

Fault Tolerant Memory Design for HW/SW Co-Reliability in Massively Parallel Computing Systems

M. Choi

Department of Electrical and Computer Engineering
University of Missouri-Rolla, Rolla, MO 65409-0040
choim@umr.edu

N.-J. Park, K.M. George, B. Jin and N. Park †

Department of Computer Science
Oklahoma State University, Stillwater, OK 74078-1053
{kmg,npark}@a.cs.okstate.edu

Y.B. Kim, F. Lombardi

Department of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115
{ybk,lombardi}@ece.neu.edu

Abstract—A highly dependable embedded fault-tolerant memory architecture for high performance massively parallel computing applications and its dependability assurance techniques are proposed and discussed in this paper. The proposed fault tolerant memory provides two distinctive repair mechanisms: the permanent laser redundancy reconfiguration during the wafer probe stage in the factory to enhance its manufacturing yield and the dynamic BIST/BISD/BISR (built-in-self-test-diagnosis-repair)-based reconfiguration of the redundant resources in field to maintain high field reliability. The system reliability which is mainly determined by hardware configuration demanded by software and field reconfiguration/repair utilizing unused processor and memory modules is referred to as HW/SW Co-reliability. Various system configuration options in terms of parallel processing unit size and processor/memory intensity are also introduced and their HW/SW Co-reliability characteristics are discussed. A modeling and assurance technique for HW/SW Co-reliability with emphasis on the dependability assurance techniques based on combinatorial modeling suitable for the proposed memory design is developed and validated by extensive parametric simulations. Thereby, design and implementation of memory-reliability-optimized and highly reliable fault-tolerant field reconfigurable massively parallel computing systems can be achieved.

Index Terms—High performance computing, Massively parallel computing, Fault-tolerant memory, Built-in-self-repair (BISR), Yield, Reliability, Field Reconfiguration, Modular Redundancy, HW/SW Co-reliability, Reliability Assurance.

I. INTRODUCTION

Advances in massively parallel computing system technology mainly depends on the high performance ultra reliable system components. Typical massively parallel multiprocessor systems are built with processors, an interconnection network and memories. Reliability and data integrity are commonly emphasized requirements of solid-state mass storage systems since faults can be experienced throughout their life time [1], [2], [3], [7]. Acceptable dependability for these solid-state mass storage systems have been commonly achieved by using redundancy and repair. Reconfiguration (repair) of memory arrays using spare memory lines is the most common technique for reliability enhancement of memories with faults [1], [2], [3], [4], [5], [6].

† N. Park the contact author

As advances in Ultra-Large-Scale-Integration (ULSI) technologies make possible the seamless embedding of numerous cores on a single chip (i.e., Commonly referred to as *System-On-Chip* technology), solid dependability becomes urgently emphasized requirement of such ultra density & high performance system since insignificant degradation or defect of core components could result in unacceptably low resultant SoC manufacturing yield and field reliability. Among the cores for SoC integration, one of the most sensitive cores is the embedded memory core since memory cells are commonly considered as prone to defects and faults [8], [6], [9], [11], [1], [13], [14], [16]. As SoC fabrication process goes toward the era of the deep-sub-micron technology such as $0.13\mu m$, need for a high yield and ultra reliable embedded memory core becomes obvious. According to Semiconductor Industry Association and ITRS2000, embedded memory will continue to dominate SoC content in the next several years, approaching 94% of the die area by 2014 as shown in Figure (1) [10]. The issues surrounding high-density multi-megabit memory yield must be solved in order to facilitate this trend and to produce cost effective SoC product.

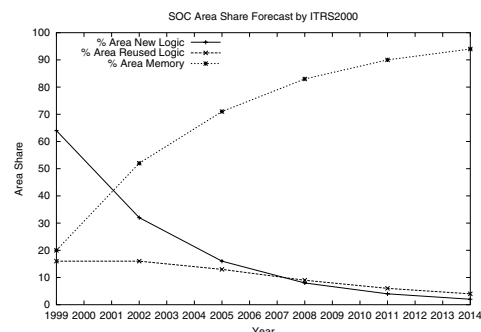


Fig. 1. SoC area share forecast of embedded memory core from 1999 to 2014 by ITRS2000

Several research results regarding multiprocessor systems have been reported. In [20], the fault detection capability (FDC) and the fault location capability (FLC) for multiprocessor systems with spare processor modules were discussed and its results showed that these schemes utilize spare capacity more efficiently, thereby improving upon the fault detection and location capabilities of the system. The proposed task allocation method for multiprocessor system in [22] was a simple yet effective method to allocate the tasks in multiprocessor systems for minimizing the interprocessor communication cost subject to resource limitations defined by the system and designer. [17] introduced a multiprocessor system with reconfigurable pipelined bus system for parallel matrix multiplication and reconfiguration algorithms. [19] proposed a dynamic reconfiguration network (DRN) and a monitoring-at-transmission (MAT) bus to support dynamic reconfiguration of an N -modular redundancy multiprocessor system in order to provide gracefully degradable operations. In [18], an availability model for multiprocessor systems based on Markov chain was proposed and validated. Availability and reliability of mesh-connected systems were discussed in [21]. In addition, another efficient technique for computing the reliability of k -to- l -out-of- n systems was also presented in [23] which can be used to estimate reliability of static multiprocessor systems. Unfortunately, HW/SW Co-reliability of field reconfigurable multi-processor-memory systems has not yet been reported.

Traditionally, reconfiguration (repair) of memory arrays using spare memory lines is the most common technique for reliability enhancement of memories with faults [8], [6], [9], [11], [1], [13], [14], [16]. In [9], very simple built-in-self-analysis-repair scheme called CRESTA for embedded DRAM is proposed while a row-column self-repair scheme for embedded SRAM for Alpha 21264 is shown in [11]. A shared built-in self-repair analysis scheme (Shared-BISA) for multiple embedded memory cores in the SoC is proposed in [8] to realize minimum area penalty independent of the number of embedded memory cores. Error-correcting-code (ECC) has been also used to develop more reliable memory devices in terms of data integrity and dependability [14], [15]. Especially, synergistic fault-tolerance capabilities of the combination of ECC and row/column redundancy is discovered in [14]. However, comprehensive research work on fault-tolerant embedded memory core organization and its dependability assurance has not been done yet.

The main motivation of this paper is to propose a high yield, ultra reliable memory system for massively parallel computing applications which is capable of both factory and field repair to enhance its factory yield and field reliability at the same time, while establishing a series of accurate dependability assurance techniques for the proposed memory architecture. Ultimately, the proposed work provides a HW/SW Co-reliability assurance technique based on a general architectural model of field reconfigurable multi-processor-memory systems with emphasis on memory system. Thereby, architecture-flexible, reliability-assured and cost-minimized design and implementation and field reconfiguration of the massively parallel multi-processor-memory systems can be realized.

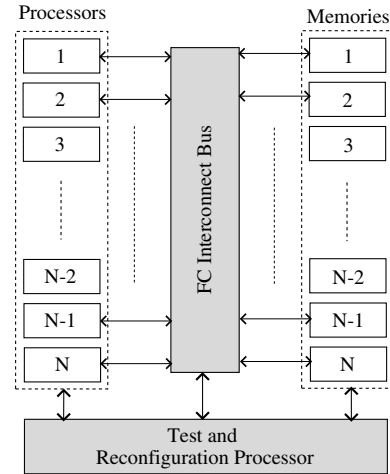


Fig. 2. The proposed Field Reconfigurable Multiprocessor System Model.

II. PRELIMINARIES AND REVIEW

A generalized architectural model for field reconfigurable massively parallel multi-processor-memory systems given in Figure (2) is considered in this paper. Characteristics of the model can be summarized as follows.

- N FC (Field Reconfigurable) processors and N FC memories communicate via a FC interconnection bus.
- One parallel processing unit (PPU) consists of p processors and $m \times n \times n$ memories and dedicated reconfigured interconnects. u noncoherent units are supposed to execute parallelly at the same time.
- 3-tuple of the hardware configuration (m, p, u) is solely demanded by software in field.
- Faulty cells occurring in a memory can be locally tolerated by field-reconfiguration of the given spare column redundancy.
- Unused processors and memories can be field-reconfigured to replace faulty processors and memories in PPUs.
- Reliability of FC interconnect bus is not considered.
- PPUs are noncoherent each other, which means that tasks assigned to PPUs are mutually exclusive.

For instance, $(1, 2, 2)$ 3-tuple configuration on a $N = 8$ system is shown in Figure (3) in which the first PPU consists of processor module 1 and memory module 1 and 2 and corresponding virtual interconnection network provided by the FC interconnect bus and the second PPU uses processor module 2 and memory module 3 and 4. Since processor module 3-8 and memory module 5-8 are unused, they can be used to replace faulty processor modules upto six and faulty memory modules upto four. Figure (4) further illustrates repair process by utilization of modular redundancy. If processor module 2 and memory module 4 are diagnosed as faulty by the Test and Reconfiguration Processor (TRP), it dynamically reallocates processor module 3 and memory module 5 to tolerate such faulty modules. As a result, the system ends up with the configuration of PPU 1 and reconfigured PPU 2 and 5 redundant processor modules and 3 redundant memory modules. Processor 2 and memory module 4 marked as faulty by TRP and will no longer be used.

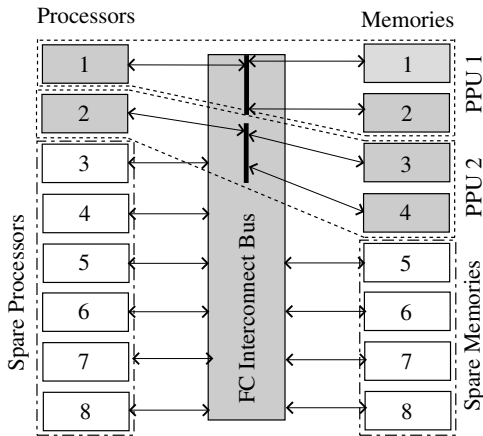


Fig. 3. (1,2,2) System Configuration (2 1-processor 2-memory PPU).

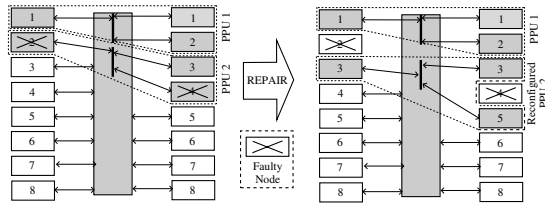


Fig. 4. Repair of faulty processor and memory throughout modular redundancy reconfiguration.

The atomic unit of the memory in the proposed massively parallel multi-processor-memory system is referred to as *block*. To enhance not only data integrity but also manufacturing yield and field reliability, both the double-error-detection, single-error-correction (DED-SEC) error-correcting-code (ECC) and the spare row/column redundancy are built in each block of memory. 128 bit data word with 9 bit ECC has been chosen while the other configuration of ECC is technically feasible. Each memory block has r ECC words and S_c spare columns and S_r spare rows for the sake of fault-and-defect-tolerance of the memory as shown in Figure (5). Almost every numerical pa-

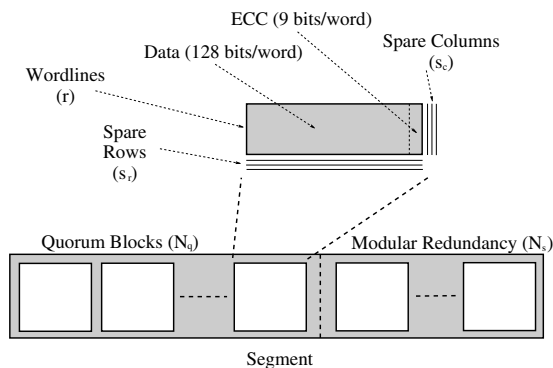


Fig. 5. Architecture of the memory block and segment

parameter of the proposed memory is not fixed to exploit the maximum flexibility to the resultant memory aspect. Thereby, cost-effective, target-specific memory design optimization is always possible. The next unit of the proposed memory organization is the *segment*. Segment consists of N_q quorum ECC memory blocks with row/column redundancy. Memory designer can assign N_s spare memory blocks to overcome costly block-wise failures. Figure (5) shows an illustration of a segment with N_q quorum memory blocks with N_s spare memory blocks. This kind of modular redundancy technique will enhance memory dependability significantly as well. The next memory unit is the *segmented memory array* which consists of N_{seg} segments. Therefore, the total size of the memory is $128 \times r \times N_q \times N_{seg}$ bits. The proposed memory consists of the following components as shown in Figure (6).

- BIST/BISD/BISR Processor : This system component governs self-test, self-diagnosis, and self-repair procedures.
- Laser Fuse : A set of laser reconfigurable fuses to permanently program the given redundancy resources.
- EEPROM Fuse : A programmable, non-volatile memory to store additional reconfiguration signature generated by the BIST/BISD/BISR processor in field.
- IEEE JTAG 1149.1 : External test equipment interface.
- Memory Array Interface : This component connects the segmented memory array and the BIST/BISD/BISR Processor together. Data, address, control and repair data flow via this component.

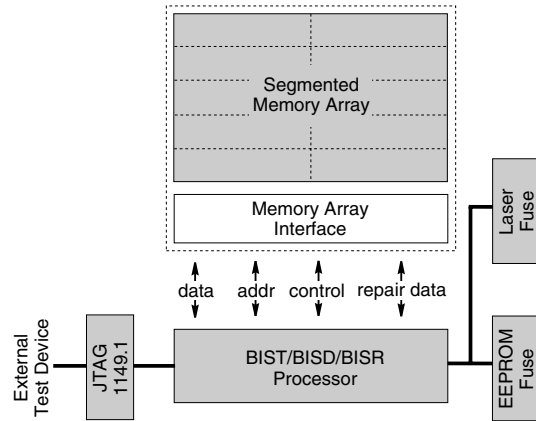


Fig. 6. Logical relationship between the proposed fault-tolerant memory core components

Memory reconfiguration using row/column redundancy is one of the most famous NP-complete problems [1], [13], [15], [16]. Thus, numerous approximation algorithms guaranteeing near-optimal solutions have been introduced on the top of the quite unrealistic assumption: high faulty cell probability, while world-class semiconductor manufacturers produce quality memories which contains usually one or less faulty cell per million cells [14]. In addition, most of the memory reconfiguration algorithms are not suitable to be embedded in the SoC in real world situation due to their complexity. Thus, very simple yet effective hierarchical repair algorithm is to be pro

follows.

1. *ECC* : single-error-correction and double-error-detection (SEC-DED) error correcting code covers one-bit permanent or transient faults and detects 2-bit permanent or transient faults.
2. *Wordline Redundancy* : If more than 2 failing bits are diagnosed in a wordline or the support circuit of the wordline itself is analyzed as faulty, a wordline spare replaces the faulty memory line.
3. *Bitline Redundancy* : Likewise, if more than 2 failing bits are diagnosed in a bitline or the support circuit of the bitline itself is analyzed as faulty, a bitline spare replaces the faulty memory line.
4. *Block-wise Modular Redundancy* : Within a segment, failing blocks also can be repaired by spare blocks. This kind of repair method is commonly referred to as *modular redundancy technique*.

The proposed fault-tolerant memory can also provide not only factory repairability but also field self-test and repair capabilities. The first repair process may take place during the wafer probe in the factory. The repair signature created by the proposed memory is sent to an external test equipment via the IEEE JTAG 1149.1, which is interface commonly used connectivity between the external tester and the embedded BIST/BISD/BISR processor [8], [9], [11]. Then, conventional laser repair equipment blows the fuse box with the required reconfiguration information. Afterwards, the memory contains the unique permanent repair signature. The factory repair also has the advantage of performing test and repair at the wafer level. The wafer can be subjected to a variety of stringent conditions during testing which helps insure high memory dependability (extended voltage, temperature and frequency conditions).

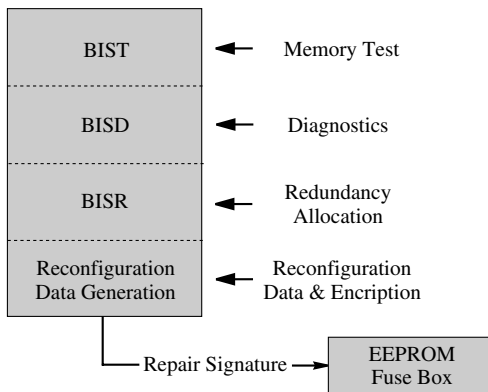


Fig. 7. Logical flow of the BIST/BISD/BISR procedure

The field repair operation tests and repairs memory instances each time the end product starts up or is reset. Logical view of the field repair procedure is given in Figure(7). The BIST/BISD/BISR Processor initiates and operates the test program, determines defective memory locations (if any), allocates redundancy resources, and produces a new repair signature that resides in the EEPROM fuse on the memory (similar to the EEPROM-controlled redundancy allocation technique proposed in [6]). This non-volatile repair signature is applied to the

dundancy control logic and remains there as long as new repair signature is overwritten. As a result, it is possible to design an MPCs that incorporates a proposed memory with both factory and field repair in which factory repair process repairs manufacturing defects, and field repair operation covers any subsequent problems that may materialize over the life of the MPCs.

Following assumptions are also made to analyze the manufacturing yield of the proposed fault-tolerant memory.

- A failing bit in a word is supposed to be corrected by Single-Error-Correction & Double-Error-Detection (SEC-DED) Error-Correcting-Code (ECC).
- Multiple faults in a word can be repaired by row redundancy.
- Multiple faults in a column can be repaired by column redundancy.
- Block-wise failure can be repaired by the block modular redundancy.

III. HW/SW CO-RELIABILITY ANALYSIS

The following notations are used in this work:

- $R_{MPCS}(t)$: Reliability of the proposed MPCs.
- t : time.
- Δt : Unit time interval.
- d : Number of data bits per ECC word.
- c : Number of code bits per ECC word.
- n_c : Number of columns per block without spare columns. ($n_c = d + c$)
- n_r : Number of rows per block without spare rows.
- s_r (s_c) : Number of spare rows (columns) per block.
- N : Number of total processor/memories
- N_q : Number of quorum blocks per segment.
- N_s : Number of modular spare blocks per segment.
- N_{seg} : Number of segments per memory system.
- λ_m : Expected number of manufacturing failures per memory cell.

The system reliability, which is mainly determined by 3-tuple hardware configuration demanded by software and field reconfiguration and repair utilizing unused processor and memory modules, is referred to as HW/SW Co-reliability estimation and assurance technique will be proposed in this section. Related with field reliability assurance, the fault-tolerant memory is able to perform Built-In-Self-Test(BIST), Built-In-Self-Diagnosis (BISD), and Built-in-Self-Repair(BISR) in field. These special capabilities of the memory architecture further enhance its reliability during field operation. Thus, rapid degradation of the system reliability can be prevented. Reliability assurance of the memory will be based on the following assumptions.

- As the yield of the laser-repaired memory at the factory wafer probe stage has been analyzed based on expected number of failures due to the manufacturing defects, failure arrival rate per unit time will be used to estimate reliability of the proposed fault-tolerant memory in field. To differentiate the expected number of failures due to the manufacturing defects and the failure arrival rate per unit time, λ_m and λ_f are used, respectively.
- Embedded BIST/BISD/BISR processor tests, analyzes and repairs faulty memory in field when it is reset or

For simplicity, the reset/restart event is assumed to occur once in every unit interval time (Δt).

Following parameters are used for reliability assurance analysis of the fault-tolerant memory.

- t : time.
- Δt : Unit time interval (e.g, day, week or month, etc).
- λ_f : Field failure arrival rate of memory cell per unit time interval.
- $R(t)$: Reliability of the memory at time t .
- $\Psi(t)$: Overall dependability of the memory at time t (i.e., $Pr\{\text{manufactured as good AND not failing during the time interval } [0, t]\}$).

The *reliability* of a system is a function of time which is defined by the conditional probability that the system performs correctly throughout the interval of time $[t_0, t]$ given the system was performing flawlessly at the initial time t_0 [12]. If we assume that the failure arrival rate of the system has a constant value of λ , the reliability of the system can be expressed as follows

$$R(t) = e^{-\lambda t} \quad (1)$$

The exponential relationship between the reliability and time is known as the *exponential failure law* which states that for a constant failure arrival rate function, the reliability varies exponentially as a function of time. The exponential failure law is quite valuable to analyze electronic components and is very well-known and widely-used relationship between the reliability and time.

Since the yield can be viewed as the reliability of the system at the initial time t_0 , the formulas for the yield assurance can be extended to derive the reliability formulas with some minor modifications. At the initial time t_0 [12], the probability of a single cell is fault-free is $Y_{cell} = e^{-\lambda_m}$. Therefore, the reliability of a single cell (i.e., the conditional probability of a single cell is fault-free from t_0 to t) can be written as

$$R_{cell}(t) = Y_{cell} \cdot e^{-\lambda_f t} \quad (2)$$

where Y_{cell} is the yield of a single cell and λ_f is the failure arrival rate of a single memory cell in field.

Since the BIST/BISD/BIST shares same redundancy structure and repair mechanism with the fabrication-time repair, the yield assurance formulae proposed can be used if $R_{cell}(t)$ replaces Y_{cell} . Thus, the field reliability assurance formulae will be listed without detailed explanations.

$$R_{word}(t) = R_{cell}(t)^{n_c} \quad (3)$$

if ECC is not used

$$R_{word}(t) = R_{cell}(t)^{n_c} + 137 R_{cell}(t)^{(n_c-1)}(1 - R_{cell}(t)) \quad (4)$$

if ECC is used.

$$P_{mfw}(t) = \sum_{i=2}^{n_c} \binom{n_c}{i} (R_{cell}(t))^{(n_c-i)} \cdot (1 - R_{cell}(t))^i \quad (5)$$

$$P_{mfw}(t) = 1 - R_{word}(t) \quad (6)$$

$$P_{s_r}(t) = \sum_{j=1}^{s_r} \binom{n_r}{j} (P_{mfw}(t))^j \cdot (1 - P_{mfw}(t))^{(n_r-j)} \quad (7)$$

$$P_{s_r \geq j}(t) = \sum_{k=j}^{s_r} \binom{s_r}{k} (R_{word}(t))^k \cdot (1 - R_{word}(t))^{(s_r-k)} \quad (8)$$

$$R_{s_r}(t) = \sum_{j=1}^{s_r} \sum_{k=j}^{s_r} \binom{n_r}{j} \binom{s_r}{k} (P_{mfw}(t))^j \cdot (1 - P_{mfw}(t))^{(n_r-j)} \cdot (R_{word}(t))^k \cdot (1 - R_{word}(t))^{(s_r-k)} \quad (9)$$

$$R_{block:s_r}(t) = R_{word}(t)^{n_r} + R_{s_r}(t) \quad (10)$$

$$P_{fc}(t) = 1 - \log_{n_c} R_{block}(t) \quad (11)$$

$$P_{s_c}(t) = \sum_{j=1}^{s_c} \binom{n_c}{j} (P_{fc}(t))^j \cdot (1 - P_{fc}(t))^{(n_c-j)} \quad (12)$$

$$P_{s_c \geq j}(t) = \sum_{k=j}^{s_c} \binom{s_c}{k} (R_{cell}(t))^{(n_r \times k)} \cdot (1 - (R_{cell}(t))^{n_r})^{(s_r-k)} \quad (13)$$

$$R_{s_c}(t) = \sum_{j=1}^{s_c} \sum_{k=j}^{s_c} \binom{n_c}{j} \binom{s_c}{k} (P_{fc}(t))^j \cdot (1 - P_{fc}(t))^{(n_c-j)} \cdot (R_{cell}(t))^{(n_r \times k)} \cdot (1 - (R_{cell}(t))^{n_r})^{(s_r-k)} \quad (14)$$

$$R_{block}(t) = R_{block:s_r}(t) + R_{s_c}(t) \quad (15)$$

$$R_{seg}(t) = \sum_{i=0}^{N_s} \binom{N_q + N_s}{i} (R_{block}(t))^{N_q + N_s - i} \cdot (1 - R_{block}(t))^i \quad (16)$$

Finally, overall reliability of the system enhanced by modular redundancy can be written as

$$R_{MPCS}(t) = \left(\sum_{i=0}^{M_p} \binom{N}{i} R_p(t)^{N-i} \cdot (1.0 - R_p(t))^i \right) \cdot \left(\sum_{j=0}^{M_m} \binom{N_{seg}}{j} R_{seg}(t)^{N_{seg}-j} \cdot (1.0 - R_{seg}(t))^j \right) \quad (17)$$

where M_p is the number of unused processor modules which can be utilized as spare processor modules and M_m is the number of unused memory modules which can be used as spare memory modules.

IV. PARAMETRIC SIMULATIONS

In field operations, the software demands 3-tuple configuration of the given hardware: i.e., (p, m, u) where p is the number of processors; m is the number of memories in PPU; and u is the number of PPUs. Utilization of the proposed field reconfigurable multi-processor-memory system can be categorized in terms of relative PPU size as follows.

1. Coarse-Grained Utilization (CG) : If PPU size is relatively large, such as $(9, 9, u)$ system, the system can be categorized as CG. For the given number of processor/memory modules N , relatively smaller number of relatively larger PPUs can be allocated. Thus, level of parallelism is low while processing/storage capabilities of each PPU are superior.
2. Medium-Grained Utilization (MG) : Compromised category, such as $(5, 5, u)$ system, between CG and FG. It has balanced level of parallelism and processing/storage capabilities.
3. Fine-Grained Utilization (FG) : If PPU size is relatively small, such as $(2, 2, u)$ system, the system can be categorized as FG. For the given number of processor/memory modules N , relatively larger number of relatively smaller PPUs can be allocated. Thus, level of parallelism is high while processing/storage capabilities of each PPU are inferior.

Utilization of the proposed system can be also categorized in terms of processor/memory utilization intensity as follows.

1. Processor Intensive Utilization (PI) : If $p > m$, then the system can be categorized as PI. PPUs have higher processing capability and lower storage capability.
2. Evenly Intensive Utilization (EI) : If $p = m$, then the system can be categorized as EI. PPUs have balanced processing and storage capabilities.
3. Memory Intensive Utilization (MI) : If $p < m$, then the system can be categorized as MI. PPUs have lower processing capability and higher storage capability.

TABLE I
3-TUPLE SIMULATION CONFIGURATIONS

Category	3-tuple
<i>CGPI</i>	(12, 9, 5)
<i>CGEI</i>	(9, 9, 6)
<i>CGMI</i>	(9, 12, 5)
<i>MGPI</i>	(7, 5, 8)
<i>MGEI</i>	(5, 5, 12)
<i>MGMI</i>	(5, 7, 8)
<i>FGPI</i>	(3, 2, 20)
<i>FGEI</i>	(2, 2, 30)
<i>FGMI</i>	(2, 3, 20)

Thus, combination of those six categories further yields nine possible utilization patterns: CGPI, CGEI, CGMI, MGPI, MGEI, MGMI, FGPI, FGEI, and FGMI. Sample 3-tuple configurations which represent those nine categories are shown in Table I. Characteristics of the categories will be investigated by extensive parametric simulations.

TABLE II
SIMULATION PARAMETERS

Parameter	Meaning	Value
Y_{cell}	Yield of a single cell	0.99
S_r, S_c	# of spare row/column	Variable(2, 4, 8)
N_q	# of quorum block	8
N_s	# of spare block	Variable(0, 1, 2)
N_{seg}	# of segment	8
Λ_{cell}	Field failure arrival rate for memory cell	$5 \cdot 10^{-6}$
Δt	Unit time interval	One week

The proposed HW/SW Co-reliability estimation and assurance technique is further verified by parametric simulation. Commonly used simulation parameters are shown in Table II (i.e., 64 processor/memory modules, 64×64 memory module with 8 spare columns, the failure arrival rate for each processor is 5×10^{-4} and the failure arrival rate for each memory cell is 1.4×10^{-6}). The 3-tuple simulation parameters in Table II are used for the proposed nine PPU categories are Parametric simulation results are shown in Figure (8) - (16) where HW/SW Co-reliability of both non-fault-tolerant and fault-tolerant systems are compared. For example, the CGEI system configuration enhanced by redundancy utilization can gracefully maintain above 95% reliability for $700\Delta t$ while the same system without repair hardly does for even $100\Delta t$. The following observations can be obtained from the results.

1. Field reconfiguration of the unused processor and memory modules to repair faulty PPU modules significantly enhances overall system reliability.
2. Since memory modules are locally fault-tolerant by spare columns, they show more graceful degradation. Thus, processor intensive configurations (PIs), which yields more memory redundant modules, are more reliable than the other system configurations (i.e., PI is more reliable than EI, and EI is more reliable than MI).
3. Difference between CG and EG and FG in terms of overall system reliability is not so significant, which means M_p and M_m mainly determines the system reliability. Only differences between them are level of parallelism.
4. The proposed HW/SW Co-reliability estimation technique is very effective and accurate so that it can be used to assure reasonable system reliability while avoiding unnecessary cost due to larger number of N . This means that this technique can be used to optimize M_p and M_m and s for the demanded 3-tuple system configuration by SW with emphasis on the effect of the proposed fault tolerant memory system.

The dependability of the proposed fault-tolerant memory in terms of the field reliability is assured by the proposed estimation techniques via a series of parametric simulations. Used parameters are summarized as shown in Table II. Although the proposed memory architecture is flexible enough to implement various embedded memory configurations, a sample 1Mbit memory has been used for the parametric simulations.

V. DISCUSSION AND CONCLUSIONS

As advances in field reconfigurable system component technology makes possible the massively parallel multi-processor-memory systems, high memory system reliability becomes commonly emphasized requirement of such systems since insignificant degradation of some components could result in unacceptably low overall system reliability. Thus, HW/SW Co-reliability measurement and estimation for the field reconfigurable multi-processor-memory systems using combinatorial modeling method has been proposed with emphasis on the fault-tolerant memory system and validated throughout the parametric simulations in this paper. System configuration patterns demanded by SW can be categorized by the proposed nine representative patterns associated with the proposed memory system. Analysis of the nine configuration patterns by using the proposed fault-tolerant memory system and HW/SW Co-reliability assurance technique reveals that 1) field reconfiguration of the unused processor and memory modules to repair faulty PPU modules significantly enhances overall system reliability and 2) processor/memory intensity is more significant than PPU size and 3) M_p and M_m still mainly determines the system reliability. The proposed HW/SW Co-reliability estimation technique is very effective and accurate so that it can be used to assure reasonable memory system reliability (especially in case of memory-intensive SW applications) while avoiding unnecessary cost due to larger N . This means that this technique can be used to optimize M_p and M_m and s for the demanded 3-tuple system configuration by SW thereby realizing optimal design of fault-tolerant memory system associated with its SW demand.

REFERENCES

- [1] Low C.P. and Leong H.W., "A New Class of Efficient Algorithms for Reconfiguration of Memory Arrays", *IEEE Transactions on Computers*, Vol 45, No 5, pp. 614-618, 1996.
- [2] N. Park and F. Lombardi "Repair of memory arrays by cutting", *Memory Technology, Design and Testing, 1998. Proceedings. International Workshop on*, pp. 124-130, August 1998.
- [3] D. M. Blough, "Performance Evaluation of a Reconfiguration-Algorithm for Memory Arrays containing Clustered Faults", *IEEE Transactions on Reliability*, Vol. 45, No. 2, pp. 274-284 June 1996.
- [4] D. M. Blough and A. Pelc, "A Clustered Failure Model for the Memory Array Reconfiguration Problem", *IEEE Transactions on Computers*, Vol. 42, No. 5, pp. 518-528 May 1993.
- [5] Y. Jeon, Y. Jun and S. Kim, "Column Redundancy Scheme for Multiple I/O DRAM using Mapping Table", *Electronics Letter*, Vol. 36, No. 11, May 2000.
- [6] R.J. McPartland, D.J. Loeper et al. "SRAM Embedded Memory with Low Cost, FLASH EEPROM-Switch-Controlled Redundancy", *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000*, Vol. 36, No. 11, pp. 287-289, May 2000.
- [7] K.A. Clark, B.W. Johnson, "A Fault-Tolerant Solid-State Memory for Spaceborne Applications", *Proceedings of the Government Microelectronics Applications Conference*, pp. 441-444, November 1992.
- [8] J. Ohtani, T. Ooishi, et al., "A Shared Built-In Self-Repair Analysis for Multiple Embedded Memories", *Custom Integrated Circuits, 2001, IEEE Conference on*, pp. 187-190, May 2001.
- [9] T. Kawagoe, J. Ohtani, et al., "A Built-In Self-Repair Analyzer (CRESTA) for Embedded DRAMs", *Test Conference, 2000. Proceedings. International*, Vol. 41, No. 9, pp. 567-574, Oct. 2000.
- [10] International Technology Roadmap for Semiconductors. "International Technology Roadmap for Semiconductors 2000", <http://public.itrs.net/Files/2000UpdateFinal/2kUdFinal.htm>, 2000.
- [11] D.K. Bhavsar, "An Algorithm for Row-Column Self-Repair of RAMs and its Implementation in the Alpha 21264", *Test Conference, 1999. Proceedings. International*, pp. 311-318, Sep. 1999.
- [12] S.-K. Lu, S.-Y. Kuo, C.-W. Wu, "Fault-Tolerant Interleaved Memory Systems with Two-Level Redundancy", *Computers, IEEE Transactions on*, Vol. 46, Issue. 9, pp. 1028-1034, Sep. 1997.
- [13] D. M. Blough, "Performance Evaluation of a Reconfiguration-Algorithm for Memory Arrays containing Clustered Faults", *IEEE Transactions on Reliability*, Vol. 45, No. 2, pp. 274-284 June 1996.
- [14] C.H. Stapper, H.-S. Lee, "Synergistic Fault-Tolerance for Memory Chips", *IEEE Trans. on Computers*, Vol. 41, No. 9, pp. 1078-1087, September 1992.
- [15] E. Fujiwara, D.K. Pradhan, "Error-Control Coding in Computers", *IEEE Computer*, Vol. 23, Issue. 7, pp. 63-72, Jul. 1990.
- [16] S.Y. Kuo and W.K. Fucks, "Efficient Spare Allocation in Reconfigurable Arrays", *IEEE Design and Test*, Vol. 41, Issue. 9, pp. 24-31, Feb. 1987.
- [17] Li Keqin and V.Y. Pan, "Parallel matrix multiplication on a linear array with a reconfigurable pipelined bus system", *Computers, IEEE Transactions on*, Vol. 50 Issue. 5, pp. 519-525, May 2001.
- [18] G. Rubino and B. Sericola, "Interval availability analysis using denumerable Markov processes: application to multiprocessor subject to breakdowns and repair", *Computers, IEEE Transactions on*, Vol. 44 Issue. 2, pp. 286-291, Feb. 1995.
- [19] Liu Jyh-Charn and K.G. Shin, "Efficient implementation techniques for gracefully degradable multiprocessor systems", *Computers, IEEE Transactions on*, Vol. 44 Issue. 4, pp. 503-517, Apr. 1995.
- [20] S. Tridandapani, A.K. Somani and U.R. Sandadi, "Low overhead multiprocessor allocation strategies exploiting system spare capacity for fault detection and location", *Computers, IEEE Transactions on*, Vol. 44 Issue. 7, pp. 865-877, Jul. 1995.
- [21] P. Mohapatra and C.R. Das, "On dependability evaluation of mesh-connected processors", *Computers, IEEE Transactions on*, Vol. 44 Issue. 9, pp. 1073-1084, Sep. 1995.
- [22] C.-I.H. Chen and V. Cherkassky, "Task allocation and reallocation for fault tolerance in multicomputer systems", *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 30 Issue. 4, pp. 1094-1104, Oct. 1994.
- [23] S.J. Upadhyaya and H. Pham, "Analysis of noncoherent systems and an architecture for the computation of the system reliability", *Computers, IEEE Transactions on*, Vol. 42 Issue. 4, pp. 484-493, Apr. 1993.

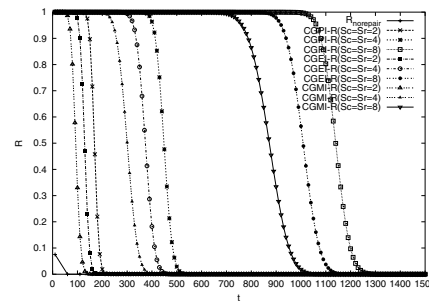


Fig. 8. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 0$)

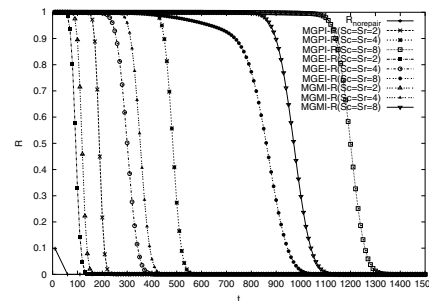


Fig. 9. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 0$)

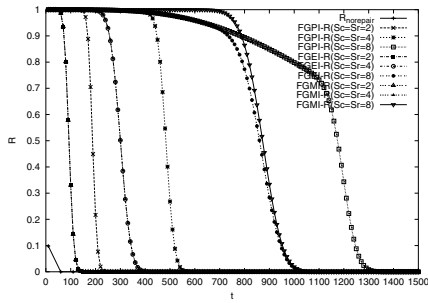


Fig. 10. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 0$)

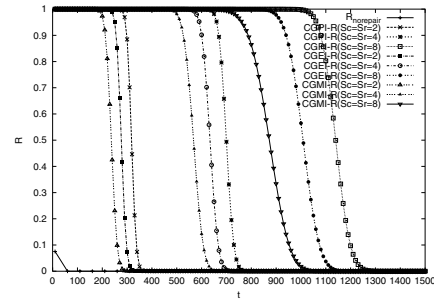


Fig. 14. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 2$)

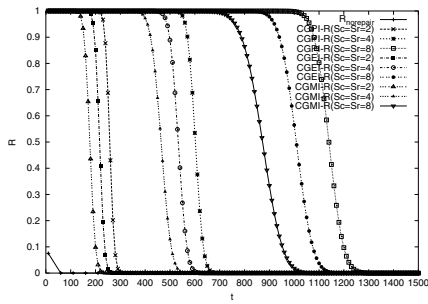


Fig. 11. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 1$)

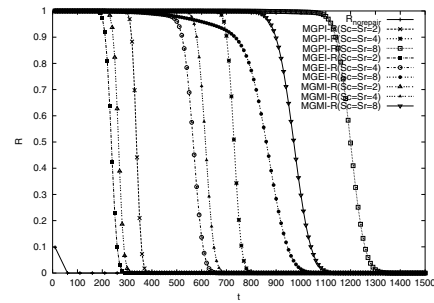


Fig. 15. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 2$)

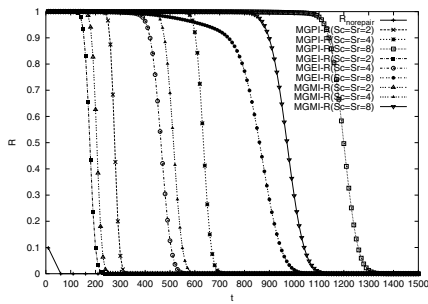


Fig. 12. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 1$)

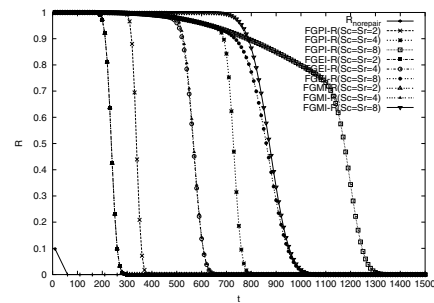


Fig. 16. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 2$)

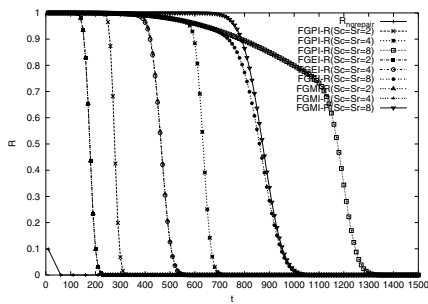


Fig. 13. Reliability analysis results of the sample 16K x 8 x 8 (1M) Segment ($N_s = 1$)