01 Dec 2002

Dynamic Yield Analysis and Enhancement of FPGA Reconfigurable Memory Systems

Minsu Choi

Missouri University of Science and Technology, choim@mst.edu

Nohpill Park

Recommended Citation

M. Choi and N. Park, "Dynamic Yield Analysis and Enhancement of FPGA Reconfigurable Memory Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 6, pp. 1300-1311, Institute of Electrical and Electronics Engineers (IEEE), Dec 2002.

The definitive version is available at https://doi.org/10.1109/TIM.2002.808046

Dynamic Yield Analysis and Enhancement of FPGA Reconfigurable Memory Systems

Minsu Choi, *Member, IEEE,* and Nohpill Park, *Member, IEEE*

***Abstract*—This paper addresses the issues of field programmable gate arrays (FPGA) reconfigurable memory systems with faulty physical memory cells and proposes yield measurement techniques. Static yield (i.e., the yield which does not take into account the inherited redundancy utilization for repair) and dynamic yield (i.e., the yield which takes into account the inherited redundancy utilization for repair) of FPGA reconfigurable memory systems and their characteristics are extensively analyzed. Yield enhancement of conventional memory systems relies on additional redundancy, but FPGA reconfigurable memory systems have inherited redundancy and customizability. Thus, they can accommodate numerous target memory configurations, and redundant memory cells, if any, can be used as spares to enhance the dynamic yield of a target memory configuration. Three fundamental strategies are introduced and analyzed; i.e., redundant bit utilization, redundant word utilization, and a combination of both. Mathematical analysis of those techniques also has been conducted to study their effects on the yield. Selecting the most yield enhancing logical memory configuration which can accommodate a target memory requirement among the candidate configurations is referred to as optimal fitting. Optimal fitting algorithms for single configuration fitting, sequential reconfiguration system fitting, and concurrent reconfiguration system fitting are investigated based on the proposed yield analysis techniques.**

***Index Terms*—Concurrent reconfiguration, dynamic yield, field programmable gate arrays (FPGA), FPGA-based instrumentation, memory yield enhancement, memory yield measurement, optimal fitting, reconfigurable memory, sequential reconfiguration, static yield.**

I. Introduction

EMBEDDED reconfigurable memory, which consists of a number of embedded array blocks (EABs), is becoming an essential component of high-density FPGAs [1]–[5], [7], [8], [12], [14]. Customized systems implemented on those FPGAs generally require high-speed system storages such as packet buffers, multimedia buffers and cache tag memory, to mention a few [12], [14]. Examples of fully customized circuits implemented on FPGAs can also be found in [6], where XOR-based decomposition methods to implement parity prediction circuits have been proposed for FPGAs. Implementing this embedded storage provides faster operations and lower instrumentation costs. Since the storage requirements of the customized systems vary in size and number, the FPGA memory architecture must be flexible to accommodate different independently addressable memories as well as different memory widths and depths. A number of recent commercial products, such as the Altera FLEX 10K family [4], the Xilinx Virtex and Spartan families [1], [2], the Actel 1200XL and 3200DX families [3], and Lattice ispLSI 6192 [5], contain a considerable number of large SRAM-based EABs.

TABLE I
Reconfigurable Memory Parameters

Parameter	N	B	W
Altera	3-16	2048	{1,2,4,8}
Xilinx	4-144	4096	{1,2,4,8,16}
Actel	8-16	256	{4,8}
Lattice	1	4608	{4,8}
Typical Range	1-144	256-16384	Several

Table I summarizes the parameters that define the FPGA reconfigurable memory array architecture, along with the values of the various commercial FPGAs as mentioned above. In the Xilinx FPGAs, for example, $B = 4096$ bits (i.e., the number of physical memory cells) and $W = \{1, 2, 4, 8, 16\}$ (i.e., the allowable logical memory width), meaning that each EAB can be logically configured as one of 4096 × 1, 2048 × 2, 1024 × 4, 512 × 8 and 256 × 16 logical memory configurations (depth and width, respectively).

Two or more EABs can be grouped into a single super-array with appropriate decoding [12]. For example, two 2-kilobit arrays with $W = \{1, 2, 4, 8\}$ can be combined into a super-array of 4 kilobits with $W = \{1, 2, 4, 8, 16\}$. Fig. 1 shows how the two 2-kilobit arrays combine to implement a 4096 × 1 super array. Note that the multiplexer required to select the desired bit from the two arrays can be implemented by the logic elements (LEs) of the FPGA.

In normal operational condition, each physical memory cell in an EAB is expected to be functional. This implies that each physical memory cell functions correctly all the time. Thus, its logical memory configurations are also supposed to function correctly all the time. FPGAs become increasingly critical in military and spacecraft instrumentation designs as the emphasis to decrease cost and mission development time continues, and hence, FPGAs for military or spacecraft instrumentation are used as part of the system in harsh operational environments. Under a harsh operational environment, each physical memory cell is prone to experience faults unexpectedly since it could be interfered by unpredictable environmental factors such as excessive radiation, temperature and physical impact [9], [10], [13]. These papers also discussed the performance degradation effects due to temporary or permanent faults. Furthermore, the issues associated with the SRAM-based reprogrammable FPGAs for space applications were discussed in [9].

Manuscript received May 29, 2001; revised September 11, 2002.

M. Choi is with the Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO 65409-0040 USA (e-mail: choim@umr.edu).

N. Park is with the Department of Computer Science, Oklahoma State University, Stillwater, OK 74078-1053 USA (e-mail: npark@a.cs.okstate.edu).

Digital Object Identifier 10.1109/TIM.2002.808046

0018-9456/02$17.00 © 2002 IEEE

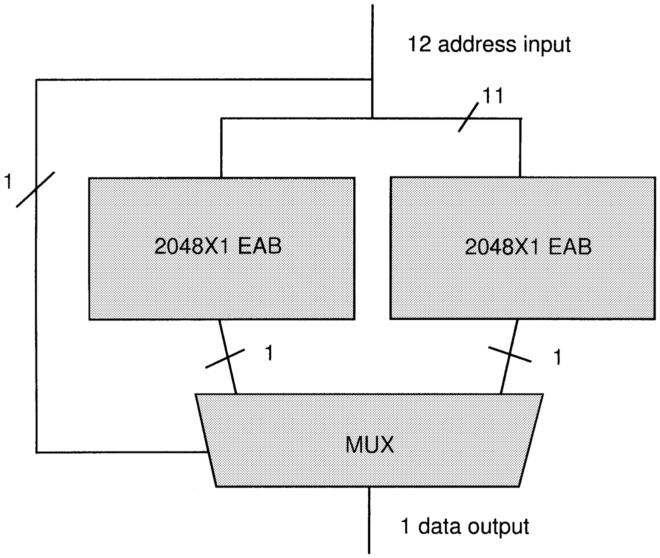

Fig. 1. Combining two 2048X1 arrays into one 4096X1 super-array.

This paper mainly focuses on the FPGA reconfigurable memory systems under harsh operational environments and its effects on the dynamic yields (i.e., the yields which take into account the inherited redundancy utilization for repair) of different target memory configurations. To tolerate the faulty cells due to the harsh environment, EABs can be tested, and the inherently redundant memory cells can be utilized to repair the faulty cells. Based on the measurement of the static and dynamic yields of the FPGA memory system, a few algorithms for selecting the most yield enhancing logical memory configuration which can accommodate the given target memory requirement among candidate configurations will be discussed.

The organization of this paper is as follows: In the next section, a review and preliminaries related to this work will be given. In Sections III and IV, both the static yield and dynamic yield of the FPGA reconfigurable memory system will be discussed, respectively. Then, the following three sections will introduce the dynamic yield assurance techniques with three possible inherited redundancy utilization techniques: bit-wise, word-wise and a combination of both. Based on the proposed assurance techniques, the optimal target memory configuration fitting algorithms for single configuration, sequential configurations and concurrent configurations will be proposed in Sections VII–X. Then, a discussion and conclusions are given in the final section.

II. Preliminaries

There are three basic architectural parameters to specify the FPGA reconfigurable memory system: N (the number of EABs), B (the number of bits per each EAB), and W (the set of allowable logical memory widths). Note that two or more EABs can be combined into one super-array in order to build a larger memory block as mentioned in the previous section. The total number of physical memory cells in the super-array is denoted by T_p, which is the result of multiplication of N and B

$$T_p = N \cdot B. \tag{1}$$

Since one logical memory word consists of W physical memory cells, the total number of physical memory cells and logical memory width determine the total number of logical memory words, which is denoted by T_l, also known as the depth of a logical memory configuration

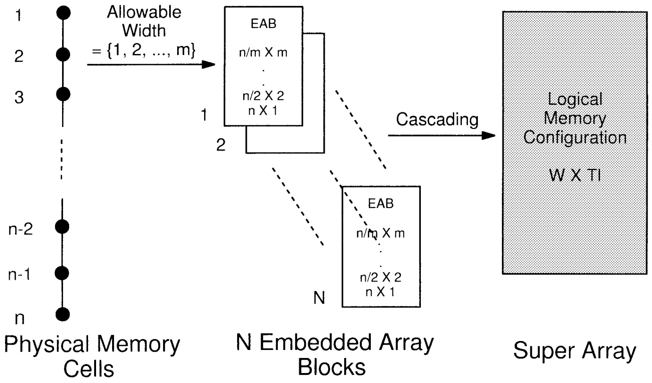

Fig. 2. Topology of the physical memory cells in EABs to implement a logical memory configuration.

$$T_l = \frac{T_p}{W}. \tag{2}$$

Fig. 2 illustrates a topology of the physical memory cells in EABs to implement a target logical memory configuration. There are n physical memory cells assigned to each EAB of $W = \{1, 2, \ldots, m\}$. Then, N EABs are cascaded to implement a super array of $W \times T_l$. The physical memory consists of T_p cells ($n \times N$ for Fig. 2), and the corresponding logical memory configuration consists of T_l words where each logical memory word contains W physical memory cells.

The following characteristics of the faults are assumed throughout this paper:

1) The expected number of faulty physical memory cells is assumed to be determined by λ.
2) The faults are also assumed to be randomly distributed over the physical memory space.
3) The yield of a single memory cell is independent of the yield of any other cells.

III. Static Yield of FPGA Reconfigurable Memory System

The yield of a single physical memory cell, Y_{pmc} can be formulated by

$$Y_{pmc} = e^{-\lambda} \tag{3}$$

as shown in [15]. Then, the probability of not having a failing cell in a T_p bit physical memory, denoted by Y_{pm}, can be formulated by using the binomial distribution, as $\binom{T_p}{0}(Y_{pmc})^{T_p}(1 - Y_{pmc})^0 = (Y_{pmc})^{T_p}$, that is

$$Y_{pm} = (Y_{pmc})^{T_p}. \tag{4}$$

As discussed in the previous section, the logical memory configurations of the FPGA memory system are to be determined by the architectural parameters, such as N, B, and W. Each logical memory word consists of W bits of physical memory cells. The probability of having a logical memory word without a faulty cell is $\binom{W}{0}(Y_{pmc})^W(1-Y_{pmc})^0 = (Y_{pmc})^W$. Let Y_{lmw} be the yield of a logical memory word, i.e.

$$Y_{lmw} = (Y_{pmc})^W. \tag{5}$$

The total number of logical memory words, T_l, can be used to formulate the yield of the entire logical memory, i.e.

$$Y_{lm} = (Y_{lmw})^{T_l}. \tag{6}$$

$Y_{lm} = (Y_{lmw})^{T_l} = ((Y_{pmc})^W)^{T_l} = (Y_{pmc})^{W \cdot T_l} = (Y_{pmc})^{T_p} = Y_{pm}$, using (4)–(6), which indicates that the yield of the physical memory of the given FPGA memory system (i.e., Y_{pm}) and the yield of the logical memory (i.e., Y_{lm}) are always equal regardless of the word length W. Thus, there is no gain or loss in yield by simply mapping a physical memory to a logical memory regardless of its width and depth.

Let the yield of the given FPGA memory system without inherited redundancy utilization be

$$Y_s = (Y_{pmc})^{T_p}. \tag{7}$$

Then, there are two possible ways to enhance the static yield Y_s of the FPGA reconfigurable memory system.

1) The static yield Y_s is determined by the average number of faults in a single cell λ. Reduction in λ could result in enhancement of the static yield Y_s. This relies on memory cell manufacturing technology. Note that enhancing the static yield Y_s through λ reduction is beyond the scope of this paper. Using more fault-tolerant flip-flops to harden the FPGA memory system is discussed in [10].
2) Redundant physical memory cells can also be used to enhance the static yield of the FPGA reconfigurable memory system. This strategy requires additional hardware such as spare memory cells, additional interconnection network, and a dedicated memory test circuit. It is obvious that these strategies are not practical to be employed in the majority of current FPGAs because of the hardware overhead. Also note that enhancing the static yield of the FPGA memory system through additional hardware is also beyond the scope of this paper. Application of additional hardware to harden the FPGA memory system is shown in [9].

IV. Dynamic Yield of FPGA Reconfigurable Memory System

FPGAs have a number of unique characteristics over application specific integrated circuits (ASICs). Among the characteristics, inherited redundancy and programmability of FPGAs can be exploited to obtain yield enhancement. One or more fully customized systems can be implemented on the FPGA, and each application may request a different memory configuration. Each requested memory configuration can be specified and represented by a target logical memory width (i.e., the number of bits per word, W_t) and a target logical memory depth (i.e., the number of words, D_t). Fig. 3(a) is an example of a target memory configuration of $W_t = 3$ and $D_t = 5$. If $W_t < W$ and/or $D_t < T_l$, $(W \cdot T_l) - (W_t \cdot D_t)$ physical memory cells are left unused and can be used as spare memory cells. Suppose that a super array of $W = 4$ and $T_l = 7$ is given. The shaded area in Fig. 3(b) shows a snapshot of the unused spare cells, and the number of spare cells, denoted by S, is $W \cdot T_l - W_t \cdot D_t$, which is 13 cells in this example.

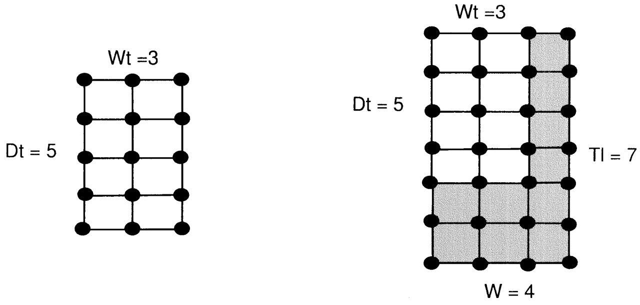

Fig. 3. Target memory configuration and spare memory cells.

The yield of a target memory configuration is referred to as dynamic yield which is denoted by DY. The dynamic yield of a target memory configuration can be enhanced by utilizing the spare cells in S and the reprogrammability of the FPGA. The DY of a target memory configuration without using spare cells in S, denoted by DY_{null}, is

$$DY_{null} = (Y_{pmc})^{W_t \cdot D_t} \tag{8}$$

where W_t is the target word length, D_t is the target depth, and Y_{pmc} is the physical memory cell yield.

V. Strategy 1—Dynamic Yield Enhancement Using Redundant Bits

The target memory configuration consists of D_t words where each word consists of W_t memory cells. Then, the dynamic yield of the target memory word, denoted by DY_{tmw}, can be formulated by

$$DY_{tmw} = (Y_{pmc})^{W_t}. \tag{9}$$

Observation 1: DY_{tmw} can be enhanced by spare bits placed in each word, if any. Fig. 4 shows a detailed illustration of the situation where each target word has R_b spare cells. Thus, the total number of physical memory cells in each target word becomes $W_t + R_b$. If W_t cells out of $W_t + R_b$ cells are fault-free, then the word is still operational. Now, the dynamic yield of the target memory word enhanced by spare bits, denoted by DY_{tmwrb}, can be formulated as

$$DY_{tmwrb} = \sum_{i=W_t}^{W_t+R_b} \binom{W_t+R_b}{i} (Y_{pmc})^i (1-Y_{pmc})^{W_t+R_b-i}. \tag{10}$$

Note that DY_{tmwrb} is always greater than or equal to DY_{tmw} if R_b is greater than 0, since the probability of having W_t fault-free cells out of $W_t + R_b$ cells is greater than or equal to the probability of having W_t fault-free cells out of W_t cells.

Then, the dynamic yield of the target memory system enhanced by redundant bits, denoted by DY_{tmrb}, can be identified as follows

$$DY_{tmrb} = (DY_{tmwrb})^{D_t}. \tag{11}$$

Note that DY_{tmrb} is always greater than or equal to DY_{null} because of the Observation 1 and (11).

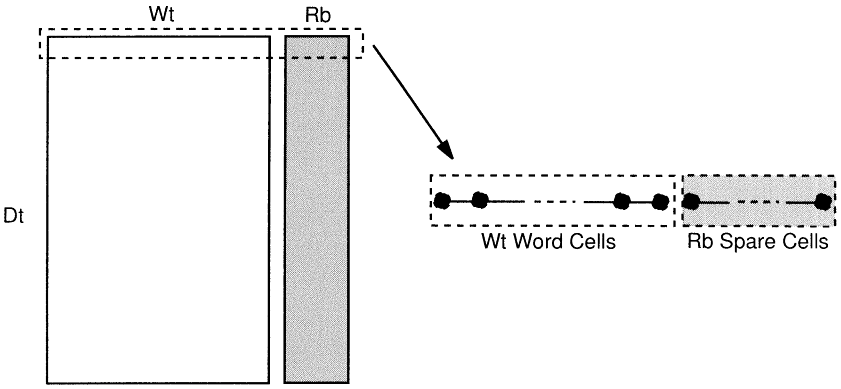

Fig. 4. Target memory configuration with redundant bits.

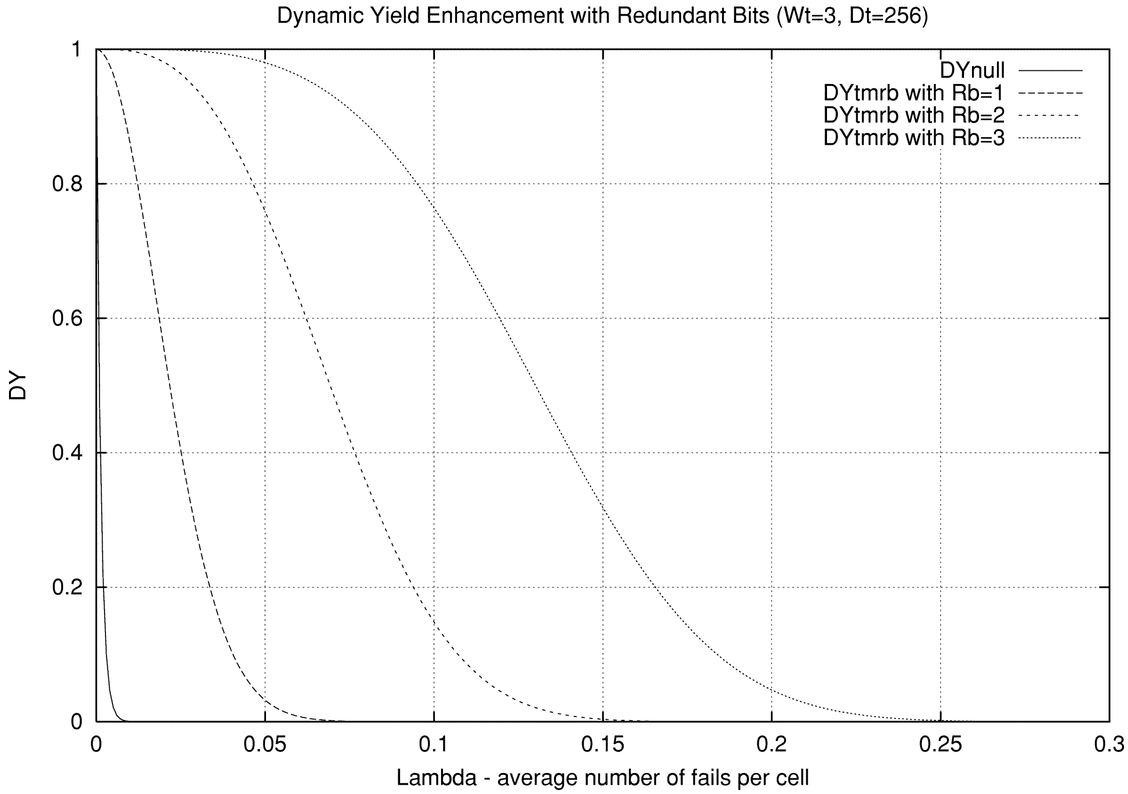

Fig. 5. Dynamic yield enhancement by redundant bits.

Simulation 1: Parametric simulation results on the dynamic yield enhancement of an experimental FPGA memory system are given in Fig. 5, where $W_t = 3$, $D_t = 256$ and $R_b = \{1, 2, 3\}$ are used. This simulation clearly reveals the effect of redundant-bit utilization for dynamic yield enhancement in which Y_{null} enhances as R_b increases. For instance, DY_{tmrb} of $R_b = 3$ at $\lambda = 0.041$ is greater than 99%, where $DY_{null} \approx 0\%$.

VI. Strategy 2—Dynamic Yield Enhancement Using Redundant Words

The target memory configuration consists of D_t number of words as shown in Fig. 3. If the given FPGA reconfigurable memory system can utilize inherently redundant words (in other words, $D_t < T_l$), the dynamic yield of the target memory configuration (i.e., DY_{null}) also can be enhanced by replacing faulty words with fault-free spare words. Suppose that R_W redundant words are utilized as spare words (see Fig. 6). Then, the

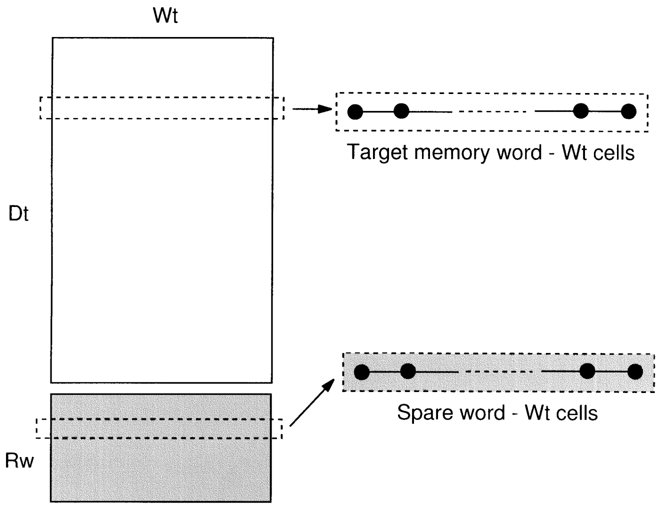

Fig. 6. Target memory configuration with redundant words.

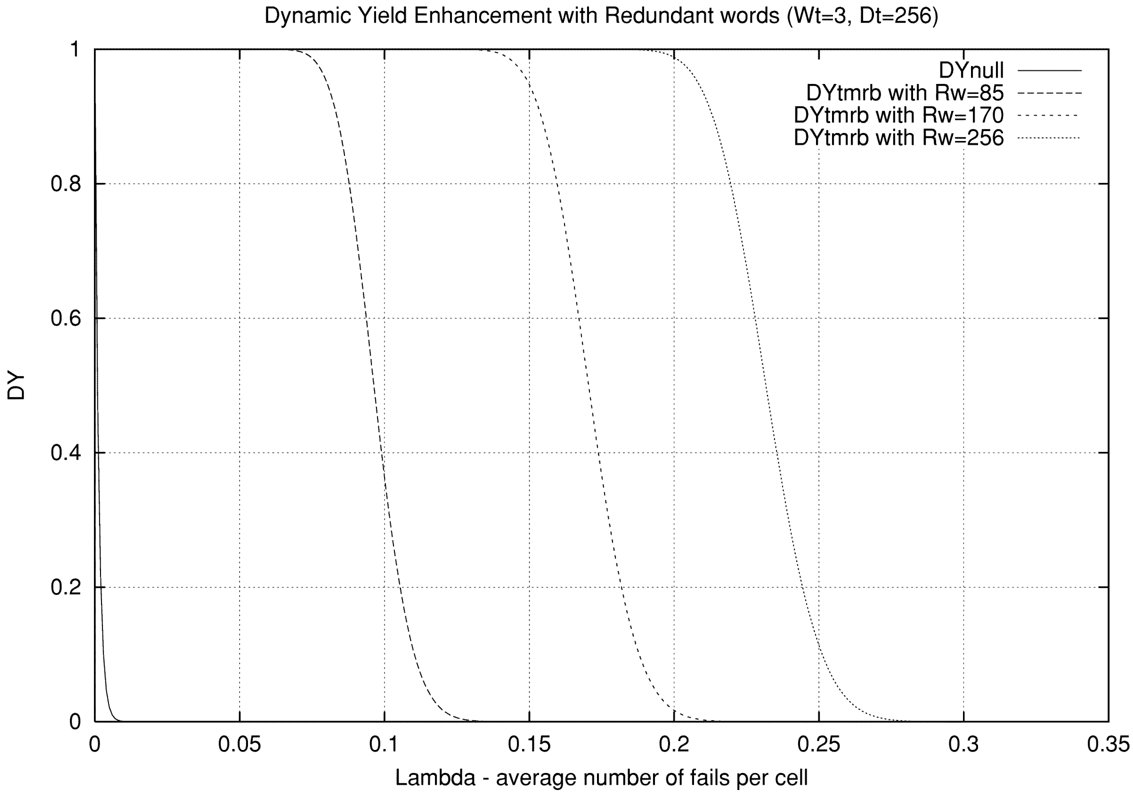

Fig. 7. Dynamic yield enhancement by redundant words.

dynamic yield of the target memory system enhanced by redundant words can be formulated as

$$DY_{tmrw} = \sum_{i=D_t}^{D_t+R_w} \binom{D_t+R_w}{i} \cdot (DY_{tmw})^i (1-DY_{tmw})^{D_t+R_w-i}. \quad (12)$$

Note that DY_{tmrw} is always greater than or equal to DY_{null}, since the probability of having D_t fault-free words out of $D_t + R_w$ words is greater than or equal to the probability of having D_t fault-free words out of D_t words.

Simulation 2: Parametric simulation results on the dynamic yield enhancement of an experimental FPGA memory system are given in Fig. 7, where $W_t = 3$, $D_t = 256$, and $R_w = \{85, 170, 256\}$ are used. This simulation also clearly reveals the effect of redundant-word utilization for dynamic yield enhancement. Dynamic yield enhancement using redundant bits and redundant words now can be compared as well, because the same number of spare cells as in the Simulation 1 is used (i.e., $R_b = \{1, 2, 3\}$ for the Simulation 1 and $R_w = \{85, 170, 256\}$ for the Simulation 2 use the same number of spare cells). Y_{null} enhances as R_w increases. For instance, DY_{tmrw} of $R_w = 256$ at $\lambda = 0.2$ is greater than 99% where $DY_{null} \approx 0$%.

Observation 2: Both the Simulations 1 and 2 show that the Strategy 2 performs better than the Strategy 1. This is because the Strategy 1 focuses on enhancing DY_{tmw}. However, this strategy is not so effective to enhance DY_{null}, since DY_{tmrb} is simply DY_{tmw} raised to the power of D_t. An insignificant degradation of DY_{tmw} could result in exponential degradation of DY_{tmrb}. On the other hand, the Strategy 2 performs better, since each additional spare word enhances DY_{null} linearly. According to the result of the Simulation 2, $R_w = 85$ successfully tolerates $\lambda = 0$ to 0.17, while the same amount of redundancy tolerates $\lambda = 0$ to 0.003 in the Simulation 1.

VII. Target Memory Configuration Fitting

The FPGA reconfigurable memory system can be configured into a number of logical memory configurations with different width and depth combinations. The logical memory width W (i.e., the number of cells in a word) and the number of logical memory words T_l (i.e., the depth) specify each logical memory configuration. If $T_p = 4096$ bits (i.e., the total number of physical memory bits) and $W = \{1, 2, 4, 8, 16\}$ (i.e., the allowable memory widths), then the FPGA reconfigurable memory system can be logically configured into one of 4096 × 1, 2048 × 2, 1024 × 4, 512 × 8 or 256 × 16 logical memory configurations. Finding the most suitable one from the candidate logical memory configurations for the given target memory configuration to maximally enhance the dynamic yield of the target memory configuration DY_{null} is referred to as "Target memory configuration fitting." Allocation of the most yield enhancing logical memory configuration for the given target memory configuration is an important issue, since the fitting directly determines the maximal dynamic yield enhancement. Because of the architectural constraints such as the limited number of programmable interconnections and the fixed allowable logical word width, allocating arbitrary spare physical memory cells one by one to repair arbitrary faulty logical memory cells requires excessive reconfiguration overhead which must be avoided by all means [12]. Therefore, two simple yet effective methods are to be investigated: bit-wise redundancy and word-wise redundancy.

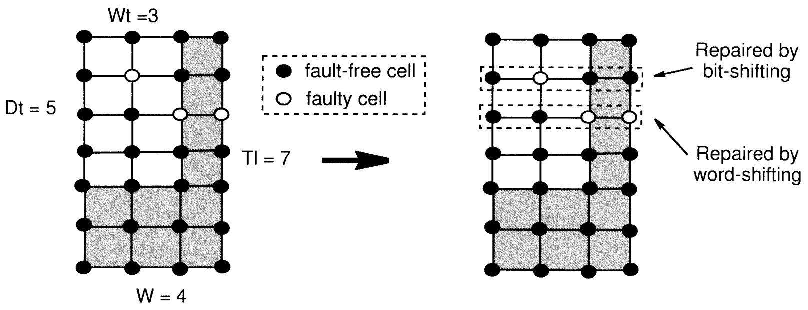

Fig. 8. Memory repair by bit-wise and word-wise redundancy utilization.

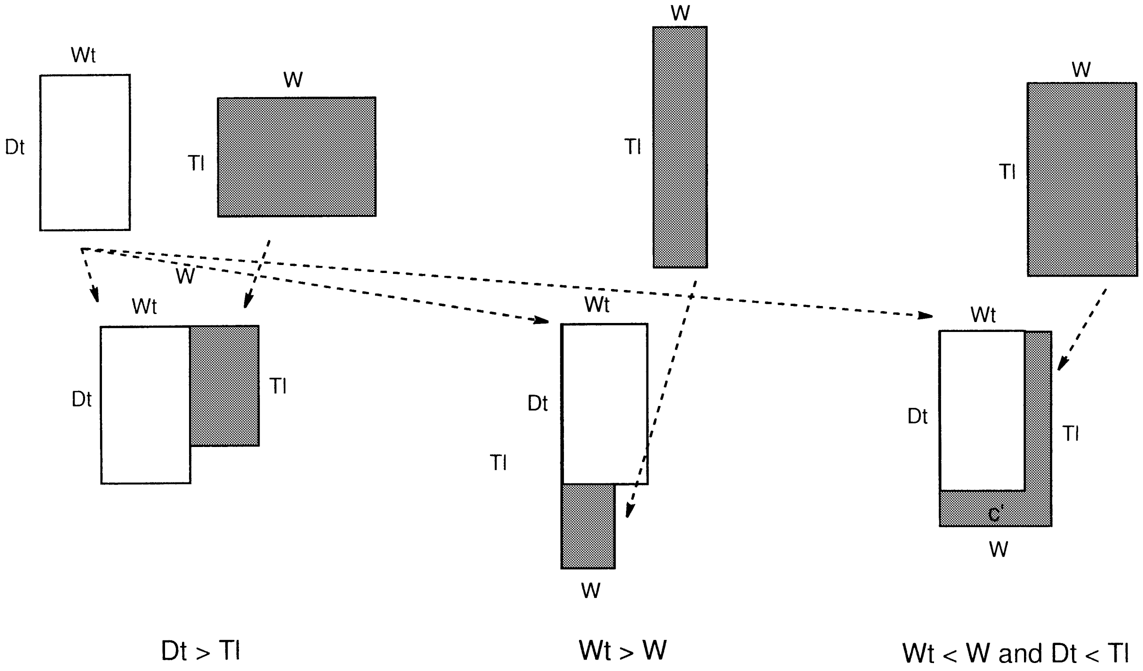

Fig. 9. Examples of target memory fitting.

If $W > W_t$, the unused bits in each word can be utilized as bit-wise redundancy. Likewise, if $T_l > D_t$, the unused words also can be used as word-wise redundancy. Fig. 8 shows an example of bit-wise and word-wise redundancy utilization in which a target memory configuration of $w_t = 3$ and $D_t = 5$ is fitted into a logical memory configuration of $W = 4$ and $T_l = 7$. If a word has a sufficient number of nonfaulty cells (i.e., $\geq W_t$), then the word can be repaired by shifting the faulty cells, which is referred to as bit-shifting as shown in the second word in Fig. 8(b). Otherwise (i.e., if the word does not have a sufficient number of fault-free cells), then the word still can be repaired by shifting itself, which is referred to as word-shifting as shown in the third word in Fig. 8(b). Both bit-wise shifting and word-wise shifting are implemented by using inherited logic and memory resources on demand; locations of the faulty bits and words are memorized and replaced by redundant bits and words. As a result, the given target memory configuration shown in Fig. 8(a) is successfully repaired by both bit-wise and word-wise redundancy utilization. Note that both bit-wise shifting and word-wise shifting can be implemented by considerable inherited redundancy utilization and, thus, should be minimal.

Another example is given in Fig. 9 to address the target memory configuration fitting issue. The blank rectangles represent the shape of the target memory configuration, and the shaded rectangles represent the shapes of candidate logical memory configurations. Note that the candidate logical memory shape in Fig. 9(a) can not be used, since $T_l < D_t$. Another candidate shape in Fig. 9(b) cannot be used either, because $W < W_t$. Fig. 9(c) shows an example of a suitable candidate logical memory shape in which both $T_l > D_t$ and $W > W_t$. Thus, the logical memory configuration in (c) successfully satisfies the requirement. The area designated by c' in Fig. 9(c) can be used as inherited redundancy.

Case Study 1: Suppose the FPGA reconfigurable memory system with $T_p = 4096$ bits and $W = \{1, 2, 4, 8, 16, 32\}$ and a target memory configuration of $W_t = 3$ and $D_t = 256$ are given. A list of all possible candidate logical memory configurations, denoted by C, then can be constructed as $C = \{(4096, 1), (2048, 2), (1024, 4), (512, 8), (256, 16), (128, 32)\}$. Both $C(1)$ and $C(2)$ are not suitable since they violate the requirement (in other words, $W < W_t$), and $C(6)$ also cannot be used since $T_l < D_t$. However, $C(3)$, $C(4)$, and $C(5)$ are suitable for the target memory configuration (i.e., $W \geq W_t$ and $T_l \geq D_t$). Those suitable logical memory configurations are denoted by C_s.

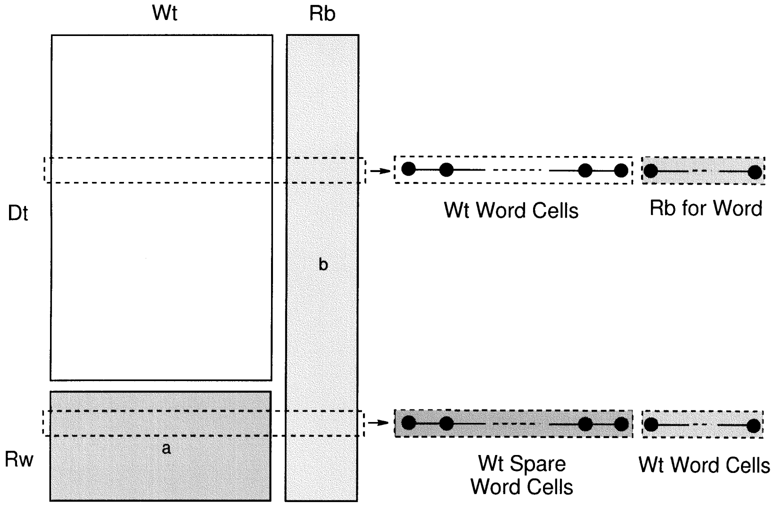

Fig. 10. Target memory configuration with redundant bits and words.

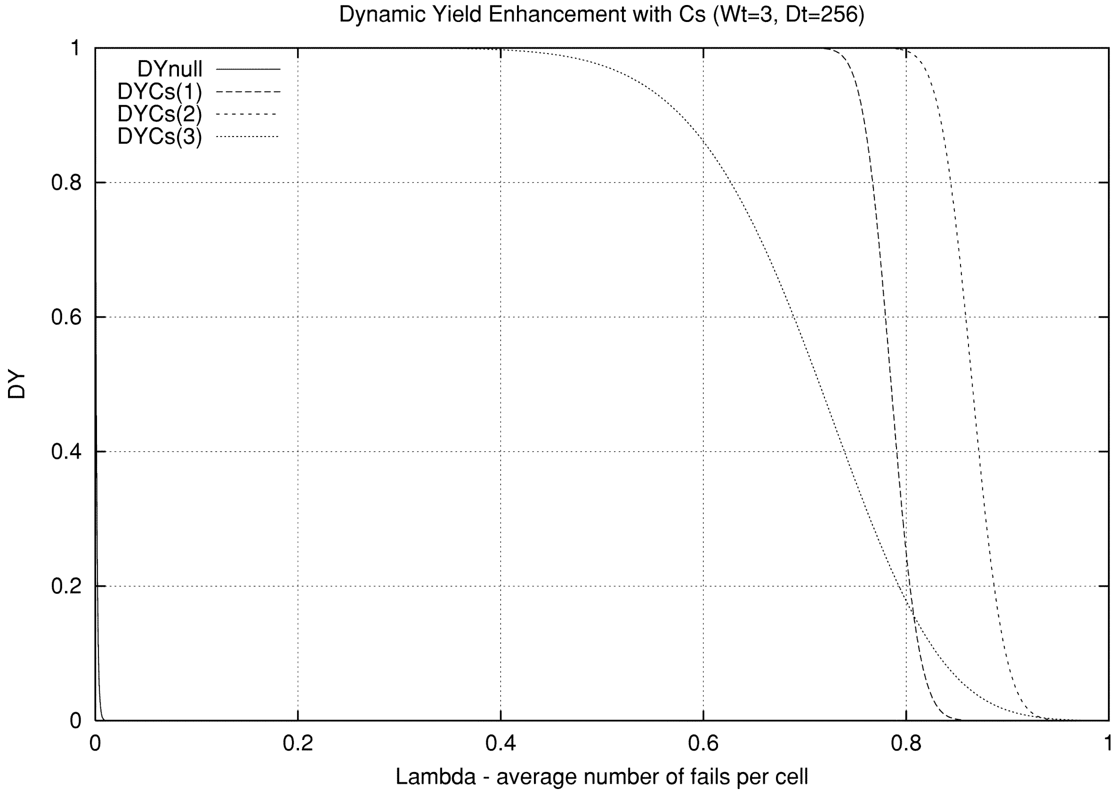

Fig. 11. Dynamic yield enhancement by C_s.

The dynamic yield of the target memory configuration enhanced by using spare cells of C_s, denoted by DY_{C_s}, is

$$DY_{C_s} = \sum_{i=D_t}^{D_t+R_w} \binom{D_t + R_w}{i} \cdot (DY_{tmwrb})^i (1 - DY_{tmwrb})^{D_t+R_w-i} \quad (13)$$

where DY_{tmw} is enhanced by R_b (Strategy 1) and DY_{null} is enhanced by R_w (Strategy 2). Note that the Strategy 2 requires partial rerouting of its interconnections where faulty words are replaced by fault-free spare words [15], [12], while the Strategy 1 does not require rerouting since fault-free spare bit(s) in a faulty logical word fix(es) faulty cells [15]. Thus, R_b is used first to fix the faulty words, and then R_w is used to replace the remaining faulty words to avoid the costly global rerouting. Fig. 10 shows how the spare cells can be divided into two areas designated by a and b in which the area a is used as redundant words and the area b is used as redundant bits.

Simulation 3: A parametric simulation result on a dynamic yield enhancement of the experimental FPGA memory system is shown in Fig. 11 where $W_t = 3$, $D_t = 256$ and $C_s = \{(1024, 4), (512, 8), (256, 16)\}$. For $C_s(3)$, $R_b = 13$ and $R_w = 0$. Thus, each word of the target memory is repaired by redundant bits, but no redundant word exists to enhance its dynamic yield further. That is why $DY_{C_s(3)}$ shows the least fault-tolerance among the candidates. For $C_s(1)$, $R_b = 1$

and $R_w = 768$. In this case, each word gets enhanced by a redundant bit ($R_b = 1$), and the overall memory gets enhanced by redundant words ($R_w = 768$). However, $DY_{C_s(1)}$ shows intermediate performance, since R_b is too small to enhance the dynamic yield of each word significantly. For $C_s(2)$, $R_b = 5$ and $R_w = 256$ show the best yield enhancement. The $DY_{C_s(2)}$ tolerates $\lambda = 0$ to 0.8.

VIII. Optimal Target Memory Configuration Fitting for Single Configuration

The issues associated with the target memory configuration fitting have been discussed in the previous section. Selecting the most dynamic yield enhancing one (denoted by $C_{optimal}$) among the candidate logical memory configurations (C) for a target memory configuration is referred to as optimal target memory configuration fitting (optimal fitting, for short). The optimal fitting guarantees the most dynamic yield enhancement. In Simulation 3, three suitable candidate logical memory configurations, $C_s(1)$, $C_s(2)$, and $C_s(3)$, were considered. The most dynamic yield enhancing logical memory configuration $C_{optimal}$ among them is $C_s(2)$, since it tolerates $\lambda = 0$ to 0.8, approximately. The following problem statement formally defines the optimal fitting.

Problem Statement 1: Among candidate logical memory configurations C, find the $C_{optimal}$ which maximally enhances the dynamic yield of a target memory configuration specified by W_t and D_t.

To effectively solve the problem stated above, the optimal fitting algorithm is proposed as follows:

```
Algorithm 1: Optimal target memory
configuration fitting.
  INPUT: $C$, $\lambda_t$ which is the target $\lambda$, $W_t$
  and $D_t$.
  STEP 1: $n = |C|$ where $n$ is the number of
  candidates. $DY_C = 0$ where $DY_C$ is dy-
  namic yield of the candidate. $C_{optimal} = \emptyset$
  where $C_{optimal}$ is the optimal candidate.
  STEP 2: FOR $i = 0$ TO $n$ BEGIN
        IF $W \geq W_t$ AND $T_l \geq D_t$ BEGIN
            Obtain $DY_{C(i)}$ using
            (11);
            IF $DY_{C(i)} > DY_C$ BEGIN
                $DY_C = DY_{C(i)}$;
                $C_{optimal} = C(i)$;
            END;
        END;
     END;
  OUTPUT: $C_{optimal}$.
```

The above algorithm requires n iterations, which is the number of candidates under consideration. Thus, the algorithm has $O(n)$ complexity. The correctness of the optimal fitting algorithm can be easily proved. The algorithm simply rejects candidates if they have $W < W_t$ or $T_l < D_t$. Then, the algorithm obtains the dynamic yields of pre-approved candidates and compares them to get $C_{optimal}$. Since the dynamic yield function is a decreasing function with respect to λ, $C_{optimal}$ tolerates $\lambda = 0$ to λ_t better than the other candidates. Thus, if $DY_{C_{optimal}} = 1$, then $C_{optimal}$ is guaranteed to tolerate $\lambda = 0$ to λ_t. The following case study further shows the effectiveness and correctness of the optimal fitting algorithm.

TABLE II
Sample Results From the Optimal Fitting Algorithm

λ_t	$DY_{C_s(1)}$	$DY_{C_s(2)}$	$DY_{C_s(3)}$	$C_{optimal}$
0.684	1.0	1.0	0.619862	$C_s(1)$
0.746	0.966653	1.0	0.372272	$C_s(2)$

Case Study 2: Suppose the FPGA reconfigurable memory system with $T_p = 4096$ bits and $W = \{1, 2, 4, 8, 16, 32\}$, and a target memory configuration of $W_t = 3$ and $D_t = 256$ are given. A list of suitable candidate logical memory configurations is then $C_s = \{(1024, 4), (512, 8), (256, 16)\}$. Table II shows how the optimal fitting algorithm reacts when different λ_ts are applied.

When $\lambda_t = 0.684$, the optimal fitting algorithm selects $C_{optimal} = C_s(1)$. $DY_{C_s(2)}$ is also 1.0, but the algorithm is supposed to choose the first one since there is no gain by selecting the second one. When $\lambda_t = 0.746$ is applied, the optimal fitting algorithm selects $C_s(2)$ as $C_{optimal}$, since $DY_{C_s(2)} = 1.0$, which is the most dynamic yield enhancing candidate.

IX. Optimal Target Memory Configuration Fitting for Sequential Configurations

The FPGA can be reconfigured as many times as needed to implement numerous configurations. More than one application can be implemented on the FPGA both sequentially and concurrently [11]. If the FPGA is required to be sequentially reconfigured over the time to implement the configurations, the situation is referred to as sequential reconfiguration system (SRS). Because the whole reconfiguration process is sequential, each configuration is disjoint from the other configurations. Thus, the dynamic yield of SRS, which is denoted by DY_{SRS}, can be formulated as

$$DY_{SRS} = \frac{\sum_{i=1}^{m} DY(i)}{m} \tag{14}$$

where m is the number of total configurations, and $DY(i)$ is the dynamic yield of the i_{th} configuration. Note that sequential utilization of the optimal fitting always guarantees the maximum DY_{SRS}. The following algorithm finds the optimal fitting for SRS which maximizes DY_{SRS}.

```
Algorithm 2: Optimal target memory fitting
for $SRS$.
  INPUT: $C$, $\lambda_t$ which is the target $\lambda$, $W_t$
  and $D_t$ of each configuration, and $m$
  which is the number of sequential con-
  figurations.
  STEP 1: FOR $i = 0$ TO $m$ BEGIN
        Using the Algorithm 1,
        find the optimal fitting for $i_{th}$
```

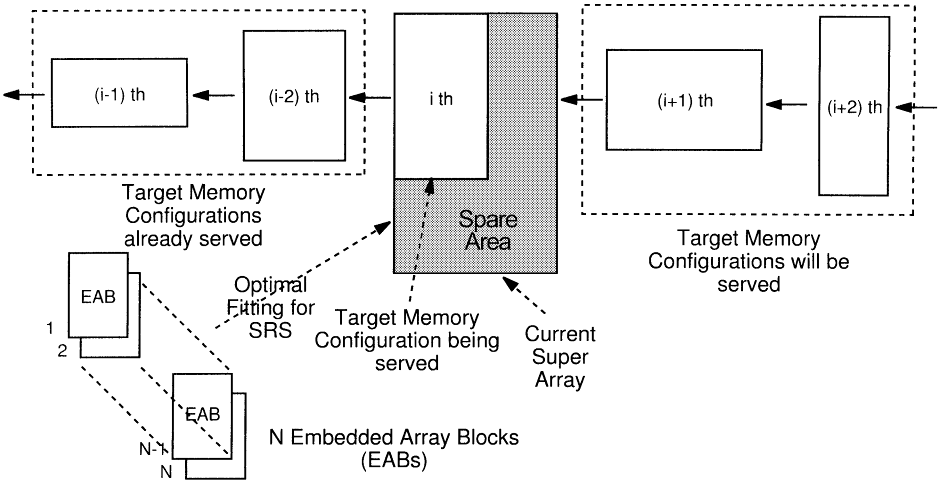
(i-1) th
(i-2) th
i th
(i+1) th
(i+2) th
Target Memory Configurations already served
Spare Area
Target Memory Configurations will be served
EAB
1
2
Optimal Fitting for SRS
Target Memory Configuration being served
Current Super Array
EAB
N-1
N
N Embedded Array Blocks (EABs)

Fig. 12. Illustration of the optimal fitting algorithm for SRS.

```
        configuration
        and store it in $C_{optimal}(i)$;
    END;
OUTPUT: $C_{optimal}(1)$ ... $C_{optimal}(m)$.
```

The Algorithm 2 outputs an array of the optimal candidates where $C_{optimal}(i)$ is the optimal candidate for the i_{th} configuration. An illustration of the optimal fitting algorithm for SRS is given in Fig. 12 where N EABs are reconfigured, as a number of target memory configurations are mapped into the FPGA sequentially. Some preliminary results are shown in Table III. The given EABs are configured into a super array of $T_p = 4096$ and $W = \{1, 2, 4, 8, 16, 32\}$, and three sequential target memory configurations (400 × 3, 600 × 3 and 100 × 5) with corresponding $\lambda_t = 0.6$, 0.3, and 0.8 are fitted into the super array. As a result, three optimal super array configurations (512 × 8, 1024 × 4 and 256 × 16) are selected by the algorithm.

Note that the asymptotic complexity of the Algorithm 2 is $O(n \cdot m)$, since the Algorithm 1 requires $O(n)$ and the Algorithm 2 requires m (i.e., the number of sequential configurations) iterations over the Algorithm 1.

X. Optimal Target Memory Configuration Fitting for Concurrent Configurations

Multiple configurations can be successfully operational on the FPGA concurrently [11], and this situation is referred to as concurrent reconfiguration system (CRS). Since multiple configurations reside in the FPGA and are functionally concurrent, the configurations are temporarily dependent on each other. Thus, the dynamic yield of CRS, which is denoted by DY_{CRS}, can be calculated by

$$DY_{CRS} = \prod_{i=1}^{m} DY(i). \tag{15}$$

The CRS requires appropriate concurrent mappings of multiple target memory configurations on the FPGA. Thus, the given FPGA memory system must be partitioned into multiple super arrays to serve the multiple target memory configurations concurrently. The dynamic yield of CRS, which is denoted by DY_{CRS}, can be optimized if carefully sized super arrays are assigned to the target memory configurations. The following observation identifies some important issues of choosing suitable candidates for CRS.

TABLE III
Sample Results From the Optimal Fitting Algorithm for SRS

Target Memory	400 × 3	600 × 3	100 × 5
λ_t	0.6	0.3	0.8
4096 × 1	N/A	N/A	N/A
2048 × 2	N/A	N/A	N/A
1024 × 4	0.470415	1	N/A
512 × 8	1	N/A	0.999710
256 × 16	N/A	N/A	1
128 × 32	N/A	N/A	1
$C_{optimal}$	512 × 8	1024 × 4	256 × 16

Observation 3: Suppose the FPGA memory system with 8 EABs ($N = 8$) and three concurrent target memory configurations which utilize 1, 3, and 3 EABs are given. There are five ways to partition the given 8 EABs into three groups: i.e., 1-1-6, 1-2-5, 1-3-4, 2-2-4, and 2-3-3 where each number represents the number of EABs in each super array and each super array is delimited by "-." Then, only 1-3-4 and 2-3-3 can be considered as the suitable candidates, since the other candidates are physically not suitable for the given target memory configuration of 1-3-3. Then, the most dynamic yield enhancing candidate (i.e., the optimal candidate) can be chosen from the eligible candidates of 1-3-4 and 2-3-3.

The following algorithm finds the optimal target memory fitting for CRS based on the unsuitable candidate pruning criteria identified in the Observation 3.

```
Algorithm 3: Optimal target memory fitting
for $CRS$.
    INPUT: $C$, $\lambda_t$ which is the target $\lambda$,
    $W_t$ and $D_t$ of each configuration, and $l$
    which is the number of concurrent con-
    figurations.
    STEP 1: Construct suitable candidates as
    described in the Observation 3 using the
    given number of EABs while unsuitable
    candidates (e.g., 1-1-6, 1-2-5 and 2-2-4
    in Observation 3) are pruned. Store the
```

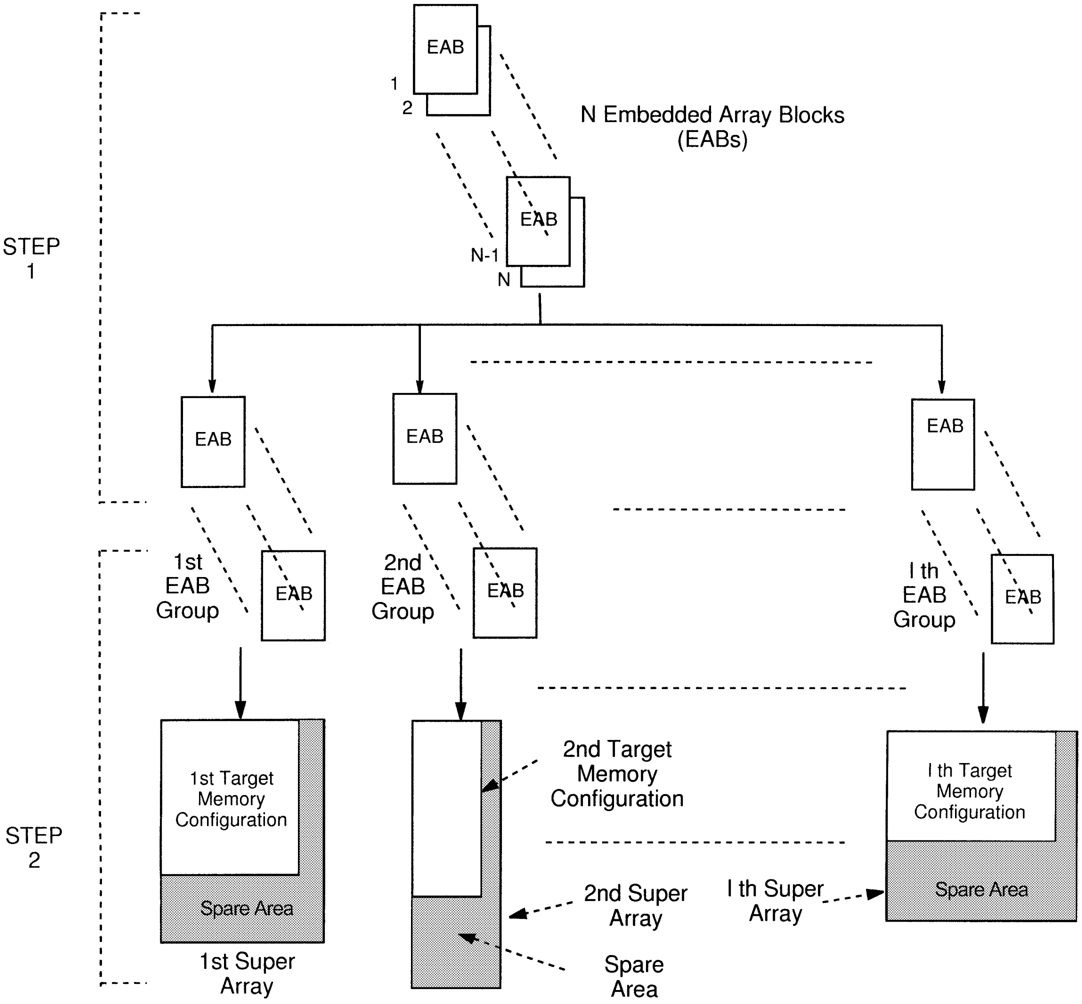

Fig. 13. Illustration of the optimal fitting algorithm for CRS.

```
remaining suitable candidates (e.g.,
1-3-4 and 2-3-3 in Observation 3) in an
array $C$ and $|C|$ in $k$. Initialize $temp = 0$.
STEP 2: FOR $i = 1$ TO $k$ BEGIN
      FOR $j = 1$ TO $l$
          Using the Algorithm 1,
          find the optimal fitting
          for $j_{th}$ configuration to
          $j_{th}$ super array of $C(i)$;
      END;
      Store the optimal fitting
      results of
      $C(i)$ in $FIT_{result}$;
      IF $(DY_{CRS}(FIT_{result}) > temp)$ THEN
      BEGIN
          $C_{optimal} = FIT_{result}$;
          $temp = DY_{CRS}(FIT_{result})$;
      END;
    END;
OUTPUT: $C_{optimal}$.
```

Upon termination, the Algorithm 3 outputs the most dynamic yield enhancing candidate. Fig. 13 shows how the optimal fitting algorithm for CRS works where EABs are partitioned into l groups by the STEP 1 and each concurrent target memory configuration gets optimally fitted into each group of EABs by the STEP 2. Asymptotic complexity of the Algorithm 3 is $O(n \cdot l \cdot k)$, since it has a nested loop, where the inner loop iterates l (i.e., the number of concurrent configurations) times, and the outer loop iterates k (i.e., the number of candidates) times, and the Algorithm 1 is embedded in the inner loop. Note that each candidate consists of l super-arrays, and the target concurrent configurations are fitted into the super-arrays by the Algorithm 1.

The following observation traces the Algorithm 3 to show how it works.

Observation 4: Suppose the FPGA reconfigurable memory system of $N = 8$, $B = 1024$ and $W = \{1, 2, 4, 8\}$ is given, and three concurrent target memory configurations of 700 × 4, 300 × 3, and 200 × 5 with corresponding $\lambda_t = 0.07$, 0.3, and 0.1, respectively, are to fit into the memory system as shown in Fig. 14. The STEP 1 of the algorithm partitions the given EABs into 3 groups, since three concurrent target memory configurations are given. There exist five different candidates: 6-1-1, 5-2-1, 4-3-1, 4-2-2, and 3-3-2. Among those candidates, only 4-2-2 and 3-3-2 are initially identified as the suitable candidates. The other unsuitable candidates (6-1-1, 5-2-1, and 4-3-1) are

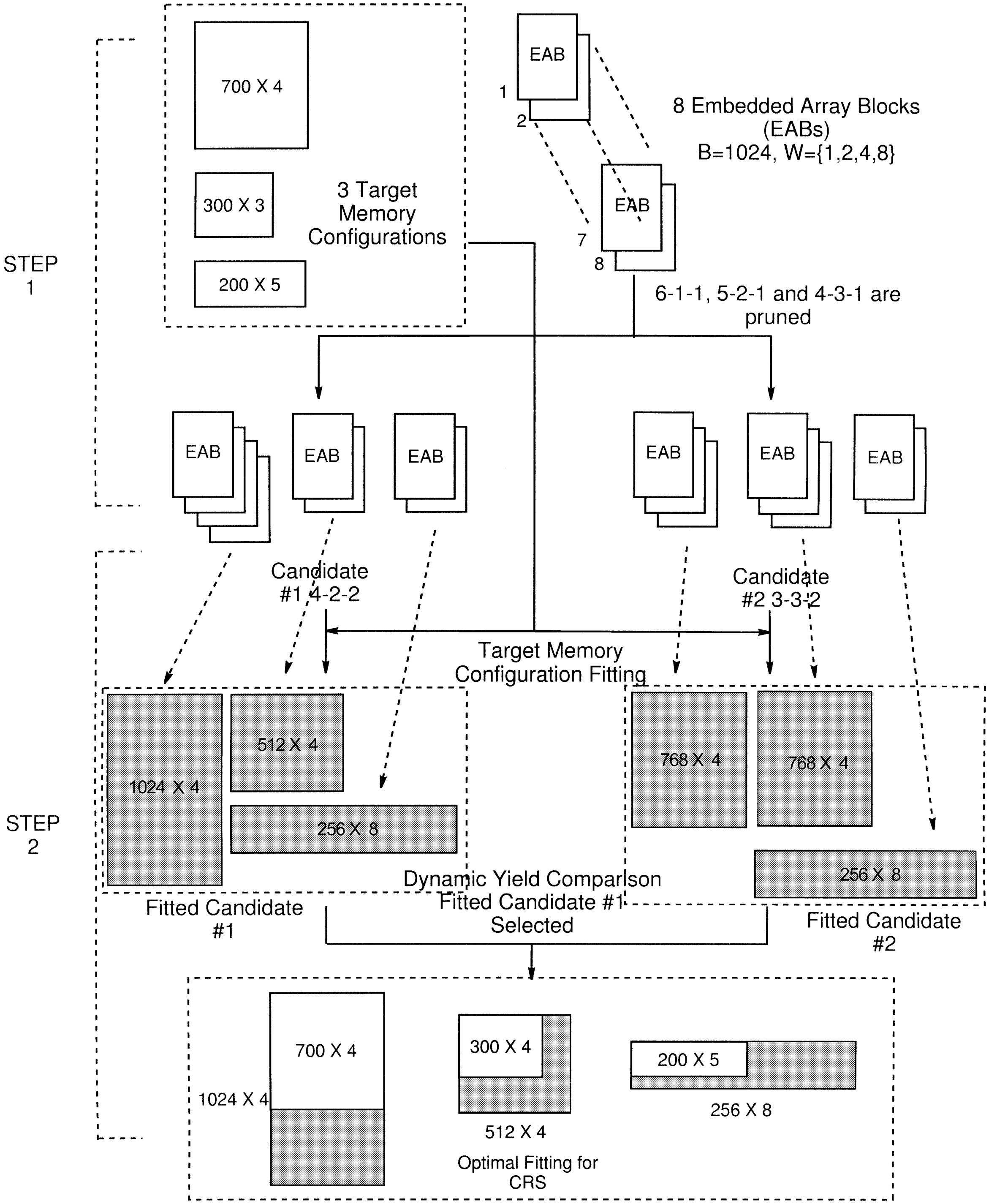

Fig. 14. How the optimal fitting algorithm for CRS works.

pruned by the STEP 1. Then, the STEP 2 searches for the possible fitting solutions using the suitable candidates (4-2-2 and 3-3-2). As a result, the fitted candidate #1 (1024 × 4, 512 × 4 and 256 × 8) and #2 (768 × 4, 768 × 4 and 256 × 8) are constructed. Finally, the STEP 2 compares the dynamic yields of the candidates using given λ_t values. The resulting super arrays of 1024 × 4, 512 × 4, and 256 × 8 are guaranteed to be suitable for the given concurrent target memory configurations and to be fault-tolerant for the given λ_t.

XI. Conclusion

This paper has proposed various new ways to enhance the dynamic yield of the embedded reconfiguration memory system with randomly distributed memory cell faults. Fundamental assurance techniques for the yield measurement and enhancement and their practical applications have been discussed. Also, various simulations have been conducted to characterize and develop the yield enhancement strategies. The balanced combination of redundant bits and words has shown effective fault-tolerance, since the target word yield gets enhanced by redundant bits, and the overall memory yield gets enhanced by redundant words. Then, the significance of the optimal fitting which finds the most yield enhancing one among the candidate logical memory configurations has also been discussed. Finally, novel optimal fitting algorithms which have moderate complexity of $O(n)$ (for single configuration fitting), $O(n \cdot m)$ for SRS fitting, and $O(n \cdot l \cdot k)$ for CRS fitting have been introduced. The optimal fitting algorithms can be applied to most of the FPGAs to make them fault-tolerant without modifying their hardware, since they exploit inherited redundancy and programmability of the FPGAs. The proposed algorithms ultimately establish a solid theoretical foundation to realize practical and specific FPGA implementations under various vendor-dependent constraints.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments.

REFERENCES

[1] "Spartan-II family advance product specification," Xilinx, Inc., Jan. 2000.

[2] "Virtex-E family advance product specification," Xilinx, Inc., Feb. 2000.

[3] "Integrator Series FPGAs: 1200XL and 3200DX families data sheet," Actel, Inc., Jan. 1998.

[4] "Flex 10K embedded programmable logic family data sheet," Altera Co., San Jose, CA, Jan. 1999.

[5] "ispLSI 6192 high density programmable logic with dedicated memory and register/counter modules data sheet," Lattice Semiconductor Co., Sunnyvale, CA, May 1999.

[6] S.-B. Ko, T. Xia, and J.-C. Lo, "Efficient parity prediction in FPGA," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2001, pp. 176–181.

[7] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Enhanced FPGA reliability through efficient run-time fault reconfiguration," *IEEE Trans. Reliability*, vol. 49, pp. 296–304, Sept. 2000.

[8] W. K. Huang, F. J. Meyer, and F. Lombardi, "An approach for detecting multiple faulty FPGA logic blocks," *IEEE Trans. Comput.*, vol. 49, pp. 48–54, Jan. 2000.

[9] J. Wang, R. Katz, J. Sun, B. Cronquist, J. McCollum, T. Speers, and W. Plants, "SRAM based re-programmable FPGA for space applications," *IEEE Trans. Nucl. Sci.*, pt. 1, vol. 46, Dec. 1999.

[10] R. Katz, J. Wang, J. McCollum, and B. Cronquist, "The impact of software and CAE tools on SEU in field programmable gate arrays," *IEEE Trans. Nucl. Sci.*, pt. 1, vol. 46, Dec. 1999.

[11] J. Jean, K. Tomko, V. Yavagal, J. Shah, and R. Cook, "Dynamic reconfiguration to support concurrent applications," *IEEE Trans. Comput.*, vol. 48, June 1999.

[12] S. Wilton, J. Rose, and Z. Vranesic, "The memory/logic interface in FPGA's with large embedded memory arrays," *IEEE Trans. VLSI Syst.*, vol. 7, Mar. 1999.

[13] R. Katz, K. LaBel, J. J. Wang, B. Cronquist, R. Koga, S. Penzin, and G. Swift, "Radiation effects on current field programmable technologies," *IEEE Trans. Nucl. Sci.*, vol. 44, Dec. 1997.

[14] W. K. Huang, F. J. Meyer, N. Park, and F. Lombardi, "Testing memory modules in SRAM-based configurable FPGAs," in *Proc. Int. Workshop Memory Technol., Design, Testing*, Aug. 1997, pp. 79–86.

[15] C. H. Stapper and H.-S. Lee, "Synergistic fault-tolerence for memory chips," *IEEE Trans. Comput.*, vol. 41, Sept. 1992.

Minsu Choi (M'02) received the B.S., M.S., and Ph.D. degrees in computer science from Oklahoma State University, Stillwater, in 1995, 1998, and 2002, respectively.

He is currently with Department of Electrical and Computer Engineering, University of Missouri, Rolla, as an Assistant Professor. His research mainly focuses on computer architecture and VLSI, embedded systems, fault tolerance, testing, quality assurance, reliability modeling and analysis, configurable computing, parallel and distributed systems, and dependable instrumentation and measurement.

Dr. Choi was the recipient of the Don and Sheley Fisher Scholarship in 2000, the Korean Consulate Honor Scholarship in 2001, and the Graduate Research Excellence Award in 2002. He is a member of Golden Key National Honor Society.

Nohpill Park (M'99) received the B.S. degree in 1987 and the M.S. degree in computer science in 1989 from Seoul National University, Seoul, Korea. He received the Ph.D. degree in 1997 from the Department of Computer Science, Texas A&M University, College Station.

He is currently an Assistant Professor in the Computer Science Department, Oklahoma State University, Stillwater. His research interests include computer architecture, defect and fault tolerant systems, testing and quality assurance of digital systems, parallel and distributed computer systems, multichip module systems, programmable digital systems, and reliable digital instrumentation.