplifier (SA) and T_5 is the sum of T_{Xbar} and T_{FF}.

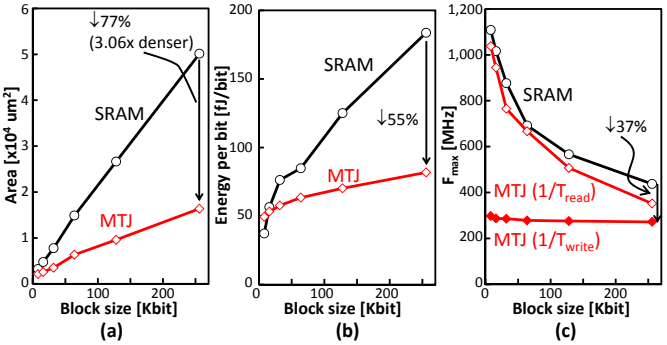

Fig. 13. Area, energy, and speed of SRAM/MTJ BRAMs.

TABLE II
AREA, ENERGY, AND DELAY BREAKDOWNS OF SRAM/MTJ BRAMS

Feature	M8K	M16K	M32K	M64K	M128K	M256K
Organization						
Size	8,192	16,384	32,768	65,536	131,072	262,144
Depth/Width	256/32	512/32	1,024/32	1,024/64	2,048/64	4,096/64
N_{bank}/M_{clmn}	2/2	2/4	2/4	2/4	2/4	2/4
N_{row}/N_{clmn}	64/64	64/128	128/128	128/256	256/256	512/256
Area (um2)						
SRAM						
Cell (efficiency)	1,387(43%)	2,774(58%)	5,548(71%)	11,096(75%)	22,193(83%)	44,386(89%)
RAM Peripheral	553(16%)	638(13%)	839(11%)	1,317(9%)	1,903(7%)	3,117(6%)
Routing	1,332(41%)	1,362(29%)	1,401(18%)	2,472(17%)	2,541(10%)	2,624(5%)
Total (in LBs)	**3,251 (3.1)**	**4,774 (4.5)**	**7,788 (7.4)**	**14,886 (14.1)**	**26,637 (25.3)**	**50,127 (47.6)**
MTJ						
Cell (efficiency)	343(16%)	685(27%)	1,370(39%)	2,741(43%)	5,481(57%)	10,963(67%)
RAM Peripheral	449(21%)	548(21%)	761(22%)	1,184(19%)	1,623(17%)	2,868(18%)
Routing	1,332(63%)	1,350(52%)	1,400(40%)	2,442(38%)	2,448(26%)	2,550(16%)
Total (in LBs)	**2,124 (2.0)**	**2,583 (2.5)**	**3,531 (3.4)**	**6,366 (6.0)**	**9,592 (9.1)**	**16,381 (15.6)**
Density(vs.SRAM)	**1.53x**	**1.85x**	**2.21x**	**2.34x**	**2.78x**	**3.06x**
Energy (fJ/bit)						
SRAM						
E_{read}(Core+Other)	47 (23+24)	66 (39+27)	85 (57+28)	97 (59+38)	136 (97+39)	191 (148+43)
E_{write}(Core+Other)	29 (22+8)	49 (37+12)	70 (58+13)	75 (59+16)	119 (102+17)	188 (169+20)
E_{wp}(Core+Other)	26 (18+8)	45 (34+12)	65 (52+13)	71 (54+16)	109 (92+17)	164 (145+20)
MTJ						
E_{read}(Core+Other)	40 (16+23)	43 (19+24)	49 (23+26)	59 (24+34)	69 (34+35)	87 (49+38)
E_{write}(Core+Other)	113 (108+5)	120 (114+7)	124 (116+8)	129 (121+8)	134 (126+8)	143 (132+10)
E_{wp}(Core+Other)	5 (0+5)	7 (0+7)	8 (0+8)	8 (0+8)	8 (0+8)	10 (0+10)
Delay (ps)						
SRAM						
T_{PC}/T_{DEC}	147/163	152/164	184/175	209/182	274/208	277/254
$T_{WL}+T_{BL}$	185+247	268+244	274+377	440+365	438+624	455+1,122
$T_{SA}+T_{Xbar}+T_{FF}$	142+146+20	142+146+20	142+145+20	142+268+20	142+267+20	142+272+20
T_{read}	902	984	1,141	1,445	1,765	2,287
Fmax (MHz)	**1,108**	**1,016**	**877**	**692**	**567**	**437**
MTJ						
T_{PC}/T_{DEC}	32/163	56/164	71/174	90/192	195/201	204/250
$T_{WL}+T_{CELL\text{-}W}$	193+3,000	319+3,000	322+3,000	396+3,000	425+3,000	423+3,000
T_{write}	3,356	3,483	3,496	3,558	3,626	3,672
Fmax_write(MHz)	**298**	**287**	**286**	**279**	**276**	**272**
$T_{WL}+T_{BL}$	193+230	319+199	322+432	396+409	425+835	423+1,642
T_{SA}	213	211	215	214	221	237
$T_{Xbar}+T_{FF}$	146+20	146+20	145+20	269+20	268+20	268+20
T_{read}	964	1,059	1,307	1,500	1,970	2,839
Fmax_read(MHz)	**1,037**	**944**	**765**	**667**	**508**	**352**

IV. COMPARISONS BETWEEN SRAM AND MTJ BRAMS

We compare SRAM/MTJ BRAMs in terms of area, energy, and delay after optimization of transistor sizes by COFFE. We implemented all the BRAM architecture subcircuits described in Sec.III.A in COFFE and defined the start and end points for each timing path. COFFE evaluates the delay and area of each subcircuit by using transistor-level simulation (SPICE) and a revised version of the minimum-width transistor area model ($1MWTA$=70 F^2), and determines transistor sizes to minimize overall area-delay product with an iterative optimization. Most of the RAM core components (cells, SAs, write drivers and CSPs) are not sized by COFFE and instead manually determined sizes are used, but the precharge FETs are sized by COFFE depending on the BL length. The layout areas in Fig. 5 are used for cells.

Figure 13 shows the area, energy, and speed of SRAM and MTJ BRAMs for 8 Kb to 256 Kb blocks. The array organizations and area, energy and delay breakdowns are detailed in Table II. We adopted WL-repeaters (division by 2) for M128K and M256K blocks and

TABLE III
AREA, ENERGY, AND SPEED OF COMMERCIAL BRAMS [1], [8]

Vendor	Device	Process (nm)	Size (Kbit)	Area* (LABs)	E_{read} (fJ/bit)	E_{write} (fJ/bit)	Fmax (MHz)
Xilinx	Ultrascale+	16	18	–	–	–	825
		16	288	–	–	–	650
	Ultrascale	20	18	–	–	–	660
Altera	Arria-10	20	20	–	160	154	730
	Stratix-V	28	20	4.0	196	185	600
	Stratix-IV	40	9	2.9	228	223	550
		40	144	26.7	503	482	465

*After refs. [9], [10]

precharger-duplication ($2\times$) for M256K block to improve the speed at the expense of area.

For validation, Table III lists the area, energy, and speed of Xilinx's and Altera's commercial SRAM-BRAMs. E_{read}/E_{write} of Altera's BRAMs were obtained from Altera Early Power Estimator [8] by dividing the operation energy of full-width simple-dual-port mode by width. The area ratio of BRAM to LB (BRAM area as a multiple of LB area) is an important factor to determine the benefits of introducing small-area BRAMs to FPGAs. The area of the LB (N=10, k=6) by COFFE is 1053 μm^2. The LB/BRAM architectures in this work well reproduce the BRAM:LB ratios of Altera's FPGAs; our M8K, M16K, and M128K areas (Table II) are 3.1, 4.5, and 25.3 in LBs, while Altera's M9K, M20K, and M144K are 2.9, 4.0, and 26.7 in LABs (also N=10, k=6) [9], [10]. Although the direct comparison in energy and speed is difficult since there is technology/design dependency (Table III), we think our methodology gives a reasonable baseline comparable to the commercial BRAMs; E_{read}/E_{write} in this work are somewhat smaller than those at the 20nm node of Altera's BRAMs, but the energy variation with BRAM size is similar – the E_{read} ratio of M128K:M8K ($2.9\times$) is close to Altera's ratio of M144K:M9K ($2.2\times$). F_{max} (1016/877 MHz) of our M16/32K can be compared to Xilinx's M18K (660 MHz) and Altera's M20K (730 MHz) at the 20nm node. F_{max} ratios of M256K:M16K (43%) and M128K:M8K (51%) in this work are somewhat lower than Xilinx's M288K:M18K ratio (79%) and Altera's M144:M9K ratio (85%).

As block size increases, cell efficiency (the ratio of cell area to total BRAM area) increases (Table II) and the SRAM:MTJ BRAM area ratio (MTJ high-density benefit) increases from 1.53 at M8K to 3.06 at M256K, approaching to the cell area ratio of 4.05. We observe no large difference in RAM peripheral and routing (including CfgDec and output Xbar) areas for SRAM/MTJ BRAMs. Routing area depends mainly on W of the BRAM. Note that FPGA-specific routing areas occupy more than 50% of the area of MTJ-BRAMs for M8K and M16K, limiting the area reduction possible with MTJs for these smaller BRAMs.

In Table II, E_{core} includes energy directly related to BL operations (cells, prechargers, sense amplifiers, and write drivers), and E_{other} corresponds to the other energy consumption including D_{in} wires (only for write), the output Xbar (only for read), all decoders, WL wires, etc. The major contribution to E_{other} is from data transfer (D_{in} wires and output Xbar), and other components are relatively small when considered as energy per bit. E_{other} is smaller for MTJ-BRAM because all wires running in the WL-direction are shorter. The parasitic read energy for unselected columns, $E_{PR}(M_{clmn}-1)$, accounts for a large part of SRAM read/write energy and becomes more pronounced for larger blocks. In spite of the large cell write energy of MTJ, the write energy of MTJ-RAM is lower than that of SRAM for M256K. This is because as block size increases, BL discharge/charge energy becomes large with respect to cell write energy and there is increasing parasitic read energy only for SRAM. The low-energy benefit of MTJ-RAM becomes more clear when we consider the write energy per *block* considering the zero write prevention energy of MTJ BRAM. Figure 14 shows how the write energy per block depends on the configured width for the M256K blocks. In the full-width mode write energy per block is 24% smaller for the MTJ-BRAM, but the improvement increases to 92% at the minimum-width mode. Note that even in the second-widest mode ($\times W/2$ mode) half of the write drivers perform write prevention.

The access energy in Fig.13 (b) is averaged over read, write and write prevention operations with weights of 1:0.5:0.5 based on the following observations and assumptions. First, we conservatively assume

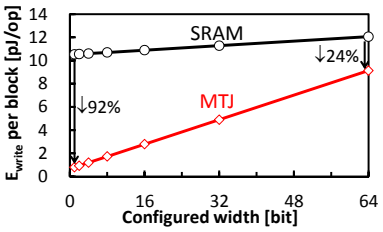

Fig. 14. Dependency of write energy per block on configured width for 256 Kb SRAM/MTJ BRAMs.

that writes are as frequent as reads, as the MTJ BRAM has a higher write energy than read energy. We examined the distribution of RAM configurations after RAM mapping for the suite of 69 benchmark circuits [25] used to develop the Stratix-V architecture and found that the percentage of full-width mode is around 25% and the average configured width is less than 50% of full-width. Hence roughly half of the write drivers will perform write prevention rather than write, leading to the above operation mix of 1 read per 0.5 write and 0.5 write prevention operations. As shown in Fig. 13 (b), the access energy of MTJ-BRAM is larger than or comparable to those of SRAM-BRAM at small blocks such as M8K and M16K, but falls below it at large blocks. The 256 Kb MTJ-BRAM is 55% more energy efficient than its SRAM counterpart. In addition, the static leakage power of SRAM cell (1.0 nW/cell) can be a major static-power source for memory-rich FPGAs (ex. 16nm Xilinx Ultrascale+ [1] has 455 Mbits in BRAMs at maximum). MTJ-BRAM can totally cut off the static leakage current.

Fig.13 (c) and Table II show the write-limited and read-limited speeds of MTJ-BRAMs. The cell write time of $t_w = 3.0$ ns makes MTJ BRAMs write-speed limited and F_{max} varies only slightly with capacity, from 298 MHz at M8K to 272 MHz at M256K. While these are below SRAM speeds, we believe BRAMs based on current cutting-edge MTJs could find many applications. First, typical FPGA bus speeds [40] and the system clocks of FPGAs in a commercial datacenter [5] are below 250 MHz, so the MTJ-BRAM speed will be adequate for most clocks and applications. Second, as block size increases, SRAM F_{max} decreases because of increases in T_{WL} and T_{BL}, becoming closer to that of MTJ $F_{max-write}$. Introducing MTJs only as large blocks in addition to small and fast SRAM blocks, or relying on LUTRAM for the fastest RAMs (which are often relatively small FIFOs near I/Os) are both options to allow some portions of an FPGA design to run at higher speeds. As well, STT-pMTJ has a well-understood scalability [13] and its speed improves with scaling. 22nmϕ MTJ [37] ($t_w = 2.0$ ns [41]) would achieve F_{max} of $\sim$400 MHz at the same area, although a more refined fabrication process to suppress variability is needed. Finally, note that if a port of MTJ-BRAM is configured as read-only port (ROM mode or read port of simple-dual-port mode), it can operate at the higher read-limited-speed.

V. ARCHITECTURE-LEVEL MTJ BENEFITS

So far we have performed block-level comparisons of MTJ and SRAM BRAMs; in this section we explore what the density advantage of MTJ BRAMs would mean to the total area and memory capacity of realistic FPGA architectures. Figures 15 (a) and (b) show three typical RAM architectures (*A1*, *A2* and *A3*) and a tile array of LBs and BRAMs: *A1* is Xilinx Virtex-7/Altera Stratix-V (28nm) like, with one 16 Kb BRAM for every 10 logic blocks (i.e. a BRAM spacing S of 10 LBs). *A2* is Altera Stratix-IV (40nm) like, with 8 Kb ($S = 10$) and 128 Kb blocks (S=180). *A3* is Xilinx Ultrascale+ (16nm) like, with 16 Kb ($S = 16$) and 256 Kb blocks (S=70). We assume the LB architecture is (N=10, k=6) and 50% of LBs are capable of supporting 640-bit LUTRAMs with a 15% larger area than a basic LB. Figure 15 (a) also shows the bit compositions for the three architectures, where 1 LB is converted to 25 LEs (its rough 4-LUT equivalent logic capacity). The *A3* architecture has a high memory-richness (204 bit/LE) and a coarse granularity (74% bits in large blocks of 256 Kb size).

Figure 15 (c) shows the FPGA core area reduction achievable with MTJ BRAMs for the three architectures with two scenarios. The scenario denoted *MTJ* replaces both BRAM block sizes with MTJ-BRAMs. The scenario denoted *MIX* replaces only the large blocks with MTJ-BRAMs. Here, we assume the area estimated in this work

(a)

ID	RAM architecture BRAM1 Size/Space	BRAM2 Size/Space	Bit Composition Total bit/LE	LUTRAM bit/LE(%)	BRAM1 bit/LE(%)	BRAM2 bit/LE(%)
A1	M16K/10	-	78	13(16%)	66(84%)	-
A2	M8K/10	M128K/180	75	13(17%)	33(44%)	29(39%)
A3	M16K/16	M256K/70	204	13(6%)	41(20%)	150(74%)
A4	M16K/16	M256K/19	600	13(2%)	41(7%)	546(91%)

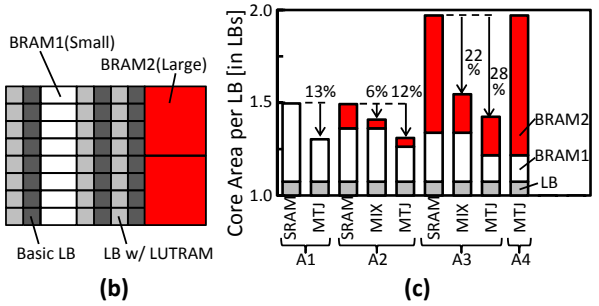

Fig. 15. Dependency of MTJ benefits on RAM architecture.

(Table II) for each block and the area of the basic LB (without LUTRAM) is 1053 μm^2. The vertical axis of Fig. 15 (c) shows FPGA core area per LB including BRAM area, in basic LB counts. With the *MTJ* scenario, MTJ BRAMs reduce core area by 13% and 12% for the *A1* and *A2* architectures and by 28% for the *A3* architecture. For the *A3* architecture, 22% area-saving is achievable with the *MIX* scenario. We believe that with a more dense BRAM, FPGA architects would likely increase the on-FPGA storage for the reasons outlined in the Introduction and we show this case as the *A4* architecture with *MTJ* scenario in Figs. 15 (a) and (c). *A4 MTJ* has the same core area as the *A3*-architecture SRAM FPGA, but has 2.95× more bits per LE.

VI. Conclusion

Many FPGA applications can benefit from larger BRAM capacity to avoid off-chip memory bandwidth and access energy constraints, and accordingly the long-term trend is for FPGAs to become more memory rich. We have completed transistor-level designs of BRAM-specific circuits and SRAM/MTJ RAM cores to show that it is possible to replace SRAM cells with MTJ cells while maintaining full BRAM functionality, including width-configurability and dual-port access. We have presented detailed quantitative assessments that take into account not only the cell design but all the related BRAM circuitry and device variation impacts. As block size increases, the MTJ benefits of high-density and low-energy increase and its drawback of slower speed is mitigated. At a 256 Kb block size, MTJ-BRAM is 3.06× more dense and 55% more energy efficient than SRAM-BRAM and MTJ F_{max} is 274 MHz, which is beyond the typical FPGA system clock. MTJ-BRAMs are most effective for memory-rich coarse-grain RAM architectures and for a RAM architecture having 204 bit/LE similar to the latest commercial FPGAs, MTJ replacement reduces FPGA fabric area by 28% or equivalently can increase memory capacity by 2.95× with no die size increase.

As MTJs scale well and only require changes to the back-end (metal stack) of a process, we believe they hold great promise to improve FPGA density. Furthermore, the non-volatility of MTJs opens other opportunities. An FPGA with MTJ BRAM could store its own configuration (saving on-board flash) when there are unused block RAMs. MTJ BRAMs would be immune to soft errors, as the writing charge is larger than radiation-event-induced charge. Finally, MTJ BRAMs could enable low-leakage sleep modes where register state was stored to MTJ before powering down most of the FPGA.

Acknowledgment

The authors would like to thank Andrew Putnam and David Lewis for helpful discussions, and the Connaught Scholarship, Toshiba, Altera, and NSERC for funding support.

References

[1] Xilinx, *www.xilinx.com/*.
[2] S. Williams, *et al.*, "Roofline: an insightful visual performance model for multicore architectures", *Commun. ACM* **52**, pp. 65–76, 2009.
[3] C. Zhang, *et al.*, "Optimizing FPGA-based accelerator design for deep convolutional neural networks", in *FPGA*, pp. 161–170, 2015.
[4] D. Boland, "Reducing Memory Requirements for High-Performance and Numerically Stable Gaussian Elimination", in *FPGA*, pp. 244 – 253, 2016.
[5] A. Putnam, *et al.*, "A reconfigurable fabric for accelerating large-scale datacenter services", in *Int. Symp. on Computer Architecture (ISCA)*, pp. 13–24, 2014.
[6] Andrew Putnam (Microsoft), *private communication*, May 2016.
[7] M. O'Connor, "Highlights of the high-bandwidth memory (HBM) Standard", in *The Memory Forum, a workshop at ISCA 2014*, *www.cs.utah.edu/thememoryforum/mike.pdf*.
[8] Altera, *www.altera.com/*.
[9] H. Wong, *et al.*, "Comparing FPGA vs. custom CMOS and the impact on processor microarchitecture", in *FPGA*, pp. 5–14, 2011.
[10] R. Rashid, *et al.*, "Comparing performance, productivity and scalability of the TILT overlay processor to OpenCL HLS", in *FPT*, pp. 20–27, 2014.
[11] R. Tessier, *et al.*, "Power-efficient RAM mapping algorithms for FPGA embedded memory blocks", *IEEE trans. on Computer-Aided Design of Integrated Circuits and Systems (CAD)* **26**, pp. 278–290, 2007.
[12] S. H. Kang, "Embedded STT-MRAM for energy-efficient and cost-effective mobile systems", in *Symp. on VLSI Technology and Circuits (VLSI)*, pp. 1–2, 2014.
[13] L. Thomas, *et al.*, "Perpendicular spin transfer torque magnetic random access memories with high spin torque efficiency and thermal stability for embedded applications", in *J. Applied Physics* **115**, p. 172615, 2014.
[14] C. Chiasson, *et al.*, "Should FPGAs abandon the pass-gate?" in *FPL*, pp. 1–8, 2013.
[15] C. Chiasson, *et al.*, "COFFE: fully-automated transistor sizing for FPGAs" in *FPT*, pp. 34–41, 2013.
[16] T. Ngai, *et al.*, "An SRAM-programmable field-configurable memory", in *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 499–502, 1995.
[17] S. Wilton, *et al.*, "Architecture of centralized field-configurable memory", in *FPGA*, pp. 97–103, 1995.
[18] D. Lewis, *et al.*, "Architectural enhancements in Stratix V^{TM}", in *FPGA*, pp. 147–156, 2013.
[19] E. Kadric, *et al.*, "Impact of memory architecture on FPGA energy consumption", in *FPGA*, pp. 146–155, 2015.
[20] CACTI, *www.hpl.hp.com/research/cacti/*.
[21] H. Yoda, *et al.*, "The progresses of MRAM as a memory to save energy consumption and its potential for further reduction", in *VLSI*, pp. T104–T105, 2015.
[22] H. Noguchi, *et al.*,"7.5 A 3.3 ns-access-time 71.2μW/MHz 1Mb embedded STT-MRAM using physically eliminated read-disturb scheme and normally-off memory architecture", in *Int. Solid-State Circuits Conf. (ISSCC)*, pp. 1–3, 2015.
[23] T. Tsuchida, *et al.*, "A 64Mb MRAM with clamped-reference and adequate-reference schemes", in *ISSCC*, pp. 258–260, 2010.
[24] X. Bi, *et al.*, "STT-RAM designs supporting dual-port accesses", in *Conf. on Design, Automation and Test in Europe (DATE)*, pp. 853–858, 2013.
[25] David Lewis (Altera), *private communication*, 2015.
[26] Predictive Technology Model (PTM), *ptm.asu.edu/*.
[27] ITRS, *Interconnect Chapter*, *www.itrs2.net/*, 2011.
[28] M. Goto, *et al.*, "The study of mobility-T_{inv} trade-off in deeply scaled High-*k*/Metal gate devices and scaling design guideline for 22nm-node generation", in *VLSI*, pp. 214–215, 2009.
[29] K. Takeuchi,, *et al.*, "Understanding random threshold voltage fluctuation by comparing multiple fabs and technologies", in *IEEE Int. Electron Devices Meeting (IEDM)*, pp. 467–470, 2007.
[30] H. Nii, *et al.*, "A 45nm high performance bulk logic platform technology (CMOS6) using ultra high NA (1.07) immersion lithography with hybrid dual-damascene structure and porous low-k BEOL", in *IEDM*, pp. 1–4, 2006.
[31] G. Nallapati, *et al.*, "Cost and power/performance optimized 20nm SoC technology for advanced mobile devices", in *VLSI*, pp. 1–2, 2014.
[32] C. Auth, *et al.*, "A 22nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors", in *VLSI*, pp. 131–132, 2012.
[33] E. Grossar, *et al.*, "Read stability and write-ability analysis of SRAM cells for nanometer technologies", *IEEE J. of Solid-State Circuits* **41**, pp. 2577–2588, 2006.
[34] TSMC: 28nm tech., in *VLSI*, pp. 64–65, 2012, 20nm tech., in *VLSI*, pp. 1–2, 2014, 16nm tech., in *IEDM*, pp. 224–227, 2013.
[35] Intel: 45nm tech., in *IEDM*, pp. 247–250, 2007, 32nm tech., in *IEDM*, pp. 659–662, 2009, 22nm tech., in *VLSI*, pp. 131–132, 2012, 14nm tech., in *IEDM*, pp. 71–73, 2014.
[36] K. Nii, *et al.*, "A 45-nm single-port and dual-port SRAM family with robust read/write stabilizing circuitry under DVFS environment", in *VLSI*, pp. 212–213, 2008.
[37] K. Ikegami, *et al.*, "Low power and high density STT-MRAM for embedded cache memory using advanced perpendicular MTJ integrations and asymmetric compensation techniques", in *IEDM*, pp. 650–653, 2014.
[38] H. Noguchi, *et al.*, "A 250-MHz 256b-I/O 1-Mb STT-MRAM with advanced perpendicular MTJ based dual cell for nonvolatile magnetic caches to reduce active power of processors", in *VLSI*, pp. C108–C109, 2013.
[39] J. Kittl, *et al.*, "Ni based silicides: material issues for advanced CMOS applications", in *Int. Symp. Advanced Short-time Thermal Processing for Si-based CMOS Devices*, p. 177, 2003.
[40] M. Abdelfattah, *et al.*, "The case for embedded networks on chip on field-programmable gate arrays", *IEEE Micro* **34**, , pp. 80–89, 2014.
[41] D. Saida, *et al.*, "Low-current high-speed spin-transfer switching in a perpendicular magnetic tunnel junction for cache memory in mobile processors", *IEEE Trans. on Magnetics* **50**, p. 3401105, 2014.