Energy-Efficient Reconfigurable Computing Using a Circuit-Architecture-Software Co-Design Approach

Somnath Paul, *Student Member, IEEE*, Subho Chatterjee, *Student Member, IEEE*, Saibal Mukhopadhyay, *Member, IEEE*, and Swarup Bhunia, *Senior Member, IEEE*

Abstract—**Reconfigurable computing frameworks such as field programmable gate array (FPGA) provide flexibility to map arbitrary applications. However, their intrinsic flexibility comes at the cost of significantly worse performance and power dissipation than their custom counterparts. Existing design solutions such as voltage scaling and multi-threshold assignment typically trade off energy for performance or vise versa. In this paper, we show that an integrated circuit-architecture-software co-design approach can be extremely effective to simultaneously improve the power and performance of a reconfigurable hardware framework, leading to large improvement in energy-delay product (EDP). First, we select a spatio-temporal reconfigurable computing architecture based on 2-D memory-array. Applications are mapped to memory as multiple-input multiple-output lookup tables (LUTs) and are evaluated in temporal manner inside a computing element. Multiple such computing elements communicate spatially through programmable interconnects. Next, we exploit the read-dominant memory access pattern in reconfigurable hardware to design an asymmetric memory cell, which provides higher read performance and lower read power leading to improvement in the overall EDP during operation. We note that the proposed memory cell is also asymmetric in terms of its content, providing better read power for one of the logic states (logic "0" or "1"). Based on this observation, next we propose a content-aware application mapping approach, which tries to maximize the logic "0" or logic "1" content in the lookup tables. A design flow is presented to incorporate the proposed architecture, asymmetric memory cell design and content-aware mapping. We show that for both nanoscale complementary metal–oxide–semiconductor (CMOS) [static random access memory (SRAM)] as well as emerging non-CMOS [spin torque transfer random access memory (STTRAM)] memory technologies, such a co-design solution can achieve significant improvement in system EDP over a conventional FPGA framework.**

Index Terms—**Asymmetric memory, content-aware mapping, low power, reconfigurable hardware.**

I. Introduction

MODERN reconfigurable computing platforms offer several major advantages over custom application specific integrated circuit (ASIC), such as reduced design cost, rapid prototyping, and flexibility to map arbitrary applications. Since their first introduction, field programmable gate arrays (FPGAs) have become the most popular reconfigurable computing platform for implementing digital circuits. The basic structure of the FPGAs have continued to consist of configurable logic blocks (CLBs) and a programmable interconnect (PI) matrix [1]. However, the downside of the reconfigurable nature is that the design mapped on a FPGA platform operates roughly three times slower, occupies 10 times more area and consumes almost two times the power compared to an ASIC implementation at the same technology node [2]. The primary reason behind such a penalty is the elaborate PI architecture which accounts for about 80% of power, 60% of delay, and 75% of area in deep submicron processes [3]. As the process shrinks, this interconnect delay does not scale in the same manner as the logic delay, leading to poor technology scalability of performance for the conventional FPGA platform [2].

In order to minimize the area dedicated to PIs, researchers have proposed mapping larger multiple-input, multiple-output partitions as lookup tables (LUTs) to embedded memory blocks (EMBs) present inside conventional FPGA frameworks [4]–[6]. A larger partition size mapped to these EMBs not only minimizes the area to implement the logic but also improves the performance of the system by reducing the contribution from the interconnects. We refer to these frameworks as "EMB based Heterogenous FPGA." Although these frameworks employ conventional two-dimensional (2-D) memory array for logic computation, it relies on a purely spatial computing model similar to that of conventional FPGA. Contrary to the purely spatial computing model, researchers have also explored the possibility of time-multiplexed hardware reconfigurable architectures [7]–[9], which allow an user to trade off between a purely spatial computing model and one that involves temporal multicycle evaluation within a single functional unit. These frameworks differ from conventional FPGAs in the following ways.

- Logic partitions with much larger number of inputs and outputs are evaluated inside each computing element over multiple cycles in topological order. The computing elements communicate among them in spatial manner.
- Logic partitions are mapped as LUTs to fast and dense 2-D memory arrays at each reconfigurable computing block.

Since memory array is used as the underlying reconfigurable computing fabric, we refer to these alternate spatio-temporal frameworks as memory-based computing (MBC) frameworks.

Memory plays an important role in the delay and power consumption for the MBC frameworks and EMB-based heterogenous FPGAs. Hence, any optimization in the memory design and subsequent changes in the mapping flow can significantly improve the energy delay product (EDP) in these computing

Manuscript received January 04, 2011; revised June 13, 2011; accepted July 20, 2011. Date of publication September 23, 2011; date of current version November 09, 2011. This work has been supported by the National Science Foundation (NSF) under Grant EECS-1002090 and Grant EECS-1002237. This work was recommended by Guest Editor V. Narayanan.

S. Paul and S. Bhunia are with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: sxp190@case.edu; skb21@case.edu).

S. Chatterjee and S. Mukhopadhyay are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: subho.chatterjee@ece.gatech.edu, saibal@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2011.2165232

platforms. We note that the memory arrays used in reconfigurable computing [4]–[8] have a read-dominated access pattern, while write occurs infrequently only during reconfiguration. Leveraging on this observation, we present a novel memory cell design that offers high read performance and improved read power compared to the conventional design. Moreover, we note that read power for the cell is significantly skewed depending on whether the cell is storing a logic "1" or logic "0" value. Based on this observation, we propose a novel application mapping algorithm, which aims at skewing the LUT contents to contain more logic "0"s or "1"s. In particular, the paper makes the following contributions.

1) It presents an integrated circuit-architecture-software co-design and co-optimization solution for achieving EDP improvement in reconfigurable computing frameworks. The architecture considered utilizes 2-D memory arrays as computing elements [4]–[8]. Effectiveness of the proposed approach is analyzed for both fully spatial and spatio-temporal memory based computing platforms.
2) It presents a novel memory cell design for memory-based reconfigurable computing frameworks which offers higher read performance and lower read power compared to the conventional design. The proposed cell design exploits the read-dominant access pattern in embedded memory to improve both energy and performance of mapped applications. Such memory cells suffer from degraded write performance and power. The paper addresses the write-ability issues in such memory using low-cost circuit techniques.
3) It proposes a content-aware application mapping scheme that further minimizes the read power by skewing the logic "0" to logic "1" ratio in LUTs.
4) It validates the effectiveness of the proposed co-design approach for two technologies, namely metal–oxide–semiconductor (CMOS) based volatile static random access memory (SRAM) and nonvolatile spin torque transfer random access memory (STTRAM). Simulation results for a set of benchmark circuits show significant improvement in EDP over conventional CMOS based spatial FPGA architectures and mapping approaches.

The rest of the paper is organized as follows. Section II summarizes earlier works on energy efficient reconfigurable computing and provides motivation for a circuit-architecture-software co-design approach to improve energy efficiency in reconfigurable frameworks. Section III presents the proposed co-design approach and describes the automation framework to validate the approach. Section IV describes the design and optimization of a novel SRAM cell for EDP improvement in computing with memory architectures. Section V reports the EDP improvement achieved through the proposed co-design approach for STTRAM arrays. Section VI concludes the paper.

II. Background

A. Energy Efficient Computing in FPGA

Earlier works have tried to achieve energy efficiency in a fully spatial FPGA computing model using both algorithmic [26] and circuit-level [27] optimization approaches. In [26], the authors propose a design methodology that implements various computational kernels in an energy efficient manner by characterizing various architectures that can implement the same kernel and choosing between the architectures based on performance and resource constraints. Circuit-level techniques for power reduction traditionally incorporate dual-V_{dd} and/or dual-V_t fabrics in the same FPGA device. Designs are mapped to a combination of computing elements with high and low V_{dd} and V_t depending on the performance and power constraints.

B. Computing With Memory

In order to minimize the contribution from PIs in a reconfigurable framework, researchers have proposed alternate frameworks that utilize dense 2-D memory arrays as the computing fabric [7]–[9]. The idea is to partition the input application into multiple-input multiple-output partitions, map these partitions as LUTs in dense 2-D memory arrays and finally evaluate the LUTs in a spatial or spatio-temporal manner. The motivation for computing with memory is manifold. Nanoscale devices are highly suitable for high-density regular memory array design. Emerging nonvolatile memory devices such as spin-torque RAM [22], memristor [28] and phase change memories [29] are extremely attractive for reconfigurable computing. In addition, recent technological innovations such as "3 D integration" suggest that future designs will have more than 60% improvement in on-chip memory speed, power and bandwidth [3]. All these make computing with memory worthwhile to investigate. Memory-based reconfigurable frameworks can be broadly classified into the following two types:

1) MBC Frameworks: Fig. 1(a) illustrates the memory based computing (MBC) framework [8], [9], [11] consisting of multiple computing elements. Each computing element is referred to as the memory-based computational block or MCB. Each MCB consists of a dense 2-D memory array referred to as the *function table*. Multiple-input, multiple-output logic partitions of the target application are mapped as LUTs to this memory array. Outputs from the LUTs are stored in a temporary register file. Operands selected from this register file using the local multiplexor network form the address for the LUTs. Inside each MCB, partitions are evaluated over multiple clock cycles in a topological manner [Fig. 1(b)]. Schedule and connectivity among the partitions is stored in a micro-code format in a small register file referred to as the *schedule table*. A small sequential logic is used to select entries from the schedule table in successive clock cycles. Partitioning and scheduling of the partitions are achieved through software intervention during the process of reconfiguration. Applications are mapped in a spatio-temporal manner across multiple MCBs, which communicate through a programmable interconnect network similar to FPGAs.

2) EMB Based Heterogenous FPGA: As emphasized in [4]–[6], the embedded memory arrays inside conventional FPGA fabrics can be used for logic computations when they are not configured as on-chip memories. Fig. 1(c) shows the packing of multiple smaller LUTs to larger multiple-input multiple-output LUTs mapped to the memory array. Such a mapping of smaller LUTs to larger embedded memory arrays can improve 1) the total area required to map a given netlist [4], [5]; 2) a delay-oriented mapping algorithm [6] achieves significant improvement in performance of the mapped application.

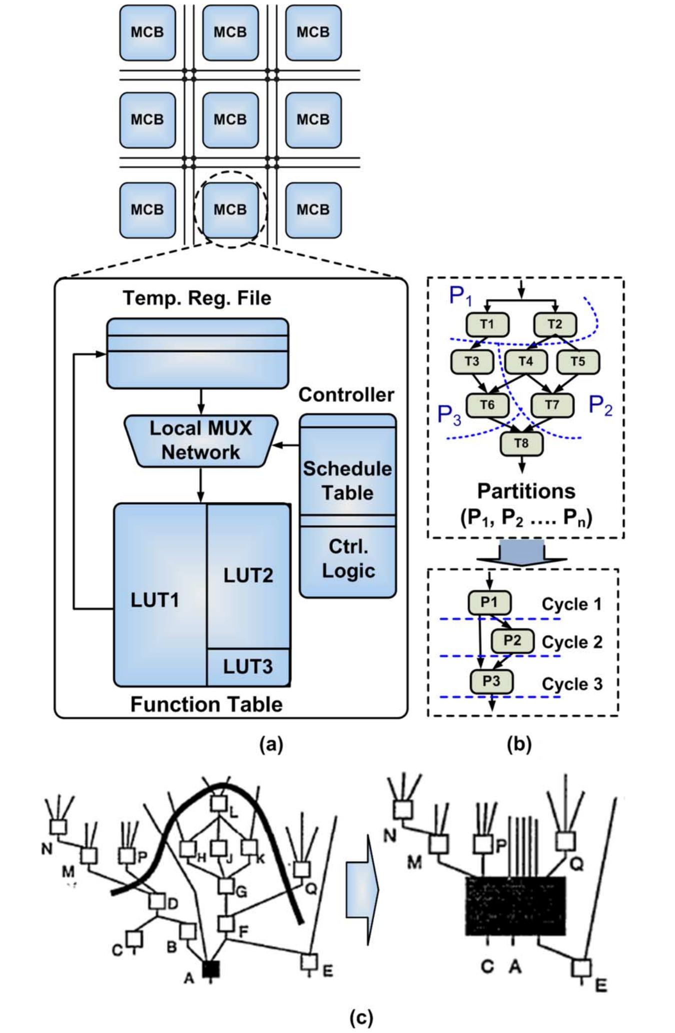

Fig. 1. (a) Hardware organization of a single MCB in a multi-MCB framework [8]. (b) Partitioning of application data flow graph (DFG) into multiple-input multiple-output partitions followed by topological execution of partitions inside each MCB. (c) Packing of logic nodes into multiple-input multiple-output partitions mapped to embedded memory blocks [5].

C. Motivation for a Co-Design Approach

The overwhelming contribution of the PIs to the FPGA area, delay and power demand exploring alternative reconfigurable frameworks which can potentially improve energy efficiency by minimizing the PI contribution. Due to low PI overhead, architectures which incorporate dense 2-D memory arrays for computing can serve as possible solutions to the above problem. Benefits due to memory-array based architecture can be augmented with circuit level optimization techniques, which exploit the read-dominated memory access pattern. Lastly, a content-aware application mapping techniques can further improve the memory power consumption by exploiting the circuit level modifications. Hence, an integrated co-design approach with optimizations at architecture, circuit and software level can be highly effective to achieve drastic improvement in energy efficiency for reconfigurable frameworks.

III. Proposed Co-Design Approach

Fig. 2 shows the proposed circuit-architecture-software co-design approach to realize an energy efficient reconfigurable framework. The integrated solution proposes optimizations at all levels of design abstraction, namely, architecture, circuit, and application mapping software.

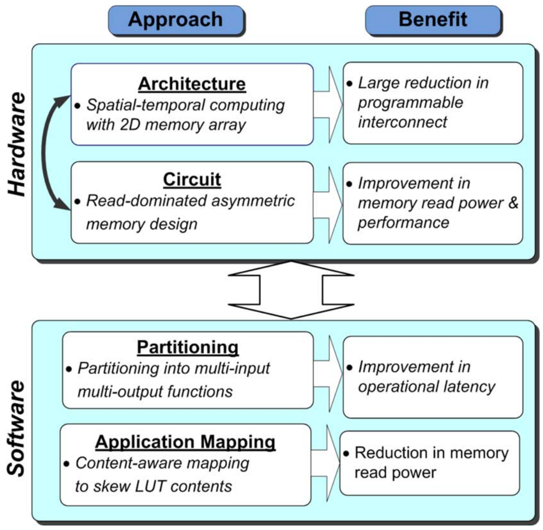
Approach
Benefit
Hardware
Architecture
• Spatial-temporal computing with 2D memory array
• Large reduction in programmable interconnect
Circuit
• Read-dominated asymmetric memory design
• Improvement in memory read power & performance
Software
Partitioning
• Partitioning into multi-input multi-output functions
• Improvement in operational latency
Application Mapping
• Content-aware mapping to skew LUT contents
• Reduction in memory read power

Fig. 2. Proposed circuit-architecture-software co-design approach to improve energy efficiency in reconfigurable computing frameworks.

A. Architecture

The proposed methodology to achieve energy efficient reconfigurable computing rests on memory based computing architectures which can be either fully spatial [4]–[6] or spatio-temporal [8], [9].

B. Circuit

Since memory is primarily used for computation in a memory based reconfigurable framework, memory access is dominated by read rather than write, which occurs only during the process of application mapping. For a given technology, optimizing a memory cell for both read and write pose contradictory requirements [14]. However, exploiting the read-dominated memory access pattern in reconfigurable frameworks, it is possible to come up with a novel memory cell design which offers better read power and performance at the cost of increased write power and performance.

C. Software

For the memory technologies investigated, the read optimized memory cell was found to have higher read power when storing one of the logic states, “0” or “1.” A content-aware storage scheme can, therefore, exploit this asymmetric nature of the proposed memory cell and amplify the energy savings by storing more logic “0” than logic “1” or vice versa. We present a greedy heuristic which reduces active power consumption by preferentially skewing the ratio of logic “0”s to logic “1”s in the LUTs.

1) Heuristic for Content-Aware Mapping: Algorithm 1 presents the pseudocode for maximizing the percentage of logic “0”s stored in the LUTs mapped to the memory array. Input to Algorithm 1 is the hypergraph representation (G(V,E)) of the partitioned netlist and the truthtables for the individual partitions. The partitions are first levelized following their topological order in the netlist. Then beginning from partitions present in the first level, if the truthtable is found to contain more logic "1"s, then each LUT location is inverted. This is only done for LUTs which do not drive any primary output of the circuit or input to any internal state element. In order preserve the correctness of the logic in the following levels, truthtable for the LUTs in the fanout of the modified LUT are rearranged. This rearrangement involves a simple reordering of the 0 and 1 locations inside the truthtable.

Algorithm 1 *Content—aware Mapping to maximize the percentage of logic "0" in the LUTs*

```
Input: 1) DAG representation G(V,E) of the partitioned netlist. 2) Truthtable $T_j$ for the individual partitions $(Part_j)$.
Objective: To maximize the percentage of logic "0" in the truthtables of the individual partitions.
Levelize partitions. Let the total number of levels be L.
for $i = 1$ to $L$ do
  for $Part_j \in Level_i$ do
    if Output from $Part_j \notin$ PO or Flop Inputs then
      if in $T_j$, # of ones > # of zeros then
        Flip 0's and 1's.
        Rearrange the 0 and 1 locations for the LUTs at the fanout of $Part_j$
      end if
    end if
  end for
end for.
```

2) Improvement in Logic "0" Count: We have validated the effectiveness of the proposed mapping approach using a set of standard circuits chosen from the ISCAS'85, ISCAS'89, and MCNC benchmark suites. From Table I, we note that the proposed mapping heuristic achieves almost 49% increase in the percentage of logic "0" count stored in the LUTs.

TABLE I
IMPROVEMENT IN STORED LOGIC "0" COUNT ACHIEVED THROUGH SKEWING OF LUTS

Ckts	# of Partitions	% of Logic '0's		% Impr
		Before Opt	After Opt	
c3540	80	44.44	73.25	28.81
c5315	85	35.17	75.04	39.87
c6288	77	17.71	82.87	65.16
c7552	109	28.45	74.68	46.23
alu4	44	27.45	91.98	64.53
apex2	254	33.87	90.24	56.37
apex4	5	46.68	64.8	18.12
des	115	24.02	75.99	51.97
ex5p	16	46.51	96.51	50
misex3	88	21.78	94.99	73.21
seq	220	27.24	90.06	62.82
spla	204	25.5	88.83	63.33
pdc	241	27.36	91.24	63.88
s1423	41	54.21	68.3	14.09
s5378	106	42.76	82.01	39.25
s38417	794	31.22	77.33	46.11
s38584	745	31.36	81.42	50.06
bigkey	160	18.61	81.55	62.94
elliptic	102	42.92	81.55	38.63
Avg.				**49.23**

D. Integrated Design Flow

Fig. 3 shows the integrated design framework that combines the memory based reconfigurable architectures (both MBC and EMB based heterogenous FPGA) with read optimized memory cell design and content-aware application mapping. Input to the automation flow is a verilog netlist containing the hypergraph representation of the target application and a constraint on the number of partition inputs (M) and outputs (N). For EMB based heterogenous FPGAs, we use *Heteromap* algorithm [6] to partition the target application into multiple-input (M) single-output partitions which are then mapped as LUTs in the memory arrays. For MBC, a novel partitioning algorithm was implemented which ensures partition evaluation in a topological manner. As illustrated in Fig. 3, the partitioning step first maps the hypergraph representation of the target application into $K-$ *input* $1-$ *output* LUTs, where $K \leq M$ using *Flowmap* algorithm [20]. This is followed by the clustering of partitions based on a maximum fanout free subgraph approach as described in [4]. This step tries to share the input space of individual $K-\mathrm{input}\ 1-\mathrm{output}$ LUTs while trying to cluster them into $M-\mathrm{input}\ N-\mathrm{output}$ LUTs. The next step in the MBC design flow is to pack the $M-\mathrm{input}\ N-\mathrm{output}$ LUTs into multiple MCBs based on the design specification of individual MCBs. These specifications include: 1) the amount of memory (Mem) present in each MCB; 2) the maximum number of primary inputs (PI) to and primary outputs (PO) from each MCB in every cycle; 3) the number of temporary registers (Reg) present inside each MCB so as to hold the intermediate partition outputs. Output from the design flows for both the target architectures is a ".net" file, which can be directly interfaced with the VPR toolset [16] for place and route. Since VPR does not support spatio-temporal execution model, in "output.net," operations inside each MCB are unrolled in time and interaction between multiple partitions is represented using a clustered FPGA model. Architectural specifications including interconnection parameters for a clustered FPGA model at 45-nm technology node were obtained from [17]. After LUTs are mapped to multiple MCBs, the proposed content-aware mapping heuristic is used to preferentially skew the ratio of logic "0" to logic "1" in the LUTs. The skewed netlist is finally placed and routed using power-aware VPR toolset [18].

Delay and power estimates for a memory array designed with the read optimized memory cell were obtained from SPICE simulations. Delay and power estimates for other components of a MCB were obtained through synthesis followed by SPICE simulations at 45-nm technology node. Routing delay and power for the PI is then combined with the estimates from individual MCBs by an architecture level power/performance simulator to estimate the overall power and performance for the mapped netlist. Fig. 4(a) and (b) shows the *s38417* sequential benchmark after it is placed and routed on a conventional FPGA and MBC

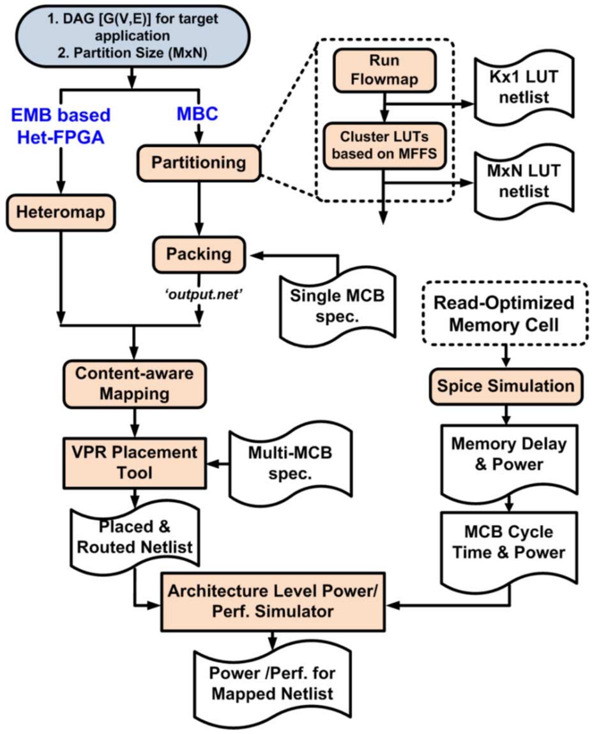

Fig. 3. Flowchart showing the major steps of the automation framework, which integrates the proposed co-design approach.

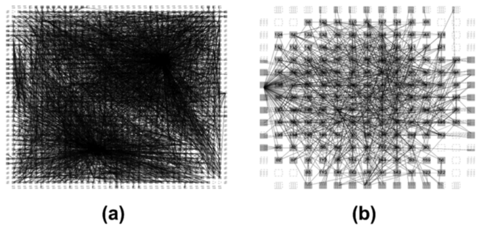

Fig. 4. Improvement in number of computing elements and interconnect requirement (estimated using VPR toolset [16]) for sequential benchmark s38417 when mapped to (a) 65 nm CMOS FPGA and (b) MBC frameworks.

framework respectively. From Fig. 4(a) and (b), we note that use of large multiple-input multiple-output LUTs and local executions inside MCBs leads to considerable improvement in the number of computing elements and routing resources for MBC compared to a baseline FPGA model.

IV. Case Study I—SRAM Array

A. Novel SRAM Cell Design

We propose a novel 6-T SRAM cell [Fig. 5(a)] which employs one-sided write and dynamic read methodology. The new cell benefits us in terms of increased SNM, more tolerance to process variability and reduced read power consumption. Hence for read stability the primary characteristic to be met is a considerable SNM. Also the proposed cell being essentially a 6-T structure the area overhead associated is less than in 7 T or 8 T structures proposed earlier [13]. However, writing to the cell is single-ended and requires word line boost up methods thereby consuming more write power than the conventional 6-T structure. So the proposed structure's energy efficiency over the conventional case is a direct function of the read write ratio. The proposed cell considers reading by controlling the source of the access transistor for read. This concept has been explored earlier in [12]. However, the solution proposed in [12] suffers from the fact that the bitline for the cells are coupled via the high impedence source. This might lead to corrupting data of the unselected cells and giving false reads. We eliminate this problem by maintaining the source of the access transistor at low impedance.

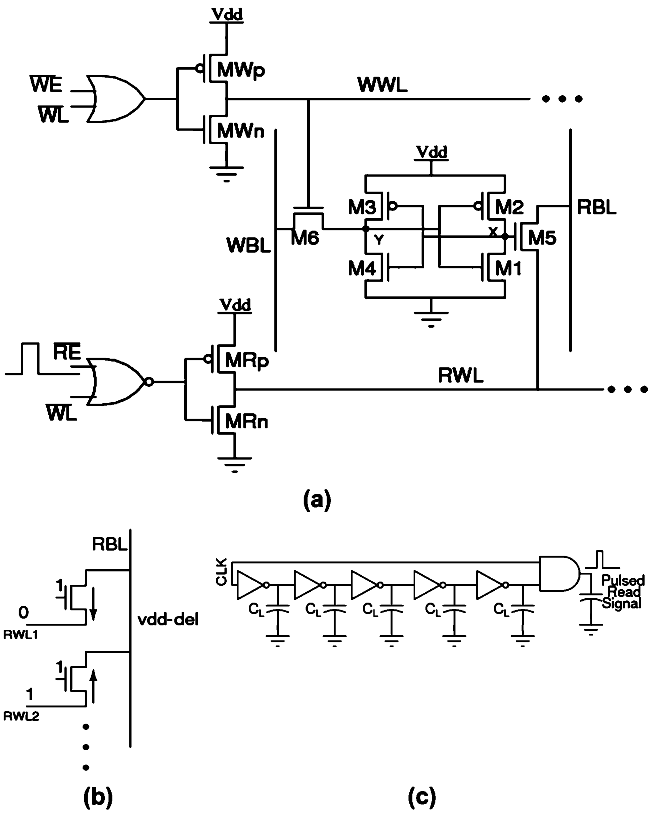

Fig. 5. (a) The proposed 6-T SRAM cell. (b) Pathological case showing the flow of static current from an unselected cell to the selected cell. (c) Pulse generation circuit for read operation.

The read and write mechanisms for this structure is quite different from the conventional 6-T structure. Following is the description of the read and write operations for such a structure:

1) Read: If a particular word line is to be read, the RWL is made low [RWL is derived from RE and WL as shown in Fig. 5(a)]. In presence of read signal and word line selected, transistor M5 has its source set to "0." If it stores a "0" at point X, the transistor does not conduct and the bitline does not discharge. If the transistor stores a "1" at point X, the bitline discharges through the transistor M5 and eventually though MRn. Proper sizing of MRn is necessary to help sink the current from the entire row corresponding to a worst case condition. The greatest advantage with this structure is that the charge cannot enter the storage node during read discharge. Hence possibility of read disturb arising from flipping of data during read is nullified.

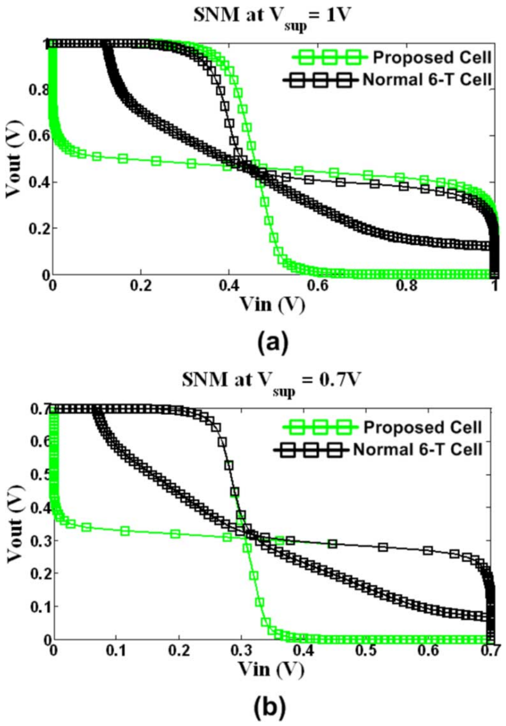

Fig. 6. (a) The proposed cell offers a higher SNM compared to the conventional 6-T SRAM at nominal voltage. (b) The improvement in SNM becomes more marked at lower supply voltages.

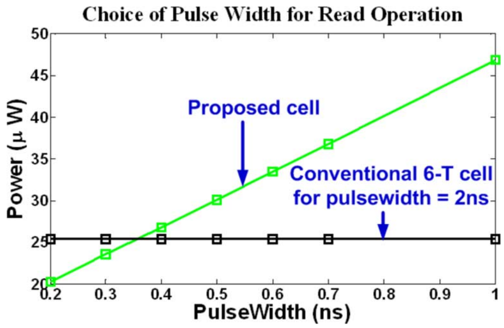

Fig. 7. Choice of pulse width to read from the proposed cell has significant impact on the total power savings in the proposed memory cell.

2) Write: The writes in this case are single ended. So to facilitate write, we use techniques like word line boosting. This however comes at a price of higher energy consumption for writes and affects the overall energy benefit. Hence the cell is suited for cases with low percentage of writes for leveraging maximum performance.

3) Read-Stability: The structure has an improved read-stability due to decoupling of the read and writes such that the cell being read from is isolated from the bitline so that there is no issue of the cell flipping while reading. The SNM curves in Fig. 6(a) illustrate the read-stability advantage of the cell at $\text{supply} = 1$ V. At lower supplies, the distinction is even more prominent as shown in Fig. 6(b).

4) Read Energy Analysis of the Cell: The word line switching energy of the proposed cell is lower as it sees the junction transistor of the read access transistors instead of the gate capacitance of two access transistors. Thus the read energy of the proposed cell is expected to be less. However, one has to confront a pathological case, which is illustrated in Fig. 5(b). When the RBL is precharged and let go, let us consider a case where row 1 is selected and let us also consider unselected cells along the same column storing "1" at the gate of the read transistor. Now as the selected cell discharges via RWL1(shown by the arrow), the bit line voltage drops to $(V_{dd} - V_{th})$. The unselected cells start contributing bringing the RBL upto $(V_{dd} - V_{th})$. Thus there is static power dissipation in the unselected cells for this case. Minimizing this static power dissipation for the worst case is an important design consideration. We propose to achieve it using a pulsed read.

5) Proposed Solution—Pulsed Read: Fig. 5(c) shows the circuit to generate read pulse for the proposed memory cell. While choosing the pulse width we have to keep in mind the fact that it must allow the bit line to be discharged by the required amount for single ended detection (200 mV or higher). Additionally, selection of the pulse width is based upon energy savings one can achieve over the standard 6-T case. In Fig. 7, we demonstrate this fact. With widths below 400 ps, the proposed structure consumes less power as the word line switching power of the conventional case dominates. For reliable operation of the proposed cell and to achieve power savings over a conventional 6-T cell, we have selected a pulse width of 150 ps.

6) Operation at Higher Frequency: The proposed cell is also a better choice for operation at higher frequencies. In the simulation, a 16×64 array of cells is considered with a 8 bit wordsize at 45 nm predictive technology [15]. The SRAM design follows the well known 1:1.5:1.8 ratio for the width of the pull up, pull down and access transistors [14]. The proposed cell is taken so as to match the area for the conventional 6-T SRAM cell. Fig. 8 indicates that the proposed cell is clearly a better choice for low voltage and high frequency operations. Even considering 200 mV of necessary drop instead of the customary 100 mV drop for the differential detection across SRAM, the cell has 2X lower access time than the 6-T structure. It is to be noted here that reading a stored logic "0" does not require the bitline to be discharged. The power expended in reading logic "0" is therefore only the wordline power. The read power reported in Fig. 8(b) considers the worst case scenario of reading logic "1" from the proposed memory cell.

7) Writability: Writing to this cell being single ended, to ensure reasonable writing time, we overdrive the write bit line and the word line. The cell voltage is maintained at 0.6 V whereas the W_{BL} and W_L are maintained at 1 V while writing. Whereas the unselected cells of the selected column do not have problems for such a scheme this can give rise to stored data being flipped for the unselected cells along selected row. This may be avoided by maintaining the unselected cells along the selected row at higher voltage $(V_{dd} = 1\text{ V})$. The overhead for the dual V_{dd} selection logic is small as this logic block caters to an entire MCB. As writes are very infrequent (99.9% operations are read) we think the dynamic energy overhead should not be a major bottleneck to the system energy-efficiency or performance. Rather maintaining robustness will be of increased significance. It is observed that writing into the cell is a slower process compared to read. We simulate the cell at the worst case corner considering

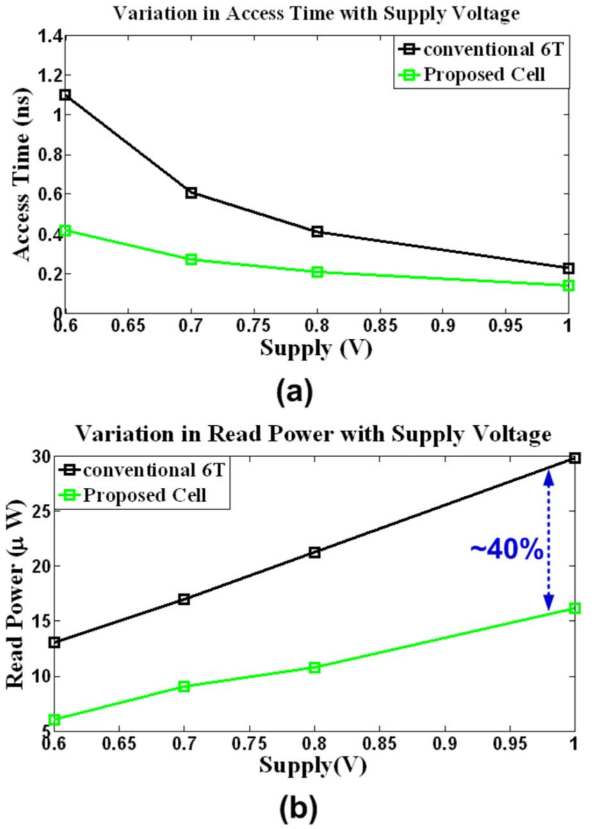

Fig. 8. (a) The proposed memory cell has improved access time compared to the conventional 6-T SRAM cell. The improvement in access time is higher for lower supply voltages. (b) The proposed cell also achieves a 40% reduction in the read power at nominal supply voltage.

50 mV variation in threshold voltage. It is found that the maximum time is taken at a node in making a transition from “0” to “1.” Hence considering the worst case condition we make a study of the writability at 0.6 V. For comparison uses, we compare our proposed cell with an ideal standard 6 T SRAM cell at 0.6 V of supply for word line. The alternate cell requires additional wordline switching for the word line resulting in a write energy overhead of—$\Delta E/E = (V_{\mathrm{WL}}/V_{\mathrm{cell}})^2 - 1 = 1.78$. Considering that the target reconfigurable framework has a read dominated access pattern ($\sim 99.9\%$ of the operations are read), the total energy savings of this structure is given by: $\Delta E = 0.999 \times 0.4 - 0.001 \times 1.78 = 0.397$, where 0.4 denotes the 40% improvement in read energy from Fig. 8(b).

8) Asymmetric Behavior: The proposed memory cell is asymmetric with respect to the power consumed during read “1” and read “0” operations. Due to higher read power during read “1,” it is intended that the memory location being accessed contains logic “0” rather than logic “1.” Furthermore, for the read “1” scenario, it is desired that the unselected rows in the same column have higher number of logic “0” to reduce the static power dissipation during the pulsed read operation. Fig. 9 shows the reduction in read-“1” power as the number of cells storing “0” in the same column increases.

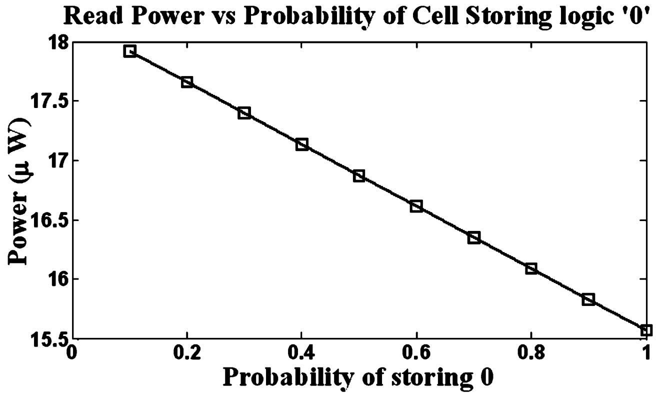

Fig. 9. Reading from a cell storing logic “0” requires less read power compared to a cell storing logic “1.” The total read power thus decreases as the probability of the memory locations storing “0” increases.

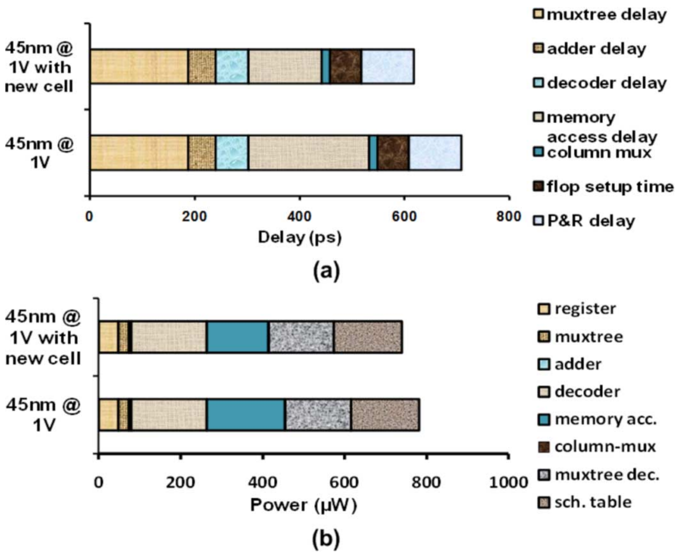

Fig. 10. Breakdown of (a) MCB cycle time and (b) power per partition in each MCB for conventional 6-T SRAM and the proposed 6-T memory cell.

B. Simulation Results

To check the effectiveness of the proposed co-design approach, we implemented the integrated design flow for a SRAM based MBC framework and mapped a set of benchmark circuits (ISCAS and MCNC) using the proposed flow. For the standard benchmark circuits considered in our simulations, a MCB specification corresponding to $M = 12$, $N = 4$, $\mathrm{Mem} = 2KB$, $PI = 32$, $PO = 32$, and $\mathrm{Reg} = 24$ was found to offer significant performance benefit over a conventional FPGA framework. Detailed spice simulations were carried out to estimate the cycle time and power for each MCB. Power and performance contribution of the memory array was estimated for both conventional 6-T SRAM and the new memory cell design proposed in this paper. Simulations were carried out for PTM 45 nm models [15] with nominal supply $V_{dd} = 1$ V. Fig. 10 shows the delay and power contributions from individual components of the MCB to the overall cycle time and power consumed by each MCB. From Fig. 10, we note that for conventional 6-T SRAM based MBC framework, the memory array contributes 29.5% to the MCB cycle time and 24.3% to the MCB power per cycle. From Fig. 10, we note that using the proposed memory cell design gives 12.7% improvement in MCB cycle time and 5.2% improvement in MCB power.

TABLE II
EDP IMPROVEMENT RESULTS IN THE MBC FRAMEWORK WITH NEW MEMORY CELL DESIGN AND CONTENT-AWARE MAPPING

	Delay (ns)			Energy (pJ)				EDP (pJ-ns)			
Ckts	FPGA	MBC with conv. 6T	MBC with alt. 6T	FPGA	MBC with conv. 6T	MBC with alt. 6T	MBC with alt. 6T and map	FPGA	MBC with conv. 6T	MBC with alt. 6T	MBC with alt. 6T and map
c3540	7.29	4.05	3.60	20.05	47.36	43.00	39.21	146.24	191.57	154.59	140.96
c5315†	4.91	3.24	2.88	-	50.32	45.69	41.56	-	162.84	131.40	119.53
c6288	16.59	6.47	5.75	37.16	45.58	41.39	37.26	616.52	295.02	238.07	214.32
c7552†	6.16	5.30	4.85	-	64.53	58.59	53.32	-	341.68	283.87	258.34
alu4	4.45	2.43	2.16	69.84	26.05	23.65	21.03	310.58	63.22	51.01	45.37
apex2	8.16	3.24	2.88	92.83	150.37	136.53	121.70	757.17	486.59	392.66	350.00
apex4	5.35	0.71	0.62	90.56	2.96	2.69	2.48	484.59	2.10	1.67	1.54
des	3.20	2.93	2.66	139.60	68.08	61.81	56.16	447.28	199.34	164.30	149.27
ex5p	2.41	0.71	0.62	8.39	9.47	8.60	7.60	20.25	6.73	5.33	4.71
misex3	5.00	2.43	2.16	73.89	52.10	47.30	41.89	369.52	126.44	102.03	90.36
seq	5.11	3.24	2.88	97.93	130.24	118.25	105.43	499.93	421.46	340.10	303.22
spla	6.86	9.46	9.01	235.40	120.77	109.65	97.93	1614.84	1142.47	987.98	882.31
pdc	7.93	3.24	2.88	232.10	142.67	129.54	115.31	1839.86	461.69	372.56	331.63
s1423	18.81	3.24	2.88	7.71	24.27	22.04	20.23	145.01	78.54	63.38	58.17
s5378	5.51	2.43	2.16	28.46	62.75	56.98	51.35	156.84	152.30	122.90	110.76
s38417	12.42	3.57	3.21	274.90	470.05	426.79	387.05	3413.16	1677.13	1369.15	1241.67
s38584†	5.54	3.57	3.21	-	441.04	400.45	361.19	-	1573.63	1284.65	1158.71
bigkey	6.68	0.71	0.62	64.48	94.72	86.00	77.56	430.86	67.25	53.32	48.09
elliptic	24.84	3.24	2.88	30.30	60.38	54.83	49.44	752.74	195.40	157.68	142.20
Avg Impr.									**50.1%**	**59.5%**	**63.5%**

†Power aware VPR tool failed to map these benchmarks to the FPGA framework.

1) EDP Improvement Results for MBC: With the simulation setup described above, we have estimated the EDP improvement in a SRAM array based MBC framework. For a given input vector, the MBC framework evaluates the target application over multiple cycles. The total execution time for each benchmark circuit is obtained as: $T_{\text{execution}} = T_{\text{cycle}} \times \#\text{of Partitions in Critical path}$, where T_{cycle} denotes the inter-MCB cycle time after the MCBs to which the design is mapped are placed and routed. T_{cycle} is calculated as $\text{Max}\{T_{\text{Inter-MCB}}, T_{\text{pos}}\} + T_{\text{neg}}$, where T_{pos} and T_{neg} denote the delay for positive and negative half of the intra-MCB cycle time, i.e., the cycle time when no inter-MCB communication is required and $T_{\text{Inter-MCB}}$ denotes the worst case MCB to MCB routing delay. As evident from the above expression, part of the inter-MCB routing delay is masked by the positive half cycle of the intra-MCB delay. The total energy expended in the computation was computed by $\text{Energy} = \text{Energy}_{\text{Partition}} \times \#\text{ of Partitions}$, where $\text{Energy}_{\text{Partition}}$ denotes the energy required to compute a single partition. The baseline FPGA model considered for comparison consists of seven-input LUTs present inside a cluster of 10. In order to compare between the spatio-temporal MBC model and the fully spatial FPGA at scaled technologies, we have used twelve-input four-output LUTs in case of MBC and seven-input one-output LUTs in case of FPGAs. Table II first demonstrates that MBC framework with the conventional 6-T SRAM achieves considerable improvement (50.1%) in EDP over a FPGA framework at the same technology node. The improvement in EDP is more pronounced (59.5%) with the use of the proposed memory cell due to higher performance at lower power dissipation. We also demonstrate that the EDP can be further improved (63.5%) with the skewing of the LUT contents to contain more logic "0" than logic "1" values.

TABLE III
EDP IMPROVEMENT IN EMB BASED HETEROGENOUS FPGAS

Ckts	Dly for conv. 6T	Dly for alt. 6T	Logic Energy for conv. 6T	Logic Energy for alt. 6T w/ map	EDP for conv. 6T	EDP for alt. 6T w/ map
c3540	1.8	1.4	120.0	102.0	219.4	138.2
c5315	1.4	1.1	127.5	108.2	174.8	114.3
c6288	3.0	2.2	115.5	97.5	343.5	217.5
c7552	1.9	1.4	163.5	138.8	306.1	197.4
alu4	1.0	0.8	66.0	55.4	69.2	44.4
apex2	1.5	1.1	381.0	320.1	581.8	359.5
apex4	0.3	0.2	7.5	6.4	2.2	1.3
des	0.8	0.7	172.5	146.3	136.6	99.3
ex5p	0.3	0.2	24.0	20.1	7.0	4.1
misex3	1.0	0.8	132.0	110.5	135.6	86.1
seq	1.5	1.1	330.0	277.3	482.1	298.6
spla	1.9	1.5	306.0	257.3	573.1	388.8
pdc	1.5	1.2	361.5	303.5	552.0	354.2
Avg. Impr.						**36.21%**

2) Improvement in EDP for EMB Based Heterogenous FPGAs: Standard benchmark circuits were mapped to twelve-input one-output LUTs using the *Heteromap* mapping algorithm proposed in [6]. Table III shows the delay and energy results for the combinational MCNC benchmarks mapped to the EMB based heterogenous FPGA framework. As we may note from Table III, compared to the baseline seven-input LUT based FPGA, the delay for most of the benchmarks improve considerably when mapped to a EMB based heterogenous FPGA framework. The proposed read-optimized SRAM cell further improves both read delay and read energy over the conventional 6-T SRAM design, which improves the EDP for EMB based heterogenous FPGAs. Table III shows that the proposed read-optimized memory cell along with content-aware mapping approach achieves a 36.21% improvement in EDP in a EMB based heterogeneous FPGA framework.

V. Case Study II—STTRAM Array

Effectiveness of the proposed circuit-architecture-software co-design approach was also validated for the promising STTRAM nonvolatile memory technology. STTRAM has certain distinct advantages over the other prevalent memory technologies. Having essentially a 1 T-1 R structure it forms a dense array with high integration density and being magnetic in nature is tolerant to particle hits. In these features it is better than the SRAM. Being non volatile, it does not require refreshing like DRAM. It has a high write endurance, which makes it a better candidate than flash memories. Another feature of STTRAM is its scalability which makes it an attractive option at scaled dimensions. In the nanometer nodes, leakage is one of the primary forms of energy dissipation. The STTRAM structure has zero standby leakage. All these attributes make STTRAM extremely attractive as a reconfigurable computing fabric. In the MBC framework, the function table inside each MCB is realized using STTRAM arrays. This offers the following benefits. 1) Since the function table holds the configuration for the partitions, it occupies the maximum area inside a MCB. A small footprint of the magnetic tunneling junction (MTJ) device [25] ensures that the area occupied by this memory array is minimized. 2) Nonvolatile nature of the STTRAM array ensures that configuration bits stored in the function table are retained when power is turned down. 3) High read performance and low read power for the STTRAM array results in considerable EDP improvement for a STTRAM based nonvolatile MBC framework.

A. *STTRAM Operation*

The basic building block of a STTRAM cell is the MTJ [Fig. 11(a)]. Each MTJ consists of two ferromagnetic layers (typically CoFe) separated by a very thin tunneling dielectric film (typically crystallized MgO). Magnetization in one of the layers (referred as pinned layer) is fixed in one direction by coupling to an anti-ferromagnetic layer (such as PtMn) [21]. The other ferromagnetic layer (referred to as free layer) is used for information storage. The direction of magnetization of free layer with respect to the pinned layer (i.e., anti-parallel or parallel) can be controlled by the injection of spin-polarized electrons. Hence the MTJ can be switched between two stable magnetic states with high (R_{AP} or R_H) or low (R_P or R_L) resistances and it retains the state without any applied power [Fig. 11(b)]. One of the quality metrics for a MTJ device is the tunneling magneto-resistance (referred as TMR) ratio [24], defined as $(R_H - R_L)/R_L$. An MTJ cell with high TMR ratio is desirable in order to easily distinguish between the two states $(\Delta R = R_H - R_L)$. The write current for an MTJ cell is required to be larger than the switching threshold current in order to switch the magnetization of the free layer from anti-parallel to parallel spin or vice versa.

B. *Novel STTRAM Cell Design*

Exploiting the read dominated nature of the memory access in the MBC framework, the STTRAM cell can too be optimized for read operation at the cost of write. The design space for STTRAM is constrained by the readability and writability conditions i.e., tunnel magnetoresistance (TMR) ratio and write current requirement. Given a MTJ, the choice of the access MOSFET width (W) and its wordline voltage (V_{WL}) can be used to navigate the design space of TMR and write-current. To minimize the energy dissipation, we propose to choose the energy optimal point in the $W - V_{\mathrm{WL}}$ plane. The total energy is evaluated considering the write/read current through the MTJ-transistor structure and the switching energy associated with the wordline and bitline. A key aspect of the solution is its dependence on the read-write probability. Fig. 12(a) shows two solutions corresponding to write probabilities of 0.5 and 0.1, respectively. A larger write probability means a solution with larger width and smaller V_{WL} as write has a quadratic dependence on V_{WL}. From read perspective, a solution with lower width is preferred due to less leakage power dissipation. For MBC with read-dominant access pattern, the $W - V_{\mathrm{WL}}$ configuration corresponding to equi-probable condition is not an optimal choice as it dissipates higher read energy [Fig. 12(b)]. Hence we choose the optimal energy point corresponding to low write probabilities which provides much lower read energy at the expense of increased write energy [Fig. 13(b)]. Fig. 12(b) shows 24% saving in total energy for write probability of 10^{-5} for an 8-bit 64 × 64 memory array (read access time of 400 ps).

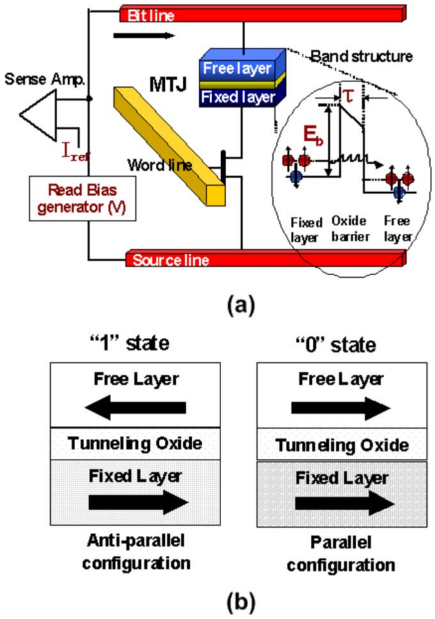

Fig. 11. (a) STTRAM cell structure. (b) Logical states of magneto tunneling junction or MTJ (anti parallel—high resistance, and parallel—low resistance).

C. *Content-Aware Application Mapping*

From the STTRAM read operation we find that a larger current flows in the circuit corresponding to read '0' than read "1." This is because resistance of state "0" is lower than state "1." Thus, in Fig. 13(a) there is a 36% difference between energy dissipated in the read "1" and read "0" operations. Write energy for "0" case dominates over the "1" case [Fig. 13(b)]. However, as a hardware reconfigurable framework, MBC is heavily biased towards read with a write probability in the range of $10^{-3} - 10^{-5}$. Hence, for applications to be mapped to the MBC framework, we conclude that if the system is biased for more read "1"s than "0"s we can have considerable energy savings with STTRAMs.

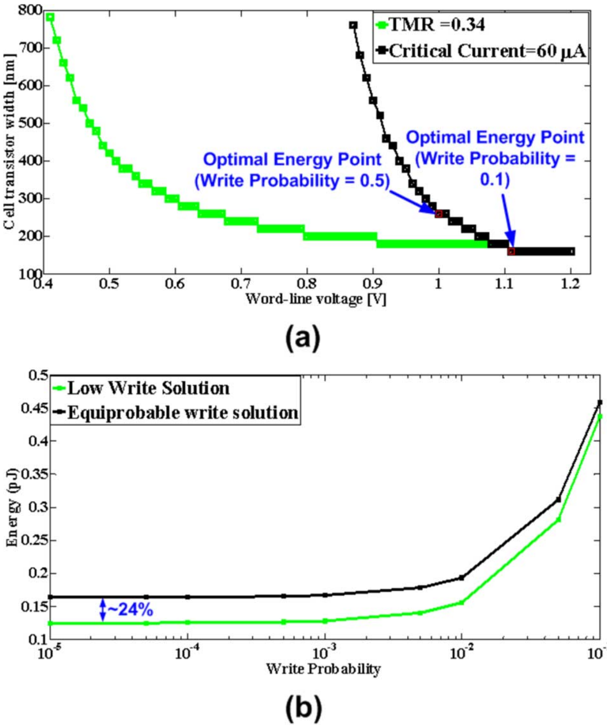

Fig. 12. (a) Design of STTRAM cell for MBC framework to achieve optimal read energy. (b) Read energy with varying write probability.

Due to higher read power during a read "0" operation, it is intended that the STTRAM array contain more logic "1" than logic "0." Considering this asymmetry, the content-aware application mapping approach was used to skew the LUTs to contain more logic "1" than logic "0" in order to harness the energy advantage, as seen in Fig. 13(a). Such a content-aware application mapping scheme therefore amplifies the energy savings by storing more logic "1" than logic "0" in the function table. A study on STTRAM array energy with varying probability of "1" storage, as shown in Fig. 13(c) points to the fact that a solution with all zero storage will result in a 16% energy access overhead compared to the case when all ones are stored in the array.

D. Improvement in EDP For STTRAM Based MBC

We have performed simulations with MTJ at 65 nm node with resistance-area product 30 Ω-$\mu\mathrm{m}^2$. The sizes of the MTJ devices have been taken as 50 × 90 nm^2 which requires approximately 60 μA of switching current assuming current density of $10^6\mathrm{A/cm}^2$. The high and low resistance states are represented by 11.1 kΩ and 6.67 kΩ, respectively. For the required STTRAM characteristics, we use the MTJ device characteristics. The resistive values for the parallel and anti-parallel states are abstracted. The abstracted resistive behavior of the MTJ is simulated in conjunction with NMOS device at 65 nm predictive technology model. The simulation for the STTRAM cell is done using HSPICE for all our measurements. To obtain the solution for varying write probabilities, first a host of simulations with the high and low resistance is performed for a range of V_{WL} and W. The solution space has to be extracted from the generated design space considering the constraints on minimum TMR and switching current. In this work we consider minimum TMR and switching current requirements of 0.34 and 60 μA, respectively. Corresponding to this extracted feasible design space of $V_{\mathrm{WL}} - W$, we evaluate the read, write, active leakage and total energy of STTRAM array. The energy evaluation considers the read and write probability ratios. The $V_{\mathrm{WL}} - W$ combination which gives the minimum energy is identified as the optimal energy solution. The read and write energies for "0" and "1" are computed for these design points.

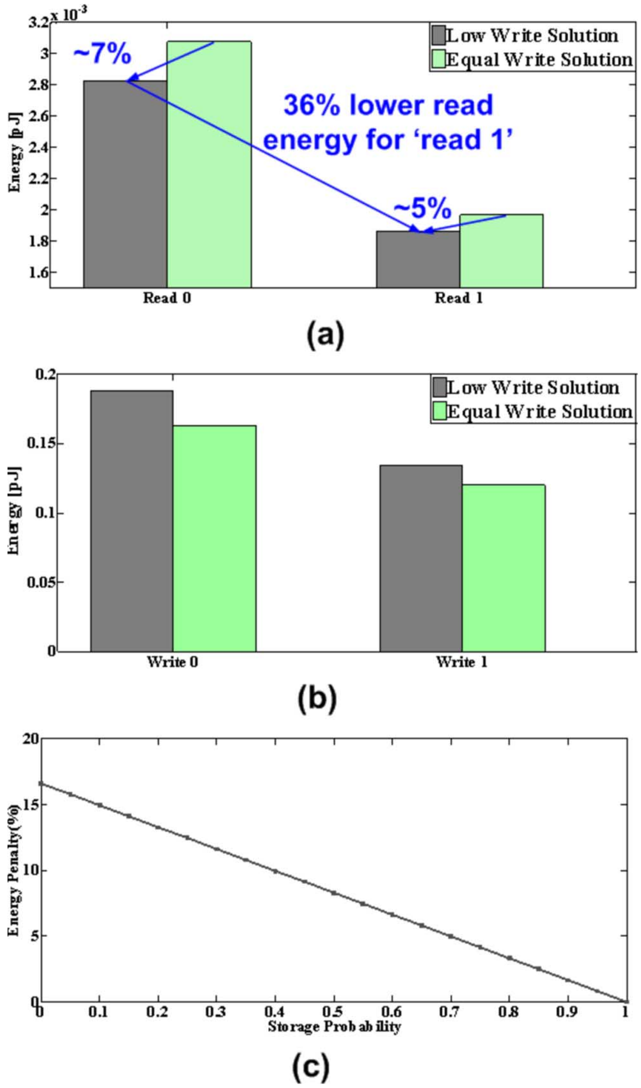

Fig. 13. (a) Read and (b) write energy for a cell storing logic "0" and "1." (c) Increase in read energy with increasing probability of storing "1."

Delay and energy requirement for the CMOS elements of the MCB were obtained through SPICE simulations using predictive models at 65-nm technology node [15]. Delay and energy estimates for the MBC framework with the read optimized STTRAM cell was obtained from the integrated automation flow described in Section III. These were then compared against that for a seven-input LUT, 10-LUT cluster FPGA model at 65 nm technology node [17]. Fig. 14(a) and (b) shows the improvement in performance and energy-delay product for STTRAM based MBC over the baseline CMOS FPGA model. As we note from Fig. 14(a), for standard benchmark circuits on an average the MBC framework improves the performance

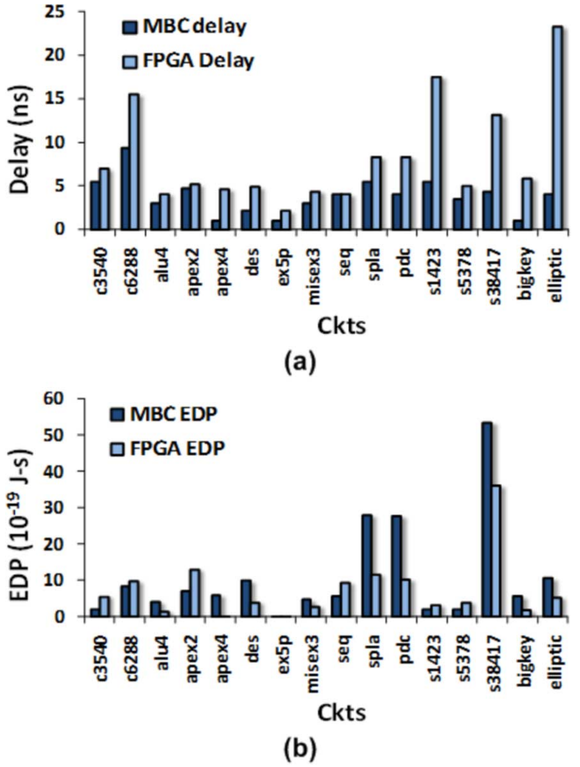

Fig. 14. Improvement in (a) delay and (b) energy delay product (EDP) for STTRAM MBC over conventional SRAM-based FPGA.

by 45.4%. Fig. 14(b) compares the EDP values between the two frameworks. The nonvolatile MBC framework achieves a 5% improvement in EDP over the CMOS FPGA framework. The performance and EDP computation includes the cell optimization for read operation. The EDP improvement is further enhanced through the content-aware mapping step which skews the LUT contents to have more logic “1” than logic “0”s. As a result of this content-aware mapping, the average EDP improvement was calculated to increase from 5% to 6.6%.

VI. Conclusion

Energy efficiency in conventional FPGA frameworks is primarily limited by the contribution from the programmable interconnects, which suffer from poor technological scalability in terms of power and performance. We have presented an circuit-architecture-software co-design approach which alleviates the interconnect overhead and achieves significant improvement in energy efficiency over conventional FPGA. The architecture uses dense 2-D memory array for computing by configuring them as multiple-input multiple-output LUTs. Exploiting the read-dominant access pattern for these memory arrays, we have presented an asymmetric memory cell design, which is preferentially optimized for read operation. The new memory cell was found to be asymmetric in read power with respect to stored logic states. We exploit this asymmetry to develop a content-aware application mapping technique, which further improves the EDP by preferentially skewing the ratio of logic “0”s and “1”s in the LUT content. We have performed two case studies, on conventional SRAM and emerging nonvolatile STTRAM technology to evaluate the effectiveness of the proposed co-design approach. Simulation results show that use of dense 2-D memory arrays for computing leads to significant improvement in interconnect overhead, which directly improves the energy efficiency for reconfigurable frameworks.

References

[1] K. Compton and S. Hauck, “Reconfigurable computing: A survey of systems and software,” *ACM Comput. Surv.*, vol. 34, no. 2, pp. 171–210, 2002.

[2] V. Betz, J. Rose, and A. Marquardt*, Architecture and CAD for Deep-Submicron FPGAs*. New York: Springer, 1999.

[3] A. P. Chandrakasan, A. Rahman, S. Das, and R. Reif, “Wiring requirement and three-dimensional integration technology for field programmable gate arrays,” *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 11, no. 1, pp. 44–54, Feb. 2003.

[4] J. Cong and S. Xu, “Technology mapping for FPGAs with embedded memory blocks,” in *Int. Symp. Field-Programmable Gate Arrays*, 1998, pp. 179–188.

[5] S. J. E. Wilton, “MAP: Heterogeneous technology mapping for area reduction in FPGAs with embedded memory arrays,” in *Int. Symp. Field-Programmable Gate Arrays*, 1998, pp. 171–178.

[6] J. Cong and S. Xu, “Performance-driven technology mapping for heterogeneous FPGAs,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 19, no. 11, pp. 1268–1281, Nov. 2000.

[7] D. Jones and D. M. Lewis, “A time-multiplexed FPGA architecture for logic evaluation,” in *Proc. Custom Integrated Circuits Conf.*, 1995, pp. 495–498.

[8] S. Paul and S. Bhunia, “MBARC: A scalable memory based reconfigurable computing framework for nanoscale devices,” in *Proc. Asia South Pacific Design Automat. Conf.*, 2008, pp. 77–82.

[9] S. Paul and S. Bhunia, “Reconfigurable computing using content addressable memory for improved performance and resource usage,” in *Design Automat. Conf.*, 2008, pp. 786–791.

[10] S. Paul, S. Chatterjee, S. Mukhopadhyay, and S. Bhunia, “Nanoscale reconfigurable computing using non-volatile 2-D STTRAM array,” in *Proc. Int. Conf. Nanotechnol. Arrays*, 1998, pp. 880–883.

[11] S. Paul, S. Chatterjee, S. Mukhopadhyay, and S. Bhunia, “A circuit-software co-design approach for improving EDP in reconfigurable frameworks,” in *Int. Conf. Comput. Aided Design*, 2009, pp. 109–112.

[12] J. Singh *et al.*, “Single ended static random access memory for low-vdd, high-speed embedded systems,” in *Int. Conf. VLSI Design*, 2009, pp. 307–312.

[13] L. Chang *et al.*, “Stable SRAM cell design for the 32 nm node and beyond,” in *Proc. Int. Symp. VLSI Technol.*, 2005, pp. 128–129.

[14] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, “Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS,” *IEEE Trans. Computer-Aided Design Integrated Circuits Syst.*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.

[15] Predictive tech. model (PTM) [Online]. Available: http://www.eas.asu.edu/~ptm

[16] Placement and Routing for FPGAs v4.3 VPR & T-VPack: Versatile Packing [Online]. Available: http://www.eecg.toronto.edu/~vaughn/vpr/vpr.html

[17] Intelligent FPGA Architecture Repository [Online]. Available: http://www.eecg.utoronto.ca/vpr/architectures/.

[18] VPR Package for FPGA power estimation [Online]. Available: http://www.ece.ubc.ca/~stevew/powermodel.html.

[19] G. Karypis *et al.*, “Multilevel hypergraph partitioning: Applications in VLSI domain,” *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 7, no. 1, pp. 69–79, Mar. 1999.

[20] J. Cong and Y. Ding, “FlowMap: An optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 13, no. 1, pp. 1–12, Jan. 1994.

[21] M. Hosomi *et al.*, “A novel non-volatile nemory with spin torque transfer magnetization switching: Spin-RAM,” in *Proc. Int. Electron Devices Meeting*, 2005, pp. 459–462.

[22] T. Kawahara *et al.*, “2 Mb Spin-Transfer Torque RAM (SPRAM) with bit-by-bit bidirectional current write and parallelizing-direction current read,” *Int. J. Solid-State Circuits*, vol. 43, no. 1, pp. 109–120, 2007.

[23] J. Hayakawa *et al.*, “Current-driven magnetization switching in CoFeB/MgO/CoFeB magnetic tunnel junctions,” *Jpn. J. Appl. Phys.*, vol. 44, no. 41, pp. L1267–L1270, 2005.

[24] S. Tehrani *et al.*, "Magnetoresistive random access memory using magnetic tunnel junctions," *Proc. IEEE*, vol. 91, no. 5, pp. 703–714, May 2003.

[25] ITRS 2007: Emerging research devices [Online]. Available: http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_ERD.pdf

[26] V. K. Prasanna, "Energy efficient computations on FPGAs," *J. Supercomput.*, vol. 32, no. 2, pp. 139–162, 2005.

[27] F. Li, Y. Lin, L. He, and J. Cong, "Low-power FPGA using pre-defined Dual-Vdd/Dual-Vt fabrics," in *Int. Symp. Field Programmable Gate Arrays*, 2004, pp. 42–50.

[28] R. Williams, "How we found the missing memristor," *IEEE Spectrum*, vol. 45, no. 12, pp. 28–35, Dec. 2008.

[29] W. Y. Cho *et al.*, "A 0.18- $\mu\mathrm{m}$ 3.0 V 64-Mb non-volatile phase transition random access memory (PRAM)," *IEEE J. Solid State Circuits*, vol. 40, no. 1, pp. 293–300, 2005.

[30] D. Lewis and G. Lemieux, "Circuit design of routing switches," in *Proc. Int. Symp. Field Programmable Gate Arrays*, 2002, pp. 19–28.

[31] Cacti 5.1 Tech. Rep. [Online]. Available: http://www.hpl.hp.com/techreports/2008/HPL-2008-20.pdf

[32] K. Zhang, U. Bhattacharya, Z. Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Y. Wang, B. Zheng, and M. Bohr, "A 3-GHz 70-Mb SRAM in 65-nm CMOS technology with integrated column-based dynamic power supply," *IEEE J. Solid State Circuits*, vol. 41, no. 1, pp. 146–151, Jan. 2006.

Somnath Paul (S'07) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2005. He is currently working towards the Ph.D. degree in computer engineering at Case Western Reserve University, Cleveland, OH.

He was a Design Engineer with Advanced Micro Devices, Bangalore, India. He has also held internship positions at Intel and Qualcomm. His research interests include development of novel hardware frameworks for reconfigurable architectures and hardware/software co-design for yield improvement in nanoscale technologies.

Subho Chatterjee (S'08) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 2004, the M.S. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 2009, and the M.S. (E.E) degree from Georgia Institute of Technology, Atlanta, in 2010. He is currently working toward the Ph.D. degree at the School of Electrical and Computer Engineering at the Georgia Institute of Technology.

His research interests include nonvolatile memories, low power and variation tolerant design techniques and 3 D integrated circuit design. He worked as a research intern with AMD Research and Advanced Development Labs during 2010.

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

He is with the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, GA, as an Assistant Professor. Prior to joining Georgia Institute of Technology, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member. His research interests include analysis and design of low-power and robust circuits in nanometer technologies and 3-D circuits and systems. He has (co)-authored over 100 papers in refereed journals and conferences and been awarded five U.S. patents.

Dr. Mukhopadhyay received the National Science Foundation CAREER Award in 2011, IBM Faculty Partnership Award in 2009 and 2010, SRC Inventor Recognition Award in 2008, SRC Technical Excellence Award in 2005, and the IBM Ph.D. Fellowship Award for 2004–2005. He has received the Best in Session Award at 2005 SRC TECNCON, and Best Paper Awards at the 2003 IEEE Nano and 2004 International Conference on Computer Design.

Swarup Bhunia (S'00–M'05–SM'09) received the B.E. (Hons.) from Jadavpur University, Kolkata, India, and the M.Tech. degree from the Indian Institute of Technology (IIT), Kharagpur, India. He received the Ph.D. degree from Purdue University, West Lafayette, IN, in 2005.

Currently, he is an Associate Professor of Electrical Engineering and Computer Science at Case Western Reserve University, Cleveland, OH. He has over ten years of research and development experience with over 100 publications in peer-reviewed journals and premier conferences in the area of VLSI design, CAD and test techniques. His research interests include low power and robust design, hardware security and protection, adaptive nanocomputing and novel test methodologies. He has worked in the semiconductor industry on RTL synthesis, verification, and low power design for about three years.

Dr. Bhunia received National Science Foundation (NSF) career development award (2011), Semiconductor Research Corporation (SRC) technical excellence award (2005), best paper award in International Conference on Computer Design (ICCD 2004), best paper award in Latin American Test Workshop (LATW 2003), and best paper nomination in Asia and South Pacific Design Automation Conference (ASP-DAC 2006) and in Hardware Oriented Test and Security (HOST 2010), nomination for John S. Diekhoff Award, Case Western Reserve University (2010) and SRC Inventor Recognition Award (2009). He has served as a Guest Editor of IEEE Design & Test of Computers (2010), in the editorial board of Journal of Low Power Electronics (JOLPE) and in the technical program committee of Design Automation and Test in Europe (DATE 2006–2010), Hardware Oriented Trust and Security Symposium (HOST 2008–2010), IEEE/IFIP International Conference on VLSI (VLSI SOC 2008), Test Technology Educational Program (TTEP 2006–2008), International Symposium on Low Power Electronics and Design (ISLPED 2007–2008), IEEE/ACM Symposium on Nanoscale Architectures (NANOARCH 2007–2010), IEEE International Conference on VLSI (ISVLSI 2008–2010), International Conference of VLSI Design as a track Chair (2010) and in the program committee of International Online Test Symposium (IOLTS 2005). He has given tutorials on low-power and robust design and test in premier conference including International Test Conferences (ITC 2009), VLSI Test Symposium (VTS 2010), and Design Automation and Test in Europe (DATE 2009).