Robust Low-Power Reconfigurable Computing with a Variation-Aware Preferential Design Approach

Somnath Paul
Intel Corporation
Hillsboro, OR, USA
somnath.paul@intel.com

Saibal Mukhopadhyay
Georgia Institute of Tech.
Atlanta, GA, USA
saibal@ece.gatech.edu

Swarup Bhunia
Case Western Reserve Univ.
Cleveland, OH, USA
skb21@case.edu

***Abstract*—Reconfigurable hardware platforms, such as Field Programmable Gate Arrays (FPGA), are being increasingly used in diverse embedded applications. These platforms often use high-density memory array, which suffer from variation-induced parametric failures. Such failures lead to incorrect operation and hence, loss in output quality for many signal processing applications. In this paper, we propose a preferential design approach at both application mapping and circuit level, which can significantly improve output quality as well as energy efficiency for signal processing applications under large parameter variations. The proposed mapping process considers the reliability map of a memory array and maps the important operations with respect to output quality to more reliable memory blocks under performance constraint. At circuit level, we exploit the read-dominant memory access pattern in reconfigurable platforms to skew the memory cells for better read stability leading to improved quality. Such an architecture/circuit co-design approach can also tolerate increased failure rate at low operating voltage, thus facilitating low-power operation.**

***Index Terms*—Reconfigurable computing, FPGA, MBC, spatio-temporal mapping, reliability, low power, preferential design.**

I. Introduction

Memory has been the fundamental building block for most of the popular reconfigurable hardware platforms including the commercial Field Programmable Gate Arrays (FPGA). Conventional FPGA fabric represents logic functions using 'k-input 1-output' one-dimensional Lookup Tables (LUT) in a purely spatial computing framework [2-3]. On the other end of the spectrum, a time-multiplexed spatio-temporal Memory Based Computing (MBC) model has been investigated that uses dense two-dimensional memory array to map large multi-input multi-output LUTs [6-7]. With increasing process variations at nanoscaled technology nodes, reliable operation for such memory based reconfigurable 7computing frameworks emerges as a major concern. Variation may potentially cause memory access failures or flipping of stored data during read-out [5], which leads to incorrect execution of a mapped application. Moreover, in order to reduce the power requirement, memory core is conventionally operated at lower supply voltages. Although it helps minimize the active and leakage power consumption, read and access failure probabilities increase significantly at low operating voltages [5] leading to reduced reliability of operation.

In order to compensate for variation-induced failures in memory, statistical design [5] along with built-in redundancy (which enables post-fabrication repair) has emerged as a popular design choice. However, improving the yield for all sections of a large embedded memory array can be extremely challenging due to within-die variation induced distribution of reliability across different memory sections. Therefore, a preferred solution for MBC would be to utilize different sections of a large memory block with different reliability in a way that minimizes the impact on performance [9]. This can be achieved by exploiting the nature of the mapped applications. We note that computations in digital signal processing (DSP) applications can typically be classified into two categories: a) *significant components:* failure to compute them correctly leads to large loss in output quality; b) *less significant components:* any failure in these components cause considerably less impact in output quality. For example, in case of Discrete Cosine Transform (DCT), it is observed that 85% or more of the input image energy is contained in the first 20 of total 64 coefficients [8]. In order to achieve graceful degradation in quality of service (QoS) for most DSP applications, we propose a *reliability map* aware application mapping methodology for MBC frameworks. The proposed methodology maps the critical computations to more reliable sections of the memory under delay constraint, and hence achieves maximum QoS under variations.

In addition, we note that MBC operation is dominated by read, while write only occurs occasionally during reconfiguration. Exploiting the read-dominant memory access pattern in MBC, we propose a preferential memory cell sizing approach that makes the memory more robust to read and access failures [10]. The resultant decrease in write stability can be addressed by a column based lowering of the cell supply during reconfiguration. In particular, the paper makes the following contributions:

1) *For a memory based reconfigurable framework, it proposes a reliability map aware application mapping process for DSP applications which can significantly reduce variation-induced output quality degradation.*
2) *It proposes a circuit-level preferential design approach for memory cells which improves stability of read operation over write. This exploits the read-dominant access pattern in MBC to improve output quality with modest impact on write performance.*

The remainder of the paper is organized as follows. Section II illustrates the effect of parameter variations on QoS for DSP applications. Section III presents the preferential mapping algorithm. Section IV presents the preferential memory cell design and Section V concludes the paper.

II. Effect of Parameter Variations on QoS

Parametric variation in nanoscale technologies can lead to incorrect execution of applications mapped to the MBC framework. Such errors in execution may arise from any one of the following sources:

- *Failure in memory:* In scaled technologies, due to parameter variations, memory cells may undergo *read*, *write* and *access* failures [5]. While both *access* and *write* failures can be addressed by increased latency during read and write, *read* failures are difficult to address and require changing operating conditions such as memory cell supply.
- *Failure in Control Unit:* The number of bits stored in the control memory is orders of magnitude less than that stored in the LUT.

978-1-4799-2153-9/14/$31.00 ©2014 IEEE

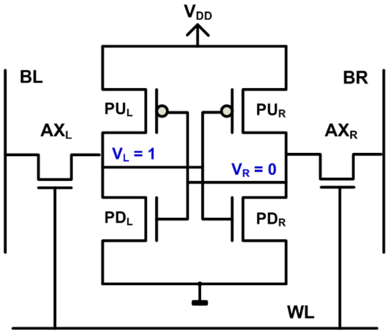

Fig. 1. Schematic for a conventional 6-T SRAM cell.

Thus in order to avoid any variation induced failures, the control memory may be conservatively designed.

- *Failure in programmable interconnects:* Variation in the interconnect dimensions and the programmable switches can lead to an increased delay in the routing network of the MBC framework. However, both these delay variations can be addressed by lowering the operating frequency for the framework.

Memory failures due to degraded cell stability is therefore most difficult to address. In this section, therefore, we study the impact of memory failures on the performance of DSP applications mapped to a memory based reconfigurable computing framework.

Variation in process parameters has emerged as a major design challenge at both circuit and architecture level [1]. Such variations can be either systematic (i.e. all devices in a die experience similar variations) or random, caused by phenomenon such as Random Dopant Fluctuations (RDF) or Line Edge Roughness (LER). The random variations can cause mismatch between adjacent transistors in a SRAM cell, which may eventually lead to a failure of the cell itself [5]. To evaluate the effect of memory failures on output QoS for an application mapped as LUTs in the MBC framework, we have considered a memory model consisting of a 16KB memory array, divided into 128 blocks, each with 1024 cells. The memory cells in each block is organized into 32 rows (N_{ROW}) with 32 cells (N_{COL}) in each column. The number of redundant columns in each block (N_{RC}) is 2. We also consider pulsed wordline and bitline isolation architecture for reducing bitline swing and a nominal V_{dd} of 0.9V. We performed simulations with this memory using HSPICE for PTM 45nm LP models [11]. We considered the following scenarios:

1) Nominal, slow and fast corners considering 15% inter-die variation ($3\sigma/\mu$ being considered as a percentage definition, where μ and σ are the mean and standard deviation of the threshold voltage (V_t) distribution).
2) Cells across the blocks were assumed to suffer from spatially correlated variation with standard deviation $\sigma_{dV_{t}sys} = 50mV$.
3) V_t fluctuations (due to random intra-die variation) in the 6-T SRAM cell were modeled as six independent Gaussian variables with with $\mu = 0$ and $\sigma_{dV_{t0}rand} = 50mV$ for minimum sized transistor. For transistors of larger sizes, the distribution was effectively determined by

$$\sigma_{dV_{t}rand} = \sigma_{dV_{t0}rand} * \sqrt{(\frac{L_{min}}{L})(\frac{W_{min}}{W})} \tag{1}$$

From the Monte Carlo (MC) simulations performed on the cells inside each block, we noted the parameters V_{read}, V_{trip}, T_{access} and T_{write} for each cell. These parameters are defined as follows:

- V_{read}: The positive voltage that is generated during a read operation at the node V_R storing '0' (refer to Fig. 1).
- V_{trip}: The trip point for the inverter $PU_L - PD_L$ (refer to Fig. 1) that holds a logic value of '1' at its output.
- T_{write}: This is the time required to pull down the node V_L below the trip voltage for the inverter $PU_R - PD_R$.
- T_{access}: It is the time required to generate a specified voltage difference between two bitlines once the wordline is activated.

These parameters are commonly used to characterize the vulnerability of memory cells to parametric failures [5]. As pointed out in [16-17], a "weak" memory cell can undergo a read/write or access failure under supply voltage variation, high temperature or coupling noise. In order to quantify the vulnerability of individual memory blocks, we introduce block level reliability metrics obtained by collating V_{read}, V_{trip}, T_{access} and T_{write} parameters from individual memory cells. The metrics are:

- Indicator for Read Stability denoted as I($V_{trip} - V_{read}$)
- Indicator for Write-ability denoted as I(T_{write})
- Indicator for Access-ability denoted as I(T_{access})

In general the indicator $I(x)$ where $x \equiv V_{trip} - V_{read}$, T_{write} or T_{access} for a memory block can be derived by:

- Classifying the cells of the block into separate bins based on the value of parameter x
- Calculating a weighted average of the cells in each bin:

$$I(x) = \frac{\sum_{i=1}^{N} n_i * w_i}{\sum_{i=1}^{N} w_i} \tag{2}$$

In equation (2), w_i denotes the weight and n_i denotes the number of cells in the i_{th} bin. For our simulations, we have divided the cells in each block into 5 bins ($N = 5$) with $w_i = 2^{-i}$ values. The range of $x \equiv V_{trip} - V_{read}$ values for the bins are: i) $x \leq 0$, ii) $0 < x \leq 100mV$, iii) $100mV < x \leq 150mV$, iv) $150mV < x \leq 200mV$, and v) $x > 200mV$. Choice for the number of bins and the range of x values were primarily driven by the distribution of the $V_{trip} - V_{read}$ obtained in our simulations. For low supply voltages, considering a parametric variation scenario with $\sigma_{dVt_{sys}} = \sigma_{dVt0_{rand}} = 50mV$, $V_{trip} - V_{read}$ values for the 'weak' cells were mostly centered around a value of 50mV. Hence, the 2^{nd} bin was chosen for $0 < x \leq 100mV$. The weight assigned to each bin is primarily determined by the operating condition of the MBC framework and the relative vulnerability to failure for cells in each bin. Ideally the weights for the 'weak' cells should be exponentially large in order to distinguish them from cells with $x > 100mV$.

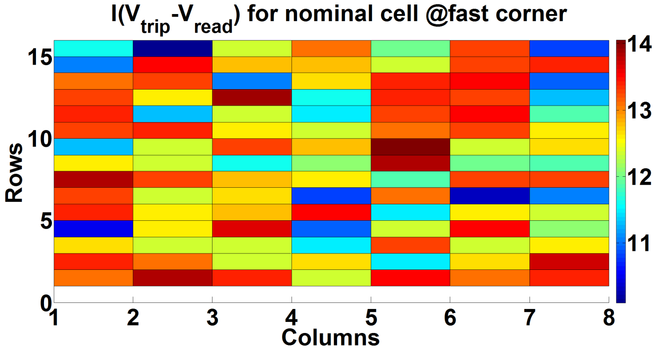

Fig. 2. Variation in $I(V_{trip} - V_{read})$ for the memory blocks at fast corner.

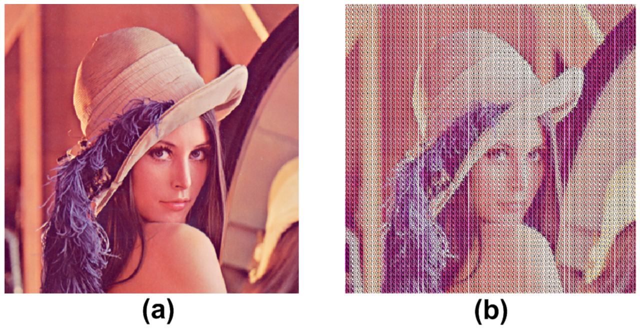

Fig. 3. (a) Original image used as input to DCT; and (b) image after inverse DCT, where 1D-DCT operation is performed using the MBC framework ($Avg_{PSNRwrst} = 28.49dB$).

A block with a larger value of $I(V_{trip} - V_{read})$ is more prone to read disturb failures under increased environmental stress (i.e. reduced voltage and or higher temperature). Fig. 2 shows the inter-block distribution of $I(V_{trip} - V_{read})$ at the fast corner. In order to see the effect of these failures on the final DCT output, LUTs for the DCT operation were first randomly mapped to the blocks of our memory model. Note that for a reliability distribution similar to Fig. 2, the effect of a read or access failure on the DCT output is observed when the following conditions are simultaneously satisfied:

- The number of failures in a block must be greater than N_{RC}, so that the failure cannot be repaired.
- Inputs to the DCT module activate the LUT location which suffers from read or access failure.

Fig. 3 shows the output image quality ($Avg_{PSNRwrst}$) for random mapping of the LUTs to the memory blocks. As observed from Fig. 3(b), due to random assignment, the final image quality suffers considerable degradation with an average worst-case PSNR of 28.49dB.

To alleviate degradation in output quality due to variation-induced memory failures in the MBC framework, we propose a three-step solution that involves joint circuit/architecture level optimizations. As we show later, the proposed approach can provide significant improvement in parametric yield under variations. As summarized in Fig. 4, the major steps in the proposed co-design approach are:

- Post-fabrication characterization of the memory and generation of the reliability map to store $I(V_{trip} - V_{read})$, $I(T_{write})$ and $I(T_{access})$ values for the memory blocks.
- A preferential application mapping approach that maps the critical computations to more reliable memory blocks under delay constraint.
- A preferential (skewed) design of memory cell using transistor sizing that increases the read stability at the expense of write failures for the read-dominated MBC framework.

Note that while the preferential mapping approach is more suitable for applications, which can be partitioned into critical and less-critical computations, the sizing based preferential memory design benefits all applications that are mapped to the MBC framework.

III. Preferential mapping

Many signal processing applications (such as DCT, FIR filtering) consist of critical and less critical computations [8]. A mapping algorithm, which: a) partitions the computations into critical and less-critical bins; and then b) maps the critical computations to more reliable sections of the memory, can effectively achieve substantial performance improvement over a random mapping approach.

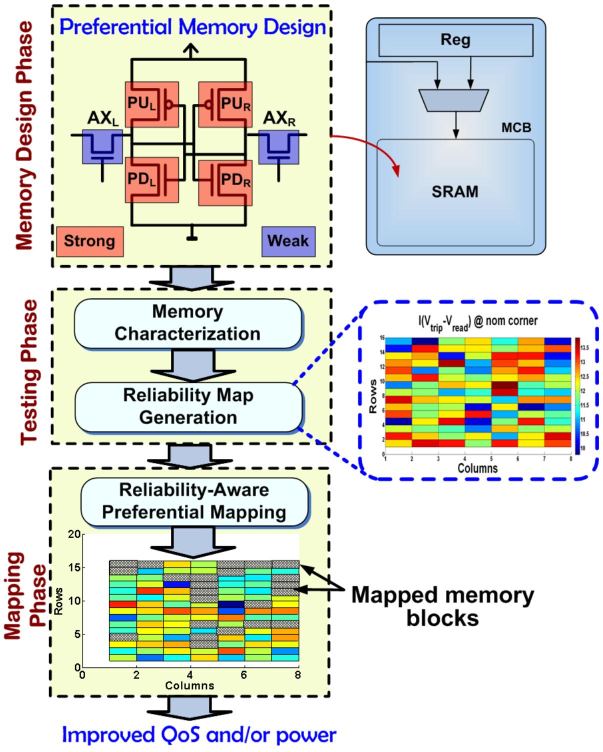

Fig. 4. Major steps in the proposed methodology for robust and low-power operation in MBC.

A. Post-Silicon Reliability Map Generation

A number of techniques [15-17] have already been proposed to generate post-silicon reliability map for embedded memories in order to cope with process variation induced parametric yield loss. These techniques either use March Test [14] to detect parametric failures or attempt to directly measure the read/write margin for memory cells [17]. Since the motivation for reliability map generation is only to identify the relative variability of the memory blocks, an array based technique can suffice for the preferential mapping approach. For an array based characterization technique with block size of 1KB, the total memory requirement to store the reliability map for a function table of size 16KB is only 64 bits. This follows from the observation that the $I(V_{trip} - V_{read})$ values can be encoded into 4 bits without significant quantization error (refer to Fig. 2).

B. Impact of Preferential Mapping on Output Quality

We propose a heuristic-based preferential mapping approach that can significantly improve the output quality of the target application under parameter variations. If c_i(i=1··P) denotes the contribution of the i^{th} computation to the overall output quality and $r_j(j = 1 \cdot \cdot N,\ N \geq P)$ denotes the reliability measure of the j^{th} memory block, then the mapping heuristic should attempt to maximize $\sum_{i=1}^{P} c_i r_j$. By reliability measure for the j^{th} memory block, we mean $r_j = \frac{1}{I_j(x)}$. A simple minded approach is to sort $C = \{c_i\}$ and $R = \{r_j\}$ in descending order of their values, and then to assign the computations in the order of their contribution to unassigned memory blocks with highest value of r_j. As an example, in the DCT application, first adder stage and the precomputer blocks are assigned to minimum $I(V_{trip} - V_{read})$ blocks for more reliable operation. This heuristic achieves about 4dB improvement in PSNR on average over a random mapping policy.

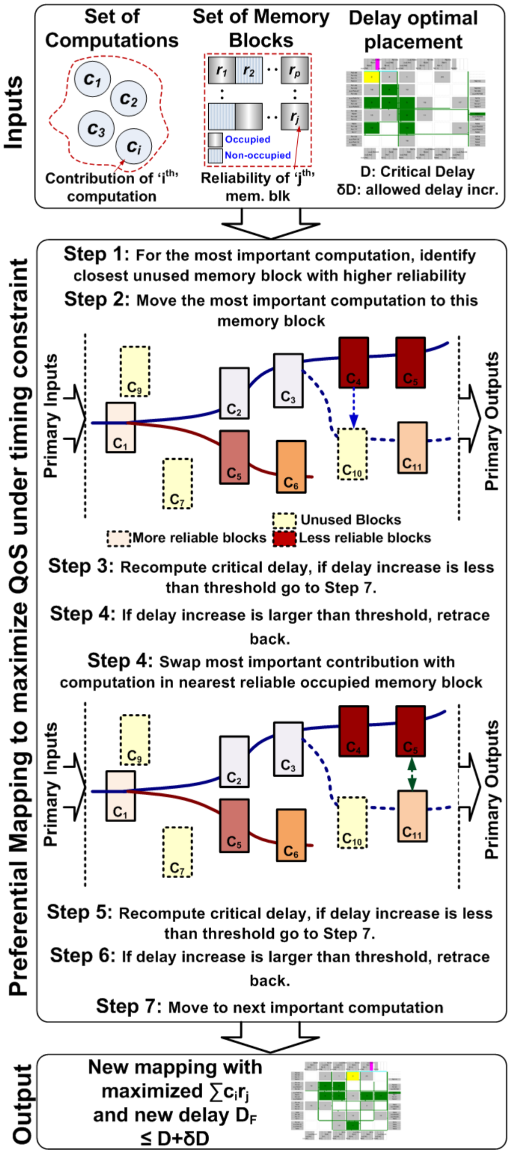
Inputs
Set of Computations
C_1
C_2
C_3
C_i
Contribution of 'ith' computation
Set of Memory Blocks
r_1 r_2 r_p
r_j
Occupied
Non-occupied
Reliability of 'jth' mem. blk
Delay optimal placement
D: Critical Delay
δD: allowed delay incr.
Preferential Mapping to maximize QoS under timing constraint
Step 1: For the most important computation, identify closest unused memory block with higher reliability
Step 2: Move the most important computation to this memory block
Primary Inputs
Primary Outputs
C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_9 C_{10} C_{11}
More reliable blocks
Unused Blocks
Less reliable blocks
Step 3: Recompute critical delay, if delay increase is less than threshold go to Step 7.
Step 4: If delay increase is larger than threshold, retrace back.
Step 4: Swap most important contribution with computation in nearest reliable occupied memory block
Step 5: Recompute critical delay, if delay increase is less than threshold go to Step 7.
Step 6: If delay increase is larger than threshold, retrace back.
Step 7: Move to next important computation
Output
New mapping with maximized $\sum c_i r_j$ and new delay $D_F \le D+\delta D$

Fig. 5. Flowchart showing the major steps for the proposed heuristic-based preferential mapping approach under timing constraint.

C. Preferential Mapping Algorithm Under Delay Constraint

The simple-minded mapping approach discussed above is suitable for improving the output quality when applications are mapped to the memory array in a single memory-based computing block (MCB), a configurable building block for MBC. However, in general, applications will be mapped to multiple MCBs which are interconnected using a programmable interconnect network similar to FPGAs. A preferential mapping approach that does not consider any delay constraint would only attempt to provide the best output QoS but may cause increased delay overhead due to larger distance between the computing elements in the critical path. Thus, in order to achieve the optimal mapping (in terms of reliability) under a delay constraint, resource allocation should be part of the placement and routing step. We have developed a heuristic-based post-processing of the delay-optimal routed netlist that maximizes QoS without violating the delay constraint.

For example, consider the mapping of the 'c880' circuit from ISCAS'85 benchmark suite to the MBC framework. Critical delay for the 'c880' circuit was estimated before and after preferential mapping was applied. For 'c880' which is a 8-bit arithmetic logic unit (ALU), higher outputs were considered more performance critical compared to the lower bits. Based on this consideration, computing blocks producing the higher bits were ranked higher than the ones producing the lower bits. After partitioning, the partitions were packed into 20 computing blocks (MCBs). These MCBs were then placed and routed using VPR [18]. After routing with minimum number of tracks option in VPR, the benchmark occupied a 6x6 array and had a critical delay of 4.93ns at 45nm technology node. The variation aware preferential mapping however provided an alternate placement, primarily because of the fact that it does not consider the effect of critical paths in the benchmark. This placement expectedly results in a higher critical delay (5.52ns) after routing using the VPR toolset. From this exercise we note that an alternate variation aware mapping approach is essential which would result in minimal increase of the optimal delay reported from VPR. To achieve such a solution we started from the delay optimal placement reported from VPR and used minimum perturbation in the placed design so as to move the critical computations to more reliable memory blocks. As opposed to the original preferential mapping approach which results in a (12%) increase in delay, the new solution only leads to only (2%) increase in critical delay. The compromise is in the total value of $I(V_{trip} - V_{read})$ for the allocated blocks. For the original preferential mapping approach it is 261 and for the timing driven mapping approach it is 271 (increase of 4%).

Fig. 5 shows the timing-driven variability-aware preferential mapping process. It starts with ranking the computations in order of their criticality and the memory blocks in order of their reliability measures (defined as r_i for the i^{th} memory block). In descending order of criticality, each computation in the routed design is then considered for reassignment if it is mapped to a memory block with unacceptable r_i. The reassignment step first looks for an unused block with higher r_i and at the smallest Euclidean distance d_i from its current position. The impact on timing is calculated for each trial using incremental timing analysis. If the resulting placement exceeds the delay constraint, we attempt to swap the current computation with a less critical one which is assigned to a more reliable memory block. As before, computing elements at the smallest Euclidean distance are given higher priority and timing for the new placement is checked so that it does not violate the delay constraint. The major steps of the proposed heuristic is illustrated in Fig. 5.

For mapping the DCT application, we consider a MBC framework with 4Kb of memory at each MCB. With this specification, the partitioning and packing algorithms were able to map the DCT application to 24 computing elements (MCBs), which, after place and route using VPR toolset [18] occupied a 12 × 12 array. It is to be noted that the large array required to map the DCT application is due to the large IO requirement for the DCT application (64-bit input and 96-bit output for mapping 1D DCT). We do not assume time-multiplexing of the IOs. Fig. 6(a) shows the original routed design from VPR and the output from the timing-driven preferential mapping heuristic is presented in Fig. 5(b). The modified placement has the same delay (4.78ns) as the delay-optimal placement. However, the output PSNR improves from a value of *28.49dB* (Fig. 3) to *31.72dB* (Fig. 6).

IV. Preferential memory design

In SRAM cell design, optimization of read and write stability presents contradictory requirements. However, we can leverage on

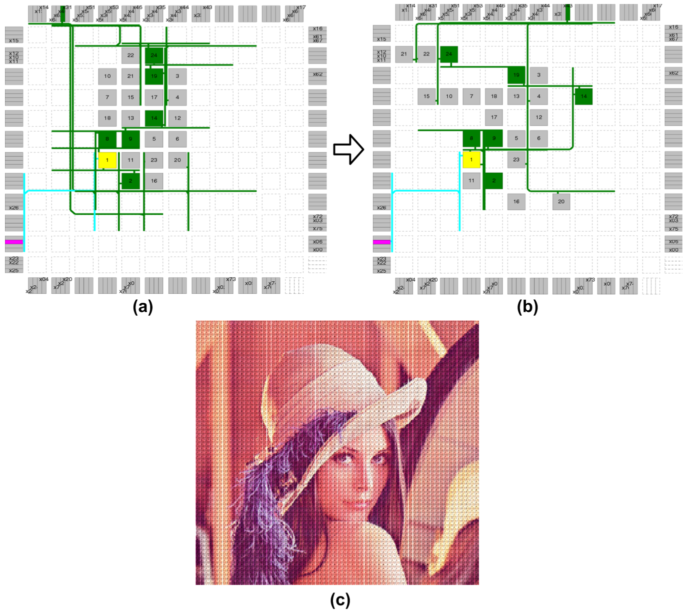

Fig. 6. a) Delay optimal placement of the DCT blocks obtained from VPR [18]; b) final placement after the proposed preferential mapping under timing constraint; and c) final image after the mapping has a PSNR of 31.72dB.

the read-dominated access pattern in the MBC framework to trade off read stability with write-ability of memory cells.

A. Preferential sizing of 6-T SRAM

The criteria followed for preferential sizing of the 6-T SRAM cell are: i) minimize the read and access failure probabilities; ii) minimize cell area increase. The cell area estimation methodology is the same as followed in [5]. In order to derive the read, write and access failures, we note that while V_{read} and V_{trip} follow normal distributions [5], inverse of T_{write} and T_{access} can be approximated with normal distributions as well [4]. For our simulations, the probabilities were therefore obtained by:

- Fitting normal distributions to V_{read} and V_{trip} followed by the calculation of $P_{RF} = P(X \equiv (V_{trip} - V_{read}) > 0) = 1 - \phi_X(0)$, where ϕ_X denotes the cumulative distribution function (cdf) of the random variable $X \equiv V_{trip} - V_{read}$.
- Fitting normal distributions to the inverse of T_{write} followed by the calculation of probability of write failure $(P_{WF}) = P(\frac{1}{T_{write}} < \frac{1}{T_{thw}}) = \phi_{\frac{1}{T_{write}}}(\frac{1}{T_{thw}})$.
- Fitting normal distributions to the inverse of T_{access} followed by the calculation of probability of access failure $(P_{AF}) = P(\frac{1}{T_{access}} < \frac{1}{T_{tha}}) = \phi_{\frac{1}{T_{access}}}(\frac{1}{T_{tha}})$.

T_{tha} and T_{thw} are the maximum T_{access} and T_{write} that can be tolerated by the system and was set to 1ns and 2ns, respectively for the DSP operations under consideration. V_{ddnom} was set at 0.9V. Since the joint probability of the read (P_{RF}), write (P_{WF}) and the access failures (P_{AF}) are negligible [5], we have neglected these joint probability values in estimating the total failure probability for a single cell. From our simulations, we observed that:

- Increasing W_{PUP} increases V_{trip} and reduces P_{RF}.
- Increasing W_{PDN} decreases V_{read} and thus reduces P_{RF}.
- Decreasing W_{AX} reduces P_{RF}.
- Increase in P_{AF} due to weakening of W_{AX} can be compensated by increasing W_{PDN}.
- First two optimizations increase T_{write} and P_{WF}.

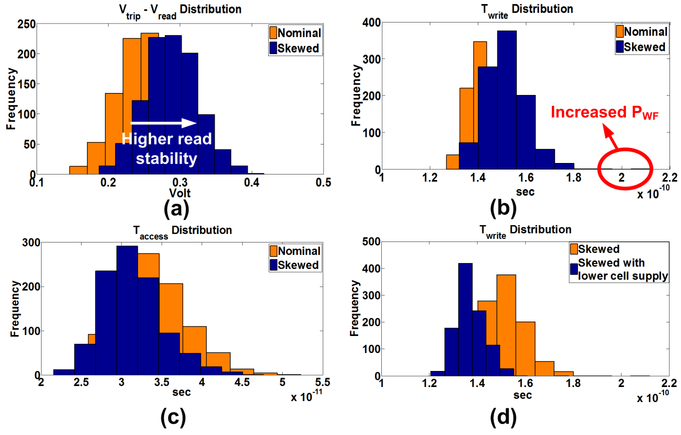

Fig. 7. Distribution of: a) $V_{read} - V_{trip}$, b) T_{write} and c) T_{access} for the nominal and the skewed memory cell. d) Shows the T_{write} distribution for cell with column-based lowering of cell supply voltage.

The old and new value for the transistor sizes are given in Table I. With L_{min} = 45nm, the increase in area per cell was calculated to be 2.5% according to the formula provided in [5].

B. Compensation for Increased Write Failure Probability

From Fig. 7(a) and Fig. 7(c), we note that the skewed memory design achieves better tolerance to read and access failures at the cost of higher write failures (refer to Fig. 7(b)). Since a higher wordline voltage degrades the half-select stability of the cells in the same row [19], we propose to use a column-based lowering of cell supply to achieve better write-ability of the skewed memory cell. Although such a scheme incurs area overhead due to complicated power grid and routing, it achieves nearly 10X reduction in single bit failure by creating a 100mV voltage differential between the wordline and cell supply. In our simulation framework, supply voltage for the memory cell was reduced to 0.7V to improve T_{write} (Fig. 7(d)). Table II shows the overall failure rate for the nominal design, the skewed design as well as skewed design with lower cell supply voltage for write. A skewed design with lower cell supply for write reduces the total cell failure probability (P_F) by factor of 112.

C. Impact of cell sizing on output quality

The memory model as described in Section III was simulated with the skewed cell design for the same inter-die and intra-die variations. Fig. 8(a) illustrates the fact that skewing leads to redistribution of the cells in the five bins for a given memory block. Since a heavy penalty is associated with each failing cell, skewing leads to an improvement

TABLE I
TRANSISTOR WIDTHS FOR NOMINAL AND SKEWED MEMORY CELL

Cell	W_{PUP}(nm)	W_{AX}(nm)	W_{PDN}(nm)
Nominal	100	150	200
Skewed	125	125	220

TABLE II
CELL FAILURE PROBABILITIES FOR $V_{ddnom} = 0.9V$

Cell	P_{RF}	P_{WF}	P_{AF}	P_F
Nominal	3.28e-8	6.66e-16	2.92e-14	3.28e-8
Skewed	2.93e-10	2.77e-12	2.22e-14	2.95e-10
Skewed w/ lower WR Vdd	2.93e-10	3.33e-16	2.22e-14	2.93e-10

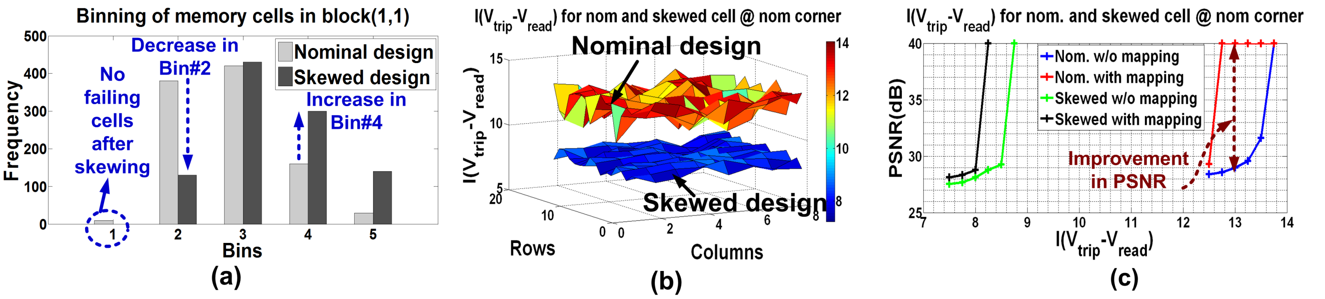

Fig. 8. a) Redistribution of memory cells into bins before and after skewing; b) skewed design achieves better $I(V_{trip} - V_{read})$ for all blocks; and (c) comparison of PSNR values.

of $I(V_{trip} - V_{read})$ values across all the blocks in the memory (Fig. 8(b)). Fig. 8(c) illustrates the PSNR improvement achieved through the proposed memory design and delay-aware preferential mapping approach. Following points may be noted from Fig. 8(c).

- Preferential mapping for the baseline (not skewed) memory design can achieve significant improvement in PSNR for a range of $I(V_{trip} - V_{read})$ as high as 1.75. Considering that a cell moving from bin #1 to bin #2 reduces $I(V_{trip} - V_{read})$ by 0.25, this is equivalent to tolerating five more read failures in the memory block without increasing N_{RC}.
- Due to significantly smaller values of $I(V_{trip} - V_{read})$, the skewed design did not suffer from any degradation in output quality in our simulations. However, if the tolerance for $I(V_{trip} - V_{read})_{max}$ is reduced to smaller value (which translates to considering a higher variation for the skewed design), a degradation of the PSNR values is observed at the output. Preferential mapping can again be applied to the latter case to improve performance.
- Improvement in output PSNR due to preferential mapping after skewing is comparatively smaller than the mapping applied to the original design. The reason being that skewing leads to large improvement in read stability for all memory cells, thereby reducing improvement with the mapping approach. The skewed memory cells however occupy larger area compared to the nominal design and experience increased probability of write failures. The preferential mapping approach, therefore, provides a complementary solution to circuit-level optimization with much lower overhead.

V. Conclusion

We have presented a circuit-architecture co-design approach for tolerating variation-induced failures in memory based reconfigurable computing frameworks. Process variations create a distribution of memory reliability across different blocks of a large memory array resulting in a reliability map. Using the reliability map, the proposed approach performs timing-aware preferential application mapping that maps most significant computations in more reliable memory blocks. Considering the read-dominant access pattern, we have also presented a preferential memory cell design approach using transistor sizing that improves reliability of read over write. The impact on write stability during occasional write operation can be addressed by existing approaches such as lower cell supply during write. Using DCT, a common DSP application, we show that in presence of process variation, the proposed approach can significantly improve QoS, while enabling low power operation.

References

[1] S. Borkar, T. Karnik and V. De, "Design and reliability challenges in nanometer technologies", *Design Automation Conference*, 2004.

[2] J. Cong and S. Xu, "Performance-Driven Technology Mapping for Heterogeneous FPGAs", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 11, 2000.

[3] S.J.E. Wilton, "SMAP: Heterogeneous Technology Mapping for Area Reduction in FPGAs with Embedded Memory Arrays", *International Symposium on Field-Programmable Gate Arrays*, 1998.

[4] K. Agarwal and S. Nassif, "Statistical Analysis of SRAM Cell Stability", *Design Automation Conference*, 2006.

[5] S. Mukhopadhyay, H. Mahmoodi and K. Roy, "Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Dec 2005.

[6] D. Jones and D.M. Lewis, "A time-multiplexed FPGA architecture for logic emulation", *Custom Integrated Circuits Conference*, 1995.

[7] S. Paul and S. Bhunia, "A Scalable Memory-Based Reconfigurable Computing Framework for Nanoscale Crossbar", *IEEE Transactions on Nanotechnology* Vol. 9, No. 2, 2010.

[8] N. Banerjee, G. Karakonstantis and K. Roy, "Process variation tolerant low power DCT architecture", *Design Automation and Test in Europe*, 2007.

[9] S. Paul, S. Mukhopadhyay, and S. Bhunia, "A variation-aware preferential design approach for memory based reconfigurable computing", *Intl. Conf. on Computer Aided Design*, 2009.

[10] S. Paul, S. Chatterjee, S. Mukhopadhyay and S. Bhunia, Energy-Efficient Reconfigurable Computing Using a Circuit-Architecture-Software Co-Design Approach, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2011.

[11] Predictive Technology Models (PTM). [Online] http://ptm.asu.edu/modelcard/LP/45nm_LP.pm

[12] L. Chang et al., "An 8T-SRAM for Variability Tolerance and Low-Voltage Operation in High-Performance Caches", *IEEE Journal of Solid State Circuits*, 2008.

[13] K. Kim, J.J. Kim and C.T.Chuang, "Asymmetrical SRAM Cells with Enhanced Read and Write Margins", *International Symposium on VLSI Technology, Systems, and Applications*, 2007.

[14] M.L. Bushnell and V.D. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", *Springer*, 2000.

[15] N.N. Mojumder, S. Mukhopadhyay, J.J. Kim, C.T. Chuang and K. Roy, "Design and Analysis of a Self-Repairing SRAM with On-Chip Monitor and Compensation Circuitry", *VLSI Test Symposium*, 2008.

[16] D.M. Kwai et al., "Detection of SRAM Cell Stability by Lowering Array Supply Voltage", *Asian Test Symposium*, 2000.

[17] A. Pavlov, M. Sachdev and M.J.P. de Gyvez, "Weak Cell Detection in Deep-Submicron SRAMs: A Programmable Detection Technique", *Journal of Solid State Circuits*, Vol. 41, No. 10, 2006.

[18] Full CAD Flow for Heterogeneous FPGAs Version 5: [Online; accessed 31st Dec, 2013] http://www.eecg.utoronto.ca/vpr/

[19] S. Mukhopadhyay, R. Rao, J.J. Kim and C.T. Chuang, "Capacitive Coupling Based Transient Negative Bit-line Voltage (Tran-NBL) Scheme for Improving Write-ability of SRAM Design in Nanometer Technologies", *International Symposium on Circuits and Systems*, 2008.