

E-RoC: Embedded RAIDs-on-Chip for Low Power Distributed Dynamically Managed Reliable Memories

Luis Angel D. Bathen
Center for Embedded Computer Systems
University of California, Irvine
Irvine CA, USA
lbathen@uci.edu

Nikil D. Dutt
Center for Embedded Computer Systems
University of California, Irvine
Irvine CA, USA
dutt@uci.edu

Abstract— *The dual effects of larger die sizes and technology scaling, combined with aggressive voltage scaling for power reduction, increase the error rates for on-chip memories. Traditional on-chip memory reliability techniques (e.g., ECC) incur significant power and performance overheads. In this paper, we propose a low-power-and-performance-overhead Embedded RAID (E-RAID) strategy and present Embedded RAIDs-on-Chip (E-RoC), a distributed dynamically managed reliable memory subsystem. E-RoC achieves reliability through redundancy by optimizing RAID-like policies tuned for on-chip distributed memories. We achieve on-chip reliability of memories through the use of distributed dynamic scratch pad allocatable memories (DSPAMs) and their allocation policies. We exploit aggressive voltage scaling to reduce power consumption overheads due to parallel DSPAM accesses, and rely on the E-RoC manager to automatically handle any resulting voltage-scaling-induced errors. Our experimental results on multimedia benchmarks show that E-RoC’s fully distributed redundant reliable memory subsystem reduces power consumption by up to 85% and latency up to 61% over traditional reliability approaches that use parity/cyclic hybrids for error checking and correction.*

I. INTRODUCTION

Embedded system designs need to satisfy multiple constraints including power, performance and reliability. Continued technology scaling and larger die sizes, coupled with the increasing amounts of on-chip memory, make memories highly vulnerable to the threat of soft-errors and process variation induced errors [16][17][18]. Aggressive voltage scaling for power reduction further increases the error rates of on-chip memories [3][13]. Traditional memory reliability techniques utilize ECC, or ECC-duplication hybrids, and incur significant power and performance overheads. This problem is exacerbated by the emergence of on-chip distributed memory subsystems, as evidenced by the trend of chip multiprocessor systems (e.g., IBM Cell [32], Intel’s Multi-core [33], Teraflops Research [34], and Tiler Tile-Gx [35]), where cores can talk to multiple on-chip memories using different access/coherency protocols and a variety of communication infrastructures (e.g., bus matrix, P2P, NoCs, etc.).

As technology scales, system failure rates due to radiation-induced transient errors continues to be a major concern for embedded system designers [2]. Memories are most vulnerable to soft-errors since the total area of the die is dominated by memory cells. This problem worsens for chip-multiprocessor platforms that have even larger amounts of on-chip memory.

Moreover, to reduce power consumption, designers employ techniques such as aggressive voltage scaling, which exponentially increases the impact of process variation on memory cells [3]. Voltage scaling reduces the capacitance that keeps the charge in a single cell, therefore affecting its vulnerability to low energy alpha particles, or cosmic rays [17]. Process variation is random in nature as it depends on many factors such as environmental (temperature, voltage), physical (mask imperfections, wear-out mechanisms), and in-die physical variations (layout, gate dimension). As process technology reaches its limits, failures due to process variation are rapidly increasing [7][12][18]. The probability of failure in SRAM technology is exponentially proportional to the decrease in voltage. Unlike soft-errors, which are transient in nature, process variation induced errors are permanent. Although aggressive voltage scaling increases the rate of failures, power savings can still be achieved by designing fault tolerant systems [19]. Efforts in reliable memory systems have focused on the design of error correction based memories, where data accesses are guarded by ECC mechanisms [4][6][8][23][24], replication based mechanisms [5][9][10][11], as well as process variation aware designs [3][7][12]. Note that some of these techniques may combine two or more different schemes to guarantee reliability of the memory subsystem. At the system level, Redundant Array of Inexpensive Disks (RAID) systems [14] have been very successful in providing reliable data storage for the storage/distributed systems domain, and have been used from simple low cost servers to large scale storage area networks [15], including operating environments that need to guarantee 24/7 uptime under heavy I/O loads.

This paper makes several new contributions. Since distributed on-chip memory hierarchies are becoming common in chip-multiprocessor systems, we adapt and tune the traditional notion of RAID to define *Embedded RAID (E-RAID)* and *Embedded RAIDs-on-Chip (E-RoC)*, a distributed dynamically managed reliable memory subsystem. Among the key concepts introduced are: the notion of reliability via redundancy using an E-RAID system; a set of *E-RAID levels* that are optimized for use in embedded SoCs; and the concept of *distributed dynamic scratch pad allocatable memories (DSPAMs)* and their allocation policies. We exploit aggressive voltage scaling to reduce power consumption overheads due to parallel DSPAM accesses. The resulting voltage-scale-induced errors that appear in the memories are handled by the E-RAID

This research was partially supported by NSF Variability Expeditions Award CCF-1029783, and SFS/NSF Grant No. 0723955

policies. We present the first proof-of-concept **E-RoC Manager** that exploits these ideas and a set of architectural designs by which E-RoC based systems can be configured.

II. CUSTOMIZING EMBEDDED RAID (E-RAIDS)

The goal of a traditional RAID system in storage systems is to guarantee the uptime of the system. In case a disk goes bad, the remaining disks are used to 1) serve data requests despite the failed disk, and 2) on disk replacement, rebuild the RAID system. In Embedded RAID (E-RAID), the notion of a failed SPM does exist; however, we cannot take the system offline, replace the SPM, and rebuild the E-RAID. Because of this, E-RAID levels need to be given a different purpose. The goal of an E-RAID is to guarantee the validity of the data stored in the E-RAID. Because of this, we must modify traditional RAID levels, and customize/optimize them for the use in embedded SoCs. Like traditional RAID systems, E-RAID systems will benefit from parallel reads/writes to multiple memories. This requires support for such a model, with the attendant need for on-chip parity calculation and checking. Since RAID system parity can be computed by a simple XOR, E-RAIDs incur less performance overheads compared with any of the ECC/hybrid schemes previously proposed [4][5][6][10]. E-RAID systems may incur power overheads due to the extra accesses to memory systems necessary to access the duplicate data during reads and writes. Aggressive voltage scaling can be utilized to offset this additional power consumption, therefore efficiently reducing the penalty of the extra memory accesses. Any errors resulting from aggressive voltage scaling of memories can then be automatically handled by the E-RAID system.

III. EROC: EMBEDDED RAIDSON-CHIP

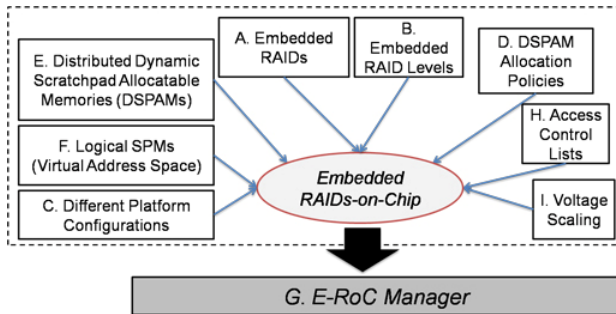


Figure 1. Concept of Embedded RAIDSON-CHIP

Figure 1 outlines the concept of Embedded RAIDSON-CHIP (E-RoC). E-RoC is composed of eight mutually dependent components (A through I) that are used to create a customized E-RoC Manager for the specific settings of each component. In the following we briefly describe each E-RoC component.

A. Embedded RAIDSON (E-RAIDSON)

E-RAIDSON exploit the idea that aggressive voltage scaling of memories significantly reduces power consumption, but increases the error rate in the memories. This intentional increase in the error rates can be automatically handled by E-RoC’s RAID-like built-in error resiliency mechanisms. Thus in the E-RAID context, we use customized, reliable E-RAID levels to automatically handle the errors generated by

aggressive voltage scaling of the memories. Moreover, as a side effect, transient errors are also automatically handled.

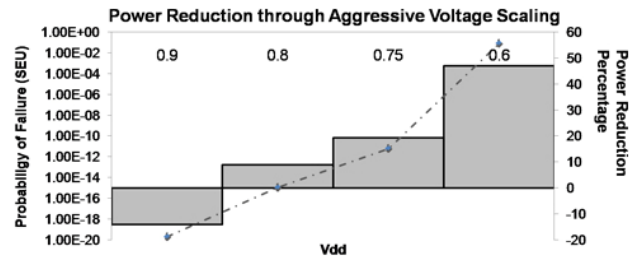


Figure 2. Power Reduction for E-RAID 0+1 Level

To illustrate the potential for power reduction using E-RAID, consider an E-RAID configuration consisting of eight 512B/8bit width SPMSON voltage-scaled with data striping and replication (referred to as “E-RAID 0+1 Level”). Although the potential number of SPMSON accessed per read/write transaction is 8 times that of a single SPMSON running at normal Vdd (1.1) and width of 32bits, **we observe up to 46% power reduction with aggressive voltage scaling**. Figure 2 shows progressive power savings due to aggressive voltage scaling in an E-RAID 0+1 configuration consisting of 65nm SPMSON (gray bars). On the left axis we see that as voltage increases, the probability of failure increases exponentially as well (dotted line).

B. E-RAIDSON Levels

We now present some sample E-RAIDSON levels to illustrate how traditional RAIDSON levels are adapted in the on-chip context (other E-RAIDSON levels are detailed in our TR[35]). On error detection, a re-refresh (write) is issue when data is fetched from lower levels of memory hierarchy.

L	Read Protocol:
1	Read block A1 from DSPAM _x
2	Read block A2 from DSPAM _y
3	IF (A1==A2) DATA=A1, RETURN CHANNEL_OK
4	ELSE RETURN SLV_ERR
Write Protocol:	
6	Write block A to DSPAM _x
7	Write block A to DSPAM _y
8	RETURN CHANNEL_OK

Figure 3. E-RAID 1 Level

1) E-RAID 1

E-RAID 1 follows the same redundancy idea of traditional RAID 1, where we keep two copies of each block in the E-RAID, with each block being a 32bit word. Figure 3 shows the E-RAID 1 level. On a read (Lines 1-4), E-RoC will fetch both copies of block A, compare them, and return the data if the comparison was successful. On an error the master¹ will be forced to fetch the data from off-chip memory, thereby paying the penalty of a main memory access. This policy uses the idea of duplicates to guarantee data integrity.

¹ The terms *master* and *CPU* will be used interchangeably throughout the paper since E-RoC’s services may benefit both CPUs and hardware modules requiring access to a reliable memory system.

L	Read Protocol:
1	Read block A1 from SPM _x
2	Read block A2 from SPM _y
3	IF (A1==A2) DATA=A1, RETURN CHANNEL_OK
4	ELSE
5	Read parity P from SPM _p
6	IF (A1 XOR P == R) DATA=A1, RETURN CHANNEL_OK
7	ELSE IF (A2 XOR P == R) DATA=A2, RETURN CHANNEL_OK
8	ELSE RETURN SLV_ERR
Write Protocol:	
9	Write block A to SPM _x
10	Write block A to SPM _y
11	Write block A XOR R to SPM _p
12	RETURN CHANNEL_OK

Figure 4. E-RAID 1 + P Level

2) E-RAID 1+P

E-RAID 1 + P mirrors traditional RAID 4 by keeping two copies of the data, and dedicating an SPM for parity (P). The parity calculations are done via simple XOR operations. At start up, E-RoC will randomly generate a large prime number (R) that is used to calculate parities on the block. Figure 4 shows the E-RAID 1 + P level. On a read (Lines 1-8), both copies of block A are read and compared, in the case the comparison fails, the parity P is read, and used to check which of the two copies is valid. This is done by simple XORing the block and the parity to reconstruct the random prime number R . E-RAID 1 + P reduces off-chip memory accesses due to erroneous data by introducing an extra parity computation. This policy avoids the extra off-chip access present in E-RAID 1 in the case when the two copies present do not match, and the parity is used to reconstruct the data.

3) NO E-RAID

In order to fully utilize the entire SPM space, E-RoC provides the capability of using the NO E-RAID mode so that masters may utilize internal DSPAM space. This mode is very useful when the criticality of the data is minimal, e.g., for pixel data in multimedia applications, where errors may be tolerated.

C. Multi-platform Support

We target our designs for multiprocessor embedded SoCs where each processing core may need to configure an E-RAID system to handle its executing task. As shown in Figure 5, E-RoC can be customized for different architectural platforms. **Since this is the first piece of work introducing E-RoC, our goal is to show the use of E-RoC on familiar platforms.** Thus we consider a simple homogeneous CMP architecture, consisting of multiple processing cores (CPUs), instruction cache, distributed SPMs, a DMA engine to facilitate the data transfers among the various SPMs, and a shared bus topology.

D. Dynamic Scratch Pad Allocatable Memories (DSPAMs)

We introduce the notion of distributed **Dynamic Scratch Pad Allocatable Memories (DSPAMs)**. These memories differ from SPMs in that although they are still part of the memory space, they are only accessed by/through the E-RoC module, and are aggressively voltage scaled. Their physical space is dynamically allocated/de-allocated by the E-RoC module using a variety of platform configurations. For instance, Figure 5(a) shows a platform configuration consisting of an E-RoC module connected to the on-chip bus, with the E-RoC module responsible for maintaining E-RAID systems for each CPU.

Each E-RAID data request will be routed by the bus to the E-RoC slave, which in turn sends slave requests to each of the respective E-RAID memories it manages. This model suffers from delays due to on-chip bus traffic. The benefits of a shared model however, are the flexibility in managing the available DSPAMs. Figure 5(b) shows a second configuration which consists of a dedicated DSPAM bus where each E-RAID request is routed to the E-RoC module, which then issues read requests to each of the DSPAMs. Unlike the model in Figure 5(a), this model does not suffer from delays due to the extra traffic on the main on-chip bus. Figure 5(c) shows a third configuration, which consists of a single stand-alone E-RoC module that has point-to-point connectivity to each of its managed DSPAMs. This point-to-point connectivity further reduces the delay due to on-chip data transfers, since each E-RoC request to each of its managed DSPAMs can be processed instantaneously. One drawback from this model is that the available DSPAM resources are pre-defined, and therefore not as flexible as the previous two models. Similarly, other configurations can be developed as detailed in our TR [35].

E. DSPAM Allocation Policies

Since DSPAMs are extremely valuable resources, the DSPAM allocation algorithm must be optimized to efficiently allocate their space for each E-RAID system. In traditional storage systems, an entire disk is dedicated to a RAID system. If we were to fully allocate a DSPAM to an E-RAID system, the performance degradation would be too large. To account for this overhead, we introduce two ideas: (i) a **virtual DSPAM address space**, which is a unified global address space viewed by the external world, and (ii) the **dynamic allocation of this virtualized address space**. The virtual address space makes it easy for the compiler to allocate/de-allocate space as it would if it was targeting a regular SPM, and the dynamic support enables us to configure E-RAID systems of various sizes at run-time, thereby providing the necessary memory space for each task [35].

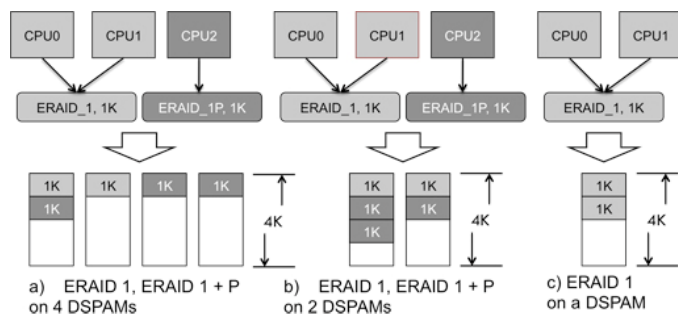


Figure 6. DSPAM Allocation Policies

Figure 6 shows three different examples where block based allocation successfully configured E-RAID systems for two different processors. CPU0 requested a 1K E-RAID 1 system, which will be shared with CPU1, and CPU2 requested an E-RAID 1+P 1K system. The first two designs with (a) four and (b) two 4KB DSPAMs respectively successfully created the E-RAID systems. The third design (c) with a single 4KB DSPAM has successfully allocated space for the E-RAID 1 1K system, and returned a SLV_ERR for the allocation of the second E-

RAID 1+P system request since there are no more resources available for its creation. The first allocation policy we explore follows the Next Fit model, where DSPAM memory space is split into k blocks ($k \in \{64, 128, 256, 512\}$ bytes). For each DSPAM we keep a free list (single bit), and we allocate on the next best-fit basis, while maintaining a circular list of free blocks per DSPAM. The number of SPMs searched in parallel for allocation depends on the E-RAID level (e.g. E-RAID 1 requires 2 DSPAMs). We walk through DSPAMs in a round-robin mode in order to fairly distribute data across DSPAMs. Like traditional RAID single disk failures, unusable DSPAMs are removed from the list of managed DSPAMs and background E-RAID re-mapping is done.

F. Logical SPMs and Virtual Address Space

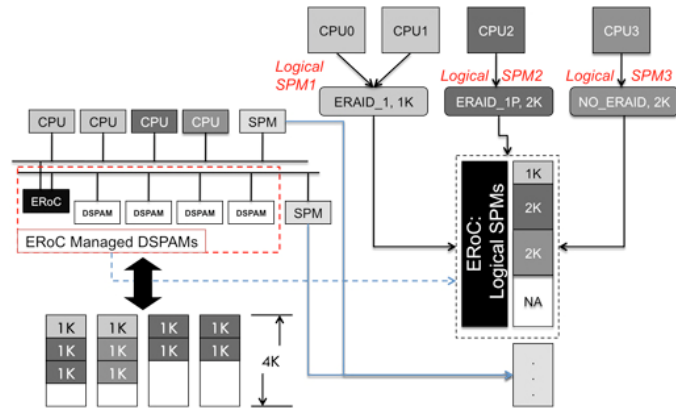


Figure 7. Virtual Address Space

Because data placement onto the E-RAID systems is still left to the compiler, we must make it possible for the compiler to configure E-RAIDs on demand, as well as manage the data efficiently. **We introduce the idea of Logical SPMs via virtual address spaces in E-RAID systems.** One of the main benefits of E-RoC is that it abstracts the complexity of the E-RAID system, and presents the compiler with a simplified view of the address space via a logical SPM. The compiler can create an E-RAID system, and regardless of the E-RAID level being enforced, the compiler will see the E-RAID system as a logical memory mapped DSPAM. All transfers between the CPU(s), main memory and the E-RAID system will follow the same process as in a system with pure SPMs and DMA support. As an example, Figure 7 shows a diagram of a 4 CPU CMP with SPM support, and an E-RoC manager with 4 DSPAMs. As shown, CPUs 0 and 1 configured a 1K E-RAID 1 shared system, CPU2 configured a 2K E-RAID 1 + P system, and CPU3 has no E-RAID, but wants to access the DSPAM space. All accesses to the E-RAID systems are transparent as the CPUs see their E-RAIDs as logical SPMs. The virtual address space presented to the outside world is shown in the dark dashed lined box. The main difference between DSPAMs and logical SPMs is that DSPAMs are physical voltage scaled SPMs visible to the E-RoC manager, while logical SPMs are addressable memory spaces visible to the compiler/OS.

G. E-RoC Manager Architecture

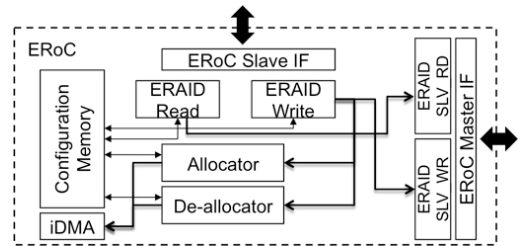


Figure 8. E-RoC Manager Architecture

Figure 8 shows the architecture of the E-RoC manager, consisting of a master and a slave interface. The slave interface handles incoming E-RAID requests, and the master interface issues read/write requests to each DSPAM managed. Each master in the system has a dedicated and restricted memory space in the E-RoC module. This mechanism prevents other masters from overwriting configuration information for another master's E-RAID system. The E-RAID Read/Write modules handle read/write requests depending on the policies being handled. If the transaction is a configuration request, then the validity of the request is checked. Given the allocation policy, on an E-RAID create request, the allocator searches for space across the various DSPAMs; if there is enough space, the E-RAID system is created. In the case the master desires the E-RAID system to be built using data from some memory space, the allocator uses its internal DMA engine (iDMA) to fetch the data and store it in the new E-RAID system. On a de-allocation request, the E-RAID is offloaded onto main memory (when desired) and the blocks occupied by the E-RAID are freed.

H. Access Control Lists (ACL) and I. Voltage Scaling

ACLs are used to guarantee that no unauthorized masters gain access to a given memory region (E-RAID). Aggressive voltage scaling is utilized to reduce power consumption due to DSPAM accesses. A more detailed description of the E-RoC organization and architecture is in our technical report [35].

IV. RELATED WORK

Much of the state of the art work in reliable memory systems focuses on caches. Makhzan et al. [7] propose the idea of exploiting error maps to correct faulty cells on the main cache. Kim et al. [25] introduce an area efficient ECC cache protection mechanism. Lee et al. [2] propose the idea of using partitioned caches to protect critical data that is mapped onto an ECC protected cache, with non-critical data mapped onto a regular cache. Zhang et al. [10] introduce a small fully associative cache into the memory hierarchy where data is replicated; the duplicates are used to detect and correct errors. In the event of process variation errors, techniques such as technology mapping and cache redundancy are used [5]. ECC/replication hybrids have also been composed in both the cache domain [10] and the SPM domain [11]. Since we target SPM based systems, the closest piece of work to E-RoC is the work done in [11], which uses parity calculations to check the validity of data; in the case of an error, an extra copy of the data is fetched, assuming no errors in the extra. Data mapped onto the E-RAID can follow the same process as proposed in

[21, 26-28] with a few minor modifications, mainly E-RAID configuration requests before data is mapped onto the E-RAID. **Because E-RoC relies on data duplication and simple comparisons/XORs to check/correct data, we have seen great improvements not only in power but also in performance**, as most ECC/replication approaches require expensive ECC parity calculations on every transaction. Similarly, E-RoC's space overhead is very similar to existing data replication techniques as it keeps at most two copies of the data and the parity (E-RAID 1+P). Most SPM approaches [26][27][28] focus on single SPM management through static analysis and profiling information. **Ours is the first piece of work that looks at the idea of dynamic allocation in distributed SPMs, as well as their use from a reliability perspective.**

V. EXPERIMENTAL RESULTS

1) Experimental Setup

We implemented our E-RAID ideas in the E-RoC DSPAM manager. The E-RoC concept has been implemented in SystemC [22] and embedded into our SystemC based modeling framework [23], which allows us to estimate both power and performance for the entire system. Since we deal with SPM based systems, we compare our work with two existing approaches: (i) a standard ECC based approach that verifies and corrects the data (labeled *ECC*), and (ii) the duplication with parity work [11] (labeled *DUP*), each memory in these configurations was voltage scaled ($V_{dd} = 0.65$). We mapped three applications from the MediaBench II benchmark (JPEG encoder, JPEG decoder and H.263) and investigated the power and performance overheads (and savings) for several CMP configurations and data mappings. Detailed results are in [35].

2) Normalized Performance and Power Consumption

Figure 9 shows the normalized performance and power

consumption for a CMP with 8 cores and 8 4KB SPMs. For this experiment we configured the E-RoC manager as shown in Figure 5 (a). The base case is a CMP with no voltage scaling applied to the SPMs. The standard SPM case outperforms all other configurations because a) the ECC/parity check overheads incurred in the ECC/DUP cases and b) the backend address translation as well as fetching the data from up to three different DSPAMs (in case data needs to be reconstructed) performed by E-RAID. Our E-RAID systems outperforms the ECC/DUP configurations by up to 14% and consumes up to 80% less power than the standard high voltage SPM CMP and up to 85% less power than the ECC/DUP approaches.

3) Selective Data Partitioning

E-RoC is well suited for approaches that can selectively partition data into critical/vulnerable and non-critical data [2]. Such approaches allow us to further reduce power consumption and improve performance as E-RoC offers the ability to choose low power policies such as NO E-RAID, where non-critical (e.g., image pixel) data may be mapped. This is illustrated in Figure 9 (labeled: **Partial** in the graphs), where by selectively creating E-RAID systems (i.e., mapping critical data to E-RAID1 space and non-critical to NO E-RAID space), performance can be improved by up to 5%, and power consumption can be further reduced by up to 15%.

4) Choosing the Right Platform Configuration

The platform configuration affects the system's performance and power consumption footprint. Figure 10 shows the normalized performance and power consumption for different CMP configurations, where C{1-3} refer to configurations (a-c) from Figure 5. As expected, performance was greatly improved when migrating from configuration C1 (shared bus) to C2 (dual bus) due to less bus contention. Configuration C3 (stand alone E-RoC), was able to further improve performance

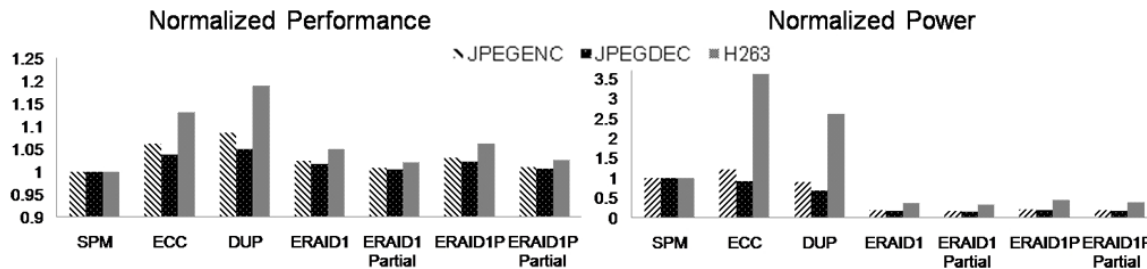


Figure 9. Normalized performance (left) and power consumption (right) for three different benchmarks.

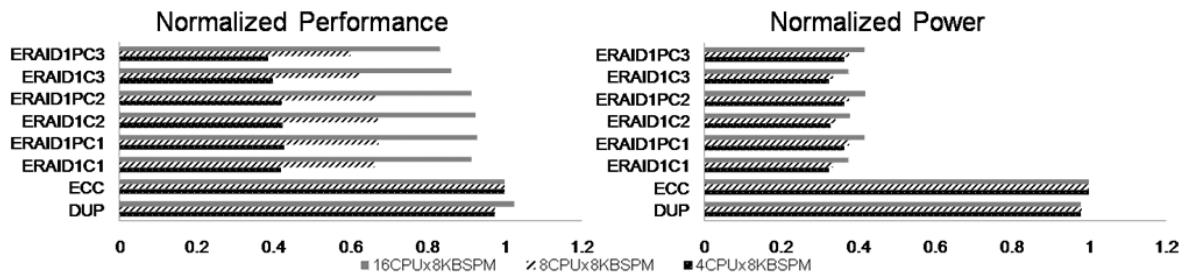


Figure 10. Normalized performance (left) and power consumption (right) for a pipelined JPEG Encoder using different platform configurations with E-RoC manager support as shown in Figure 5.

by 10% with respect to C1. Power consumption remains within a 2% difference for all three configurations (C1-3). This pipelined implementation of JPEG showed up to 61% latency reduction and 67% power reduction when compared to standard ECC/DUP approaches.

5) Effects of Voltage Scaling on E-RoC

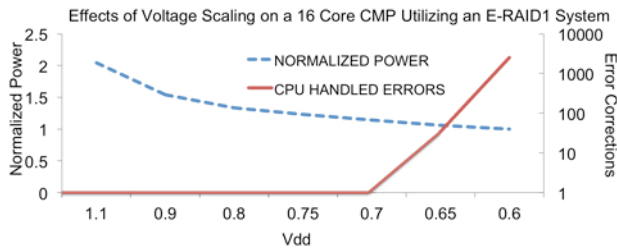


Figure 11. Effects of Voltage Scaling on E-RoC

Figure 11 shows the effects of voltage scaling on a 16 Core CMP with a total of 16 concurrently managed E-RAID systems. As we scale down voltage, power consumption is indeed being reduced (dashed line), while the error rate skyrockets (straight line). This behavior confirms our initial observations from Section 2 that aggressive voltage scaling of SPMs increases the memory error rate (handled by our E-RAID systems) but reduces the power consumption.

VI. CONCLUSION

In this paper we introduced the notions of Embedded RAID (E-RAID) and Embedded RAIDs-on-Chip (E-RoC), a distributed dynamically managed reliable memory subsystem. Among the key concepts introduced are: the notion of reliability via redundancy using an E-RAID system; a set of E-RAID levels that are optimized for use in embedded SoCs; and the concept of distributed dynamic scratch pad allocatable memories (DSPAMs) and their allocation policies. We exploited aggressive voltage scaling to reduce power consumption overheads due to parallel DSPAM accesses. We defined the first proof-of-concept E-RoC manager that exploits these ideas. We presented a set of architectural designs by which E-RoC based systems can be configured. Our experimental results on multimedia benchmarks show that E-RoC's *fully distributed redundant reliable memory subsystem* can attain up to 85% in power consumption reduction, and up to 61% latency reduction due to error checks/corrections. Since E-RoC is a first of its kind piece of work, there are many opportunities for further optimization that we are currently exploring: E-RAID allocation policies might lead to DSPAM space fragmentation. Similarly, the shared bus model drives E-RoC performance down, thereby motivating the use of more complex communication fabrics (i.e., Bus Matrix and NoCs).

REFERENCES

- [1] P. Shivakumar et al., "Modeling the Effect of Technology Trends on the Soft Error Rate of Comb. Logic." DSN '02, Bethesda, MD, June, 2002.
- [2] K. Lee et al., "Mitigating soft error failures for multimedia applications by selective data protection." CASES '06, Seoul, Korea, Oct., 2006
- [3] A. Sasan et al., "A fault tolerant cache arch. for sub 500mV operation: resizable data composer cache.," CASES '09

- [4] H. Vergos et al., "Efficient Fault Tolerant Cache Memory Design", Microprocessing/Microprogramming Journal, vol.41, pp.153-169, 1995.
- [5] H. Lucente, et. al., "Memory System Reliability Improvement Through Associative Cache Redundancy", Proc. of CICC'90, pp.19.6.1-19.6.4
- [6] V. Papirla et al., "Energy-aware error control coding for Flash memories," In Proc. of DAC '09, San Francisco, California, July, 2009
- [7] A. Sasan et al., "Limits of Voltage Scaling for Caches Utilizing Fault Tolerant Techniques", ICCD 2007.
- [8] S. Ghosh et al., "Reducing Power Consumption in Memory ECC Checkers", ITC-2004, Charlotte, N.C., October 2004
- [9] W. Zhang et al., "Enhancing data cache reliability by the addition of a small fully-associative replication cache.," Proceedings. of ICS '04
- [10] W. Zhang et al., "ICR: in-cache replication for enhancing data cache reliability". Proceedings of DSN, 2003
- [11] F. Li et al., "Improving scratch-pad memory reliability through compiler-guided data block duplication." In Proceedings of ICCAD '05
- [12] M. Makhzan et al., "Process Variation Aware Cache for Aggressive Voltage-Frequency Scaling," Proceedings of DATE '09
- [13] F. Kurdahi et al., "Low-Power Multimedia System Design by Aggressive Voltage Scaling." Transactions on VLSI 2009
- [14] D. Patterson et al., "A case for Redundant Arrays of Inexpensive Disks (E-RAID)." University of California Berkeley. 1988
- [15] R. Morris, and B. Truskowski, "The evolution of storage systems". IBM Syst. J. 42, 2 (Apr. 2003), 205-217.
- [16] F. Ruckerbauer et al., "Soft Error Rates in 65nm SRAMs--Analysis of new Phenomena." In Proceedings IOLTS '07, July 2007.
- [17] R. Mastipuram and E. C. Wee. Soft Errors' Impact on System Reliability. <http://www.edn.com/article/CA454636>, Sep 2004
- [18] S. Nassif., "Modeling and Analysis of Manufacturing Variations", Proceedings of 2001 IEEE CICC Conference.
- [19] A. Djahromi et al, "Cross Layer Error Exploitation for Aggressive Voltage Scaling." In Proceedings of ISQED '07.
- [20] P. Chen et al., "RAID: high-performance, reliable secondary storage." ACM Comput. Surv. 26, 2 (Jun. 1994)
- [21] L. Bathen et al., "Inter-kernel Data Reuse and Pipelining on Chip-Multiprocessors for Multimedia Applications," *ESTMedia '09, Grenoble, France, Oct 2009*
- [22] SystemC LRM, May 2005, (ver2.1). <http://www.systemc.org>
- [23] L. Bathen et al., "A Methodology for Power-aware Pipelining via High-Level Performance Model Evaluations," MTV '09, Austin, TX, Dec '09
- [24] S. Ramaswamy et al., "Improving cache efficiency via resizing + remapping", ICCD 2007: 47-54
- [25] S. Kim. Area-efficient error protection for caches. In Proceedings of DATE'06, Mar 2006
- [26] M. Verma et al., "Data Partitioning for Maximal Scratchpad Usage," ASP-DAC, pp. 77-83, 2003
- [27] M. Kandemir et al., "Dynamic Management of Scratch-pad Memory Space," DAC, pp. 690-695, 2001.
- [28] I. Issenin et al., "Data Reuse Analysis Technique for Software-Controlled Memory Hierarchies," DATE, pp. 202-207, 2004.
- [29] M. S. Hrishikesh et al. "Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures," *ISCA*, p. 248, 2000
- [30] K. Olukotun et al. "The case for a single-chip multiprocessor." *SIGPLAN*, pp. 2-11. Sep. 1996.
- [31] The Cell project at IBM Research, <http://www.research.ibm.com/cell/>
- [32] Intel Multi-Core Technology, http://www.intel.com/multi-core/index.htm?iid=tech_as_1ln+multi
- [33] Intel's Teraflops Research Chip, <http://techresearch.intel.com/articles/Tera-Scale/1449.htm>
- [34] Tiler's Tile Gx Family, http://www.tiler.com/products/processors/TILE-Gx_Family
- [35] UCI Center for Embedded Computer Systems TR #10-12, Dec. 2010