ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Architecture for BISR (Built-In Self-Repair) Using Extended Essential Spare Pivoting (EESP) Algorithm

A.Vishnupriya
M.Tech,
Dept of ECE,
SVCET, Chittoor.

Mr.J.Madan Kumar
Assistant Professor,
Dept of ECE,
SVCET, Chittoor.

ABSTRACT: In this paper, BISR (built-in self-repair) techniques with hierarchical redundancy architecture are proposed for word-oriented embedded memories. Our BISR circuit consists of a built-in self-test (BIST) module and a built-in redundancy-analysis (BIRA) module. Spare words, spare rows, and spare columns are added into the memory cores as redundancy. However, the spare rows and spare columns are virtually divided into spare row blocks and spare column group blocks. The address reconfiguration is performed at row block or column group block level instead of the traditional row or column level. An extended essential spare pivoting (EESP) algorithm is proposed for redundancy analysis based on the proposed redundancy organization. A practical 16K u 32 SRAM with BISR circuitry is designed and implemented. Experimental results show that we can obtain a higher repair rate with negligible area overhead (2.56%) of the BISR circuit for a 1024K u 2048-bit SRAM chip.

Index Terms: Built-In Self-Test (BIST), BISR (Built-In Self-Repair), Built-In Redundancy-Analysis (BIRA), Extended Essential Spare Pivoting (EESP).

1.INTRODUCTION :

In the past, many redundancy mechanisms can be used to increase the reliability and yield of embedded memories .These approaches can be categorize as
1.Redundant row or redundant column: Redundant rows or columns are added into the memory array. One of the redundant rows/columns is used to replace the faulty row/column.
2.Redundant row and redundant column [2-6]: Bothredundant rows and redundant columns are incorporated into the memory array. When a faulty cell is detected,either a redundant row or a redundant column can beused to replace the faulty cells. It is more efficient thanthe first approach when multiple faulty cells exist in thememory array. However, the optimal redundancy allocation problem becomes NP-complete [7].There are many built-in redundancy analysis/built-in selfrepair (BIRA/BISR) techniquesproposed for embeddedmemories in the past [6,7,8]. However, these techniques need long computation time to find the optimal solution dueto the exponential complexity of the algorithm. In this paper,in order to alleviate the drawbacks of the above tworedundancy mechanisms, a new redundant mechanism is adopted in this paper.

Spare words, spare rows, and spare columns are added into the word-oriented memory cores asredundancy. However, the spare rows and spare columns arevirtually divided into spare row blocks and spare column group blocks. Moreover, because we focus on word-oriented memories, redundant words are also used as redundancy.The address reconfiguration is performed at different redundancy level instead of the traditional row or column level. An extended essential spare pivoting (EESP) algorithm is proposed for redundancy analysis based on the proposed redundancy organization. A practical 16K u 32 SRAM with BISR circuitry is designedandimplemented.Experimental results show that we can obtain a higher repair rate with negligible area overhead (2.56%) of the BISR circuit for a 1024 u 2048-bit SRAM chip.The organization of this paper is described as follows.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Section 2 introduces the redundancy organization adopted inthis paper. Section 3 describes the extended essential spare pivoting algorithm.. Section 4 gives the BISR architecture and procedures. Section 5 gives a practical design example.Repair efficiency and hardware overhead are analyzed inSection 6. Finally, some conclusions are given in Section 7.As aforementioned, most of the BISR schemes need a local bitmap to store fault information. Since the local bitmap is a cache-like storage element and it is only used in the test phase, we can use it as a spare element in normal operation phase. In this paper, therefore, we propose a high repairefficiency (HRE)-BISR scheme by reusing local bitmap asthe bit redundancy. Thus, the Available redundancy Resourceis increased, which can boost the repair efficiency. An RAalgorithm for a RAM with 2-D and bit redundancies is proposed to allocate redundancies as well.

2Proposed Redundancy Mechanism:

Fig. 1 shows a RAM cell array with redundancy rows and columns. The 32 u 4-bit RAM contains one spare row (SR0),two spare words (SW0, SW1), and two spare columns (SC0, SC1). The normal memory array is also divided into two rowbanks (RB0, RB1) and two column banks (CB0, CB1) [5, 6].Therefore, the spare row is further divided into two spare row blocks (SRB0, SRB1). The spare columns are also divided into four spare column blocks (SCB0, SCB1, SCB2,and SCB3). The spare column blocks contained in the same column bank are combined as spare column group blocks(SCGB). For example, in Fig. 1, the group size ofSCGB0 is 2,which contains SCB0 and SCB1. The complexity of the address remapping circuit can be reduced due to the SCGB architecture.

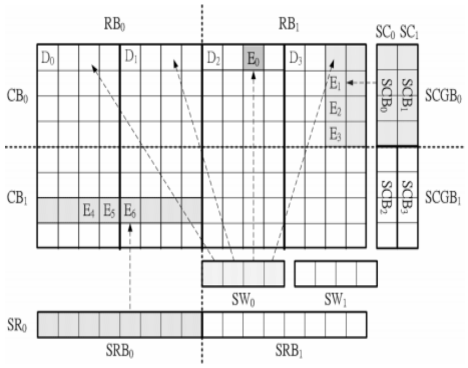


Fig. 1: The proposed redundancy organization.

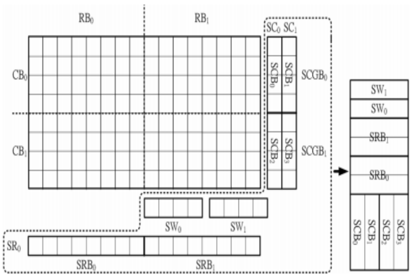


Fig. 2: Architecture of redundancy.

Table 1: TRUTH TABLE OF MBE SCHEME:

Fig. 3 shows a operation flow between the BIST and BIRA. The BIST generates test patterns for the RAM. When a fault in the RAM is detected, the information of the fault is sent to the BIRA. The BIRA first checks if the fault has been stored in the bitmap or repaired. If not, the fault is stored in the bitmap. Then, the BIRA checks if the bitmap is full or not.

A bitmap is called full if either its CAR or RAR has no space for storing a fault. If yes, the BIRA allocates the redundancies according to the stored fault information and the implemented RA algorithm. If the test process is completed, the BIRA checks if the bitmap is empty or not. A bitmap is called empty if no fault information of faulty cells is stored in the bitmap. If not, the BIRA performs the redundancy allocation process to repair the remained faults.

Once the BIRA process is completed and the RAM is repairable, the repaired addresses are stored in the fuse macro. Fig. 4 shows a simplified block diagram of a BIRA with k × l-bit bitmap, where some functional blocks are not shown. Some of interaction signals between the BIST and BIRA are shown in the figure, which are faulty address (FA), syndrome (SYN), and repair (REP).

Once the BIST detects a fault, the corresponding FA and Hamming syndrome are sent to the BIRA through the FA and SYN, where Hamming syndrome is defined as the modulo-2 sum of the fault-free data output vector and the output vector from the RAM under test [8]. The signal REP is used to indicate the RA result of the BIRA is repairable or unrepairable.

ISSN No: 2348-4845

International Journal & Magazine of Engineering, Technology, Management and Research

A Peer Reviewed Open Access International Journal

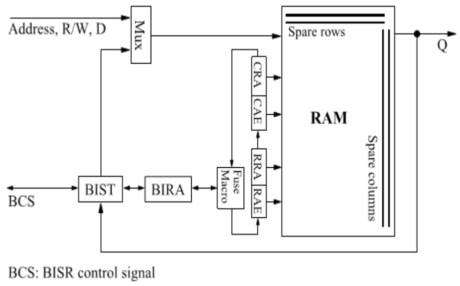


Fig Typical BISR scheme for a repairable RAM [16]

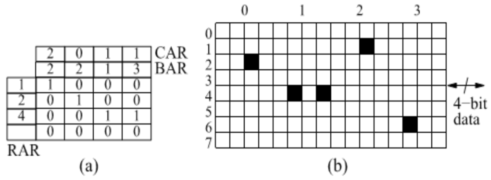


Fig (a) Conceptual diagram of a 4 × 4-bit bitmap. (b) Exemplary 8 × 4 × 4-bit RAM with five faulty cells.

local bitmap of the BIRA. Section IV describes the proposed RA algorithm for a RAM with spare rows, spare columns, and spare bits. Section V shows simulation and analysis results. Finally, Section VI briefly concludes this paper

II. OVERVIEW OF A BISR SCHEME WITH BIRA

Repairable RAM: A RAM with redundancies and reconfiguration circuit is called as a repairable RAM. Figure depicts an example of an 8*8 bit-oriented RAM with 1 spare row and 1 spare column. If a spare row is allocated to replace a defective row, then the row address of the defective row is called row repair address (RRA). Then a decoder decodes the RRA into control signals for switching row multiplexers to skip the defective row if the row address enable (RAE) signal is asserted. The reconfiguration of the defective column and the spare column is performed in a similar way, i.e., give a column repair addresses (CRA) and assert the column address enable signal to repair the defective column using the spare column.

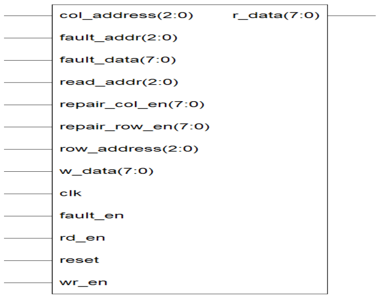


Fig 5: Repairable RAM block diagram

BIST Circuit:

It can generate test patterns using Linear feedback shift register (LFSR) for RAMs under test. While a fault in a defective RAM is detected by the BIST circuit, the faulty information is sent to the BIRA circuit.

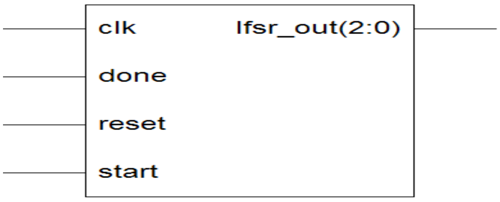


Fig 6: BIST LFSR block diagram

BIRA Circuit:

It collects the faulty information sent from the BIST circuit and allocates redundancies according to the collected faulty information using the implemented redundancy analysis algorithm.

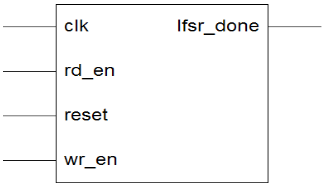


Fig 7: BISR block diagram

Fuse Macro:

It stores repair signatures of RAMs under test. Figure shows the conceptual block diagram of a typical implementation of fuse macro. The fuses of the fuse box can be implemented in different technologies, e.g., laser blown fuses, electronic-programmable fuses, etc. The fuse register is the transportation interface between the fuse box and the repair register in the repairable RAM.

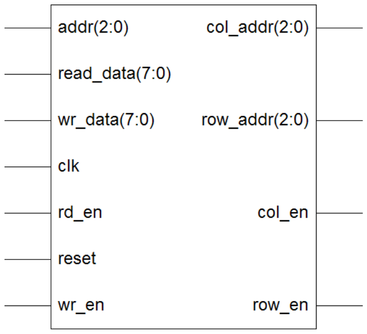
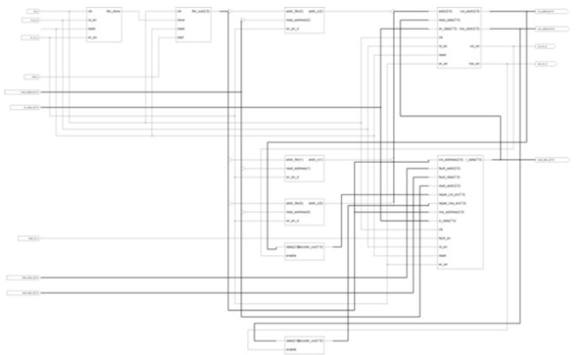


Fig 8: Fuse Macro block diagram



III. PROPOSED HRE-BISR SCHEME:

A.Architecture of HRE-BISR Scheme:

As aforementioned, the BIRA of a BISR circuit for a RAM with 2-D redundancy typically has a bitmap for storing fault information during the process of redundancy allocation. The bitmap is a cache-like element which has the parallel comparison and storing functions. Since the bitmap is onlyused in test mode for RA, we can modify it into a spare memory in normal mode. Fig. 5 shows the block diagram of the proposed HRE-BISR scheme for RAMs by reusing the bitmap as a spare memory. Here, we assume that the RAM has 2-D redundancy, e.g., spare rows and columns, and the reconfiguration of redundancies is done by shift redundancy technique [9]. The BIRA consists of a multiple-fault-bit detector (MFBD), a finite state machine (FSM), a repaired fault checker (RFC), a repair signature register (RSR), and a modified bitmap. The FSM realizes the RA algorithm. The MFBD is used to check if the number of faulty bits of adetected faulty word are larger than 2. The RSR is used to store the repaired addresses during the process of RA. Once the BIST detects a fault, the RFC checks if the detected fault has been stored in the RSR. If yes, the detected fault has been repaired. The modified bitmap is used to collect fault information in test mode and serves as spare bits in normal mode.

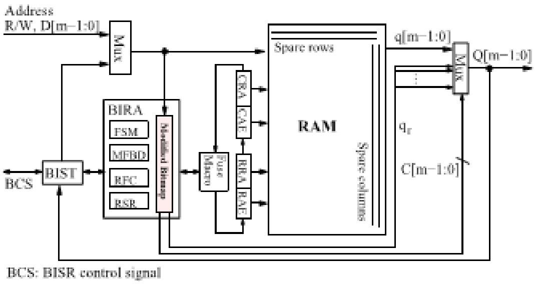


Fig. 5. Block diagram of the proposed HRE-BISR scheme.

Since the size of bitmap is typically much smaller than the size of a word of RAMs, the bitmap is used as a spare-bit memory. To reuse a bitmap as a spare memory, the bitmap should be modified such that it can support the read/write operation in normal mode. Fig. 6(a) shows the simplified block diagram of a modified BIRA with a k × l-bit modified bitmap.

A mode setting signal MODE is used to set the BIRA to be operated in normal or test mode. Normal IOs of the RAM are connected to the BIRA as well. In test mode, the operation of the modified bitmap is the same as that of the original bitmap. In normal mode, the operation of the modified bitmap is described as follows. If the RAM is accessed, the applied address is compared with the stored addresses including therow and column addresses. If one of stored row addresses is the same as the row address of the input address, then the corresponding row address hit (RAH) signal is set to 1, i.e., RAHi= 1. Similarly, if one of stored column addresses is the same as the column address of the input address, then thecorresponding column address hit (CAH) signal is set to 1, i.e., CAH j = 1. Therefore, when the accessed address is matched with one of the stored addresses, i.e., RAHi= 1 and CAH j = 1, one bit of the addressed word is repaired by the spare bit.

To support the read/write operation, each storage element of the bit array is modified, as shown in Fig. 6(b). If a write operation is executed, i.e., R/W = 1, the repair data input din is written into the cell bijOtherwise, the data stored in the cell bijis read out while R/W = 0. Furthermore, a repair bit selection (RBS) module is added to select the repair bit from the data input D [m – 1 : 0] for the data input din of bit array and generate the control signal C [m – 1 : 0] determining which bit of data output is replaced by the data output of the bit array qr. Fig. 6(c) shows the simplified circuit diagram of the RBS module. In test mode (i.e., Test = 1), the din is set to logic 1 such that the hit bit can be set to logic 1. In addition, the C [m – 1 : 0] is set to all-0 state, which forces Q [m – 1 : 0] = q [m – 1 : 0], as shown in Fig. 5. In normal mode (i.e., Test = 0), once the input address is matched with one of the stored addresses, RAHiand CAH j , the hit signal is asserted and the decoder is enabled. Simultaneously, the value of the corresponding BAR BARj is exported to the decoder. Then, the decoder decodes the BAR j into m-bit control signal C [m – 1 : 0] for controlling the data input and data output multiplexers.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

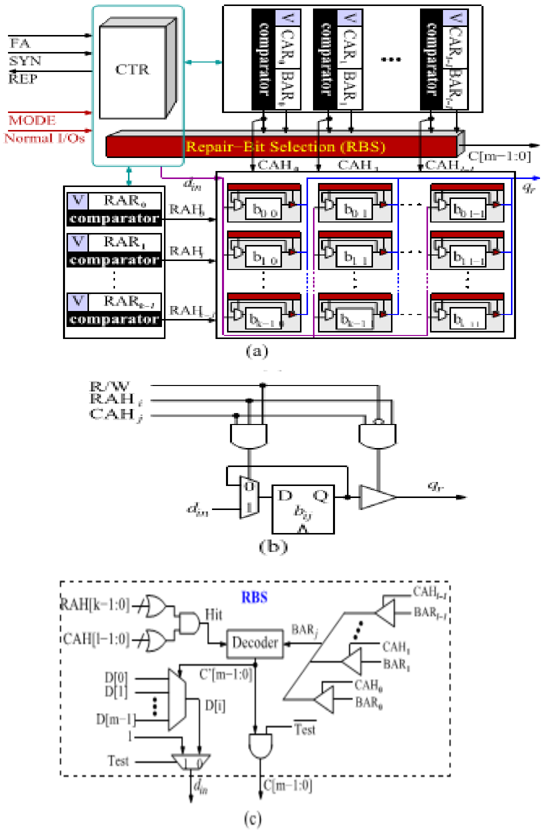


Fig. 6. (a) Simplified block diagram of the modified BIRA with k × l-bit modified bitmap. (b) Modified cell structure. (c) Simplified circuit diagram of RBS.

To reduce the hardware complexity, only one bit addressstored in the BARs is selected, as shown in Fig. 6(c). Therefore, the bitmap cannot repair multiple-faulty-bit words. If the bitmap stores a multiple-faulty-bit word, for example, then there are two CARs that store the same column address and thecorresponding BARs that store different bit addresses. Thus, if the column address stored in the CARs is accessed, the two different bit addresses are exported to the bus, which results in bus congestion.

One straightforward approach to support the repair of multiple-faulty-bit words is to design an individual decoder for each pair of CAHiand BARi. Then, the result of bitwise OR operation of the outputs of all decoders is theC [m – 1 : 0]. However, the area cost is drastically increased.Consequently, we modify the bitmap to support the repair of single-faulty-bit word only here.

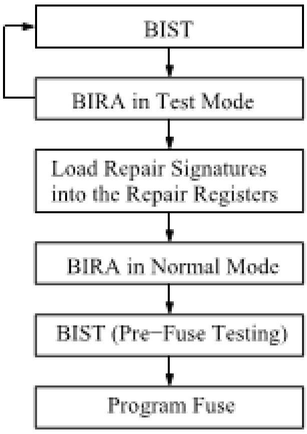


Fig. 7. Test and redundancy allocation flow

B.Test and Repair Flow:

In test mode, the test and redundancy allocation flow is as shown in Fig. 7. First, the BIST tests the RAM under repair and the BIRA is in test mode. If a fault is detected, the fault information is sent to the BIRA for analysis. This processis iterated until the testing of the RAM is completed. If the RAM is repairable, then the repair signatures are loaded into the repair registers and the BIRA are set to normal mode. Subsequently, the BIST is used to test the repaired RAM again for the prefuse testing. If the prefuse testing is completed and the repaired RAM is fault-free, then the fuses in the fuse macro can be programmed and the test and repair process is completed. Here, we assume that the electrically programmed fuse (e-fuse) is used to realize the fuses in the fuse macro [34].

In addition, the fuse macro has a fuse CTR to handle the programming of e-fuse. The e-fuse enables the function of on-chip self-repair, whereby repair data is programmed into e-fuse when the test and repair process is completed. Therefore, if the BIST and the fuse CTR are asserted by the poweron reset signal, on-chip self-repair can be achieved [10].Fig. 8 shows the repair flow of the proposed HRE-BISR scheme in normal mode. The repair flow consists of two phases: the repair setup and the normal access phases. In the repair setup phase, once the power is turned on, the repair signatures stored in fuse macro are loaded into the repair registers and the RAR, CAR, and BARs of the bitmap. In normal access phase, the applied address is imported to the repairable RAM and bitmap simultaneously.

If the applied address is not matched with the addresses stored in the bitmap,then the repairable RAM is accessed only. Otherwise, both the repairable RAM and the bitmap are accessed. Then, the faulty bit of accessed word in the repairable RAM is replaced by the hit spare bit of bitmap.

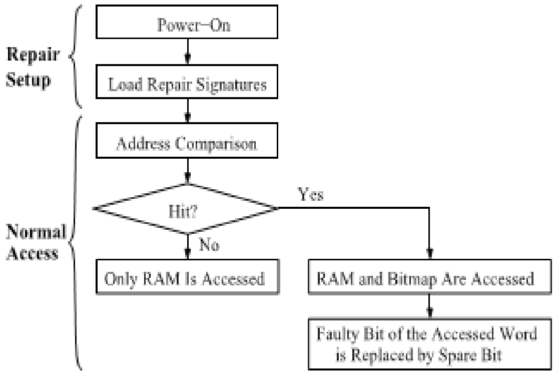
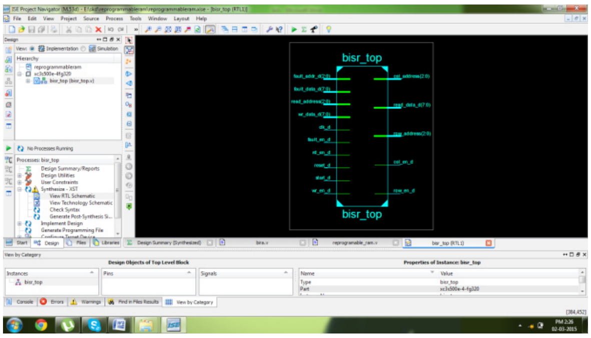


Fig. 8. Repair flow diagram.

Simulation results:



RTL SCHAMATIC:

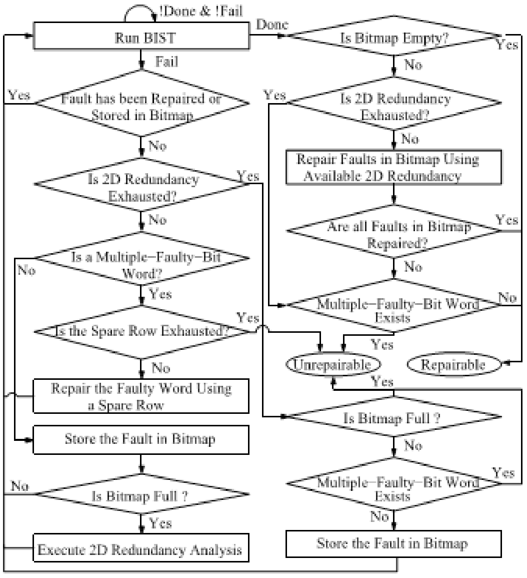
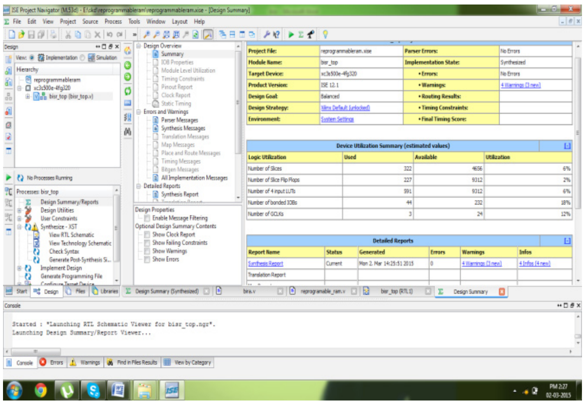

!Done & !Fail
Run BIST
Done
Fail
Yes
Fault has been Repaired or Stored in Bitmap
No
Is 2D Redundancy Exhausted?
Yes
No
Is a Multiple–Faulty–Bit Word?
No
Yes
Is the Spare Row Exhausted?
Yes
No
Repair the Faulty Word Using a Spare Row
Store the Fault in Bitmap
No
Is Bitmap Full ?
Yes
Execute 2D Redundancy Analysis
Is Bitmap Empty?
Yes
No
Is 2D Redundancy Exhausted?
Yes
No
Repair Faults in Bitmap Using Available 2D Redundancy
Are all Faults in Bitmap Repaired?
Yes
No
Multiple–Faulty–Bit Word Exists
No
Yes
Unrepairable
Repairable
Yes
Is Bitmap Full ?
No
Multiple–Faulty–Bit Word Exists
Yes
No
Store the Fault in Bitmap

Fig. 9. Flow of proposed RA algorithm for RAMs with 2-D and spare-bit redundancies.



Project File:	reprogrammableram.xise	Parser Errors:	No Errors
Module Name:	bisr_top	Implementation State:	Synthesized
Target Device:	xc3s500e-4fg320	• Errors:	No Errors
Product Version:	ISE 12.1	• Warnings:	4 Warnings (3 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)

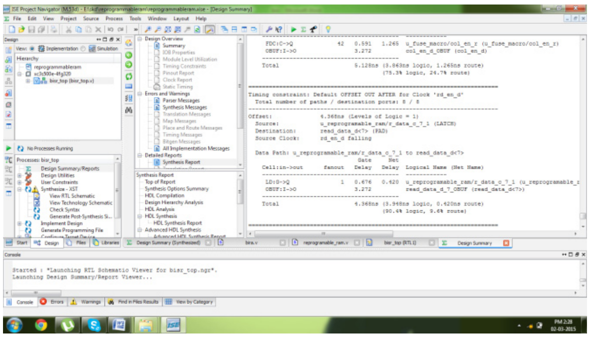
Logic Utilization	Used	Available	Utilization
Number of Slices	322	4656	6%
Number of Slice Flip Flops	227	9312	2%
Number of 4 input LUTs	591	9312	6%
Number of bonded IOBs	44	232	18%
Number of GCLKs	3	24	12%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Mon 2. Mar 14:25:51 2015	0	4 Warnings (3 new)	4 Infos (4 new)
Translation Report					

```
Started : "Launching RTL Schematic Viewer for bisr_top.ngr".
Launching Design Summary/Report Viewer...
```

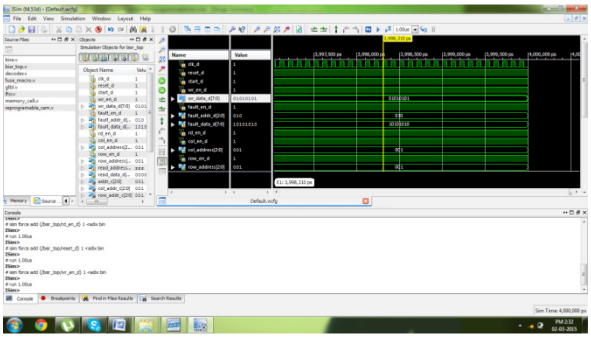
AREA CALCULATION:



```
    FDC:C->Q              42   0.591   1.265  u_fuse_macro/col_en_r (u_fuse_macro/col_en_r)
    OBUF:I->O                  3.272          col_en_d_OBUF (col_en_d)
   ----------------------------------------
   Total                       5.128ns (3.863ns logic, 1.265ns route)
                                       (75.3% logic, 24.7% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'rd_en_d'
  Total number of paths / destination ports: 8 / 8
-------------------------------------------------------------------------
Offset:              4.368ns (Levels of Logic = 1)
  Source:            u_reprogramable_ram/r_data_c_7_1 (LATCH)
  Destination:       read_data_d<7> (PAD)
  Source Clock:      rd_en_d falling

  Data Path: u_reprogramable_ram/r_data_c_7_1 to read_data_d<7>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     LD:G->Q               1   0.676   0.420  u_reprogramable_ram/r_data_c_7_1 (u_reprogramable_r
     OBUF:I->O                 3.272          read_data_d_7_OBUF (read_data_d<7>)
    ----------------------------------------
    Total                      4.368ns (3.948ns logic, 0.420ns route)
                                       (90.4% logic, 9.6% route)

=========================================================================
```

```
Started : "Launching RTL Schematic Viewer for bisr_top.ngr".
Launching Design Summary/Report Viewer...
```

DELAY CALCULATION:



WAVE FORMS:

CONCLUSION:

In this paper, we have proposed a HRE-BISR scheme for RAMs with 2-D redundancy. By reusing the bitmap for the RA to serve spare bits in normal mode, the HRE-BISR scheme can enhance the RR. To support the redundancy allocation of RAMs with 2-D and spare-bit redundancies, a RCB-RA algorithm has also been proposed.Simulation results show that the HRE-BISR scheme only incurs about 0.08-ns delay penalty for a 512 × 16 × 32-bit RAM using 3 × 3-bit bitmap for RA.

In addition, the RCB-RA algorithm can provide 0.48%–11.95% increment of RR for different fault distributions. Furthermore, only about 0.44% additional hardware overhead is needed to modify the bitmap as spare-bit redundancy.

REFERENCES:

[1] Y. Zorian and S. Shoukourian, "Embedded-memory test and repair: Infrastructure IP for SoC yield," IEEE Des. Test. Comput., vol. 20, no. 3, pp. 58–66, May/Jun. 2003.

[2] S. Shoukourian, V. Vardanian, and Y. Zorian, "SoC yield optimization via an embedded-memory test and repair infrastructure," IEEE Design Test Comput., vol. 21, no. 3, pp. 200–207, May/Jun. 2004.

[3] T. Kawagoe, J. Ohtani, M. Niiro, T. Ooishi, M. Hamada, and H. Hidaka, "A built-in self-repair analyzer (CRESTA) for embedded DRAMs," in Proc. Int. Test Conf. (ITC), 2000, pp. 567–574.

[4] D. Xiaogang, S. M. Reddy, W.-T. Cheng, J. Rayhawk, andN. Mukherjee, "At-speed built-in self-repair analyzer for embedded word-oriented memories," in Proc. Int.Conf.VLSI Design, 2004, pp. 895–900.

[5] P. Ohler, S. Hellebrand, and H.-J. Wunderlich, "An integrated built- in self-test and repair approach for memories with 2D redundancy," in Proc. 12th IEEE Eur. Test Symp. (ETS), Freiburg, Germany, May 2007, pp. 91–99.

[6] M.-H. Yang, H. Cho, W. Kang, and S. Kang, "EOF: Efficient built- in redundancy analysis methodology with optimal repair rate," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 7, pp. 1130–1135, Jul. 2010.

[7] W. Jeong, I. Kang, K. Jin, and S. Kang, "A fast built-in redundancy analysis for memories with optimal repair rate using a line-based search tree," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 12, pp. 1665–1678, Dec. 2009.

[8] W. Jeong, J. Lee, T. Han, K. Lee, and S. Kang, "An advanced BIRA for memories with an optimal repair rate and fast analysis speed by using a branch analyzer," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 12, pp. 2014–2026, Dec. 2010.

[9] D. K. Bhavsar, "An algorithm for row-column self-repair of RAMs and its implementation in the Alpha 21264," in Proc. Int. Test Conf. (ITC), Atlantic City, NJ, USA, Sep. 1999, pp. 311–318.

[10] V. Schober, S. Paul, and O. Picot, "Memory built-in self-repair using redundant words," in Proc. Int. Test Conf. (ITC), Baltimore, MD, USA, Oct. 2001, pp. 995–1001.