A Concurrent Self Repair Scheme for Defects in Random Access Memories

Sandeep Sivvam, Solomon Gotham

Abstract*—Built-in self-repair (BISR) techniques are widely used for repairing embedded random access memories (RAMs). One key component of a BISR module is the built-in redundancy-analysis (BIRA) design. This paper presents an effective BIRA scheme which executes the 2-D redundancy allocation based on a 1-D local bitmap. Two BIRA algorithms for supporting two different redundancy organizations are also proposed. Simulation results show that the proposed BIRA scheme can provide high repair rate (i.e.,the ratio of the number of repaired memories to the number of defective memories) for the RAMs with different fault distributions. Experimental results show that the hardware overhead of the BIRA design is only about 2.9% for an 8192 64-bit RAM with two spare rows and two spare columns. The design is implemented on Xiliinx Spartan3E FPGA and the device used 532 flip-flops out of 3840 available and 439 LUT's out of 3840 and the number of IO blocks used is 13. Moreover, the time overhead of redundancy analysis is very small. Embedded memories are among the most widely used cores in current system-on-chip (SOC) implementations. Total power utilized by the device was0.041mW. Memory cores usually occupy a significant portion of the chip area, and dominate the manufacturing yield of the chip. The BIRA module executes the proposed redundancy analysis (RA) algorithm for RAM with a 2-D redundancy structure, i.e., spare rows and spare columns.

***Index Terms*—BIRA, ReBIRA**

I. INTRODUCTION

Memories are key components of a typical system-onchip (SOC). They normally are dense and covers a large portion of the chip area, thus dominate the yield of the chip. Keeping the memory cores at a reasonable yield level is thus vital for SOC products. For such purpose, memory designers usually employ redundancy repair—using, e.g., spare rows and/or spare columns of cells—to improve the yield [1–4]. However, redundancy increases silicon area and thus has a negative impact on yield. To maximize the yield with a reasonable cost, redundancy analysis (RA) is necessary. Conventionally, equipment (ATE) if it is an on-line process, or on a separate computer if it is an offline process. Either way it is time consuming since RA algorithms are complicated and the memories that implement redundancies are usually large. Moreover, embedded memories are harder to deal with using ATE, and few believe that any known ATE architecture can accurately test tomorrow's system chips for the demanded yield and reliability[5]. The defective cells detected by the BIST circuit are replaced by the cells of the spare SRAM. The RA is performed on the host computer of the automatic test.

Revised Manuscript Received on September, 2012.

Sandeep Sivam, M.Tech ECE Department,JNTU Kakinada University/ Kaushik College of Engineering/Visakhapatnam, India.

Solomon Gotham, Professor & Head, Dept. of ECE, Kaushik College of Engineering , India.

Embedded random access memory (RAM) is one key component in modern complex system-on-chip (SOC) designs. Typically, many RAMs with various sizes are included in an SOC, and they occupy a significant portion of the chip area. Furthermore, RAMs are subject to aggressive design rules, such that they are more prone to manufacturing defects. That is, RAMs have more serious problems of yield and reliability than any other embedded cores in an SOC. Keeping the RAM cores at a reasonable yield level is thus vital for SOC products. Built-in self-repair (BISR) technique has been shown to improve the RAM yield efficiently. For example, the work in [1] shows that the BISR circuit can improve the RAM yield from 5% to 20%, such that the net SOC yield increase can range from 2% to 10%. Therefore, the BISR technique is a promising and popular solution for RAM yield improvement. Built-in redundancy-analysis (BIRA) algorithm is one key component of a BISR scheme, and it is responsible for allocating redundancies of RAMs under test. Thus, the BIRA circuit has heavily influence on the repair efficiency of the BISR scheme. If a RAM has only spare rows or spare columns, i.e., 1-D redundancy, then the redundancy allocation is simple and straightforward.

The BIRA module is programmable and one repair strategy is programmed to allocate redundancies each time. Therefore, if the programmed repair strategy cannot repair the memory under test, then another repair strategy is programmed in the BIRA module and the memory under test is retested. This process is repeatedly until a successful repair strategy is found or all possible repair strategies are tried. Although the area cost of the BIRA circuit is reduced, this results in very high time cost of test and repair. Therefore, some of existing BISR schemes use heuristic redundancy analysis algorithms to allocate the redundancy of RAMs under test.

II. TYPICAL MEMORY BISR ARCHITECTURE

Fig. 1 shows the block diagram of a typical BISR scheme for RAM, which consists of four major components.

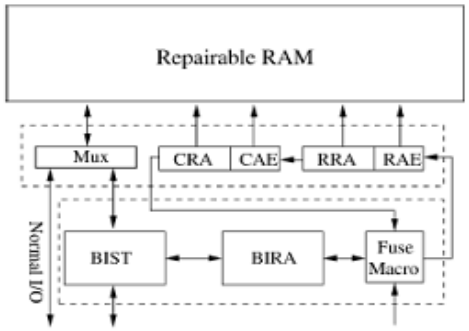

Fig. 1. Typical BISR scheme for embedded RAMs.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

1) Repairable RAM: A RAM with redundancies and reconfiguration circuit. Fig. 2 depicts an example of an 8x8 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.

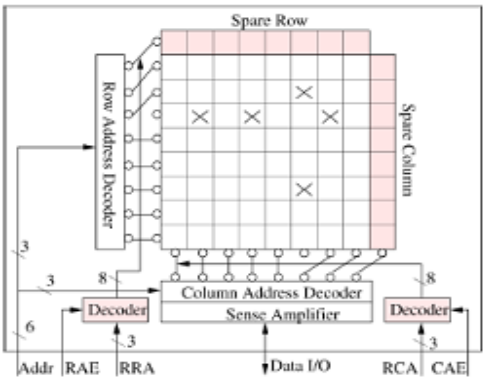

Fig. 2. Conceptual diagram of an 8□8 bit-oriented repairable RAM with one spare row and one spare column.

***2) Built-in Self-Test (BIST) Circuit*:** It can generate test patterns 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.

3) 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.

4) Fuse Macro: It stores repair signatures of RAMs under test. Fig. 3 shows the conceptual block diagram of a typical implementation of fuse macro [4]. 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.

III. ARCHITECTURE OF THE REBISR SCHEME

The proposed ReBISR scheme for repairing multiple repairable RAMs in an SOC is discussed below. The Wrapper of a RAM under test consists of multiplexers, a test pattern generator (TPG), and repair registers. The multiplexers switch the RAM between test/repair mode and normal mode. The TPG generates desired test patterns according to the given command from the test controller (CTR). The repair registers store the repair signatures. The CTR coordinates the operations of the TPG and the ReBIRA circuit. The Fuse Macro consists of the fuse and the fuse register. The number of bits of the fuse, the fuse registers, and the repair register is the same. Different technologies can be used to implement the fuse, e.g., the laser-blown fuse, the programmable electronic fuse, etc. If the laser-blown fuse is used, then the ReBISR only can repair the target RAMs onetime. If the programmable electronic fuse is used, then the Re-BISR can perform RAM repair multiple times. In our ReBISR scheme, moreover, the repairable RAMs can be equipped with one of the following two redundancy organizations: 1) spare rows and spare columns and 2) spare rows and spare IOs. As Fig. 4 shows, the CTR and ReBIRA circuits are shared by multiple RAMs. Therefore, the area cost of the Rebus circuit is reduced.

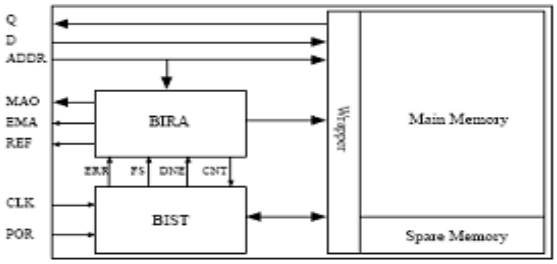

Figure 2. Block diagram of the proposed BISR scheme.

Above Fig shows the repair process of the proposed ReBISR scheme during test and repair phase. If the BIST detects a fault, then the fault information is exported to the ReBIRA circuitry, and then the ReBIRA performs redundancy allocation on the fly using the rules of the implemented redundancy algorithm (the proposed redundancy algorithm will be described in the next subsection). The ReBIRA allocating redundancy on the fly means that the redundancy allocation process and the BIST process are performed concurrently. The proposed ReBIRA scheme uses a local bitmap (i.e., a small bitmap) to store fault information of the faults detected by the BIST circuit. Once the local bitmap is full, the BIST is paused and the ReBIRA allocates redundancies according the the fault information. After the ReBIRA allocates a redundancy to repair a corresponding faulty row or column, the local bitmap is updated and the BIST is resumed. This process is iterated until the test and repair process is completed. Once one spare element is allocated, the ReBIRA module stores the corresponding repair signature in its Repair Signature Register. If the BIST is not completed, then the BIST continues to test the RAMs. When the BIST and BIRA are completed, the repair signatures stored in the Repair Signature Register are shifted into the Fuse Register of Fuse Macro through the RSO (repair signature output). Before programming the fuses, the user can first load repair signatures into the repair registers in the Wrappers. Then the BIST is used to test the repaired RAMs again. This is called *prefuse testing*. Subsequently, if the prefuse testing is completed and no fault is detected, the repair signatures in the Fuse Macro are exported to the fuse-programming equipment or circuit through TDO.Then the repair signatures can be programmed into the fuses. Note that the prefuse testing can be optional. The user can directly program the fuse without executing the prefuse testing if programmable fuses are used. Fig. 5(b) shows the repair process during the normal operation phase. Note that if a soft repair strategy (only registers are used to store repair signatures) is used in the ReBISR scheme, then this phase is not needed. In the life time of the repaired Rams, once the power of the RAMs is turned on, the repair signatures stored in the fuses are loaded into the

Fuse Register of the Fuse Macro by asserting the signal LD. Then the repair signatures in the Fuse Register are shifted into the repair registers in Wrappers through the Fuse input (FI) and Fuse output (FO). The time for setting the repair signatures is called *repair setup time*. Once the repair signatures have been loaded into the repair registers, the RAMs can be accessed normally. In an SOC, multiple ReBISR circuits are allowed. Thus, RAMs in the SOC can be divided into groups and each group is served by one ReBISR circuit.

IV. POWER ON REBISR

The BISR procedure is shown in Fig. 3. Upon turning on the power, the BIST module starts to test the spare memory. Once a fault is detected, it informs the BIRA module to mark the defective spare row or column as faulty through the error (ERR) and fault syndrome (FS) signals. After finishing the spare memory test, it tests the main memory. If a fault is detected (ERR outputs a pulse), the test process pauses and the BIST module exports the FS to the BIRA module, which then performs the RA procedure.

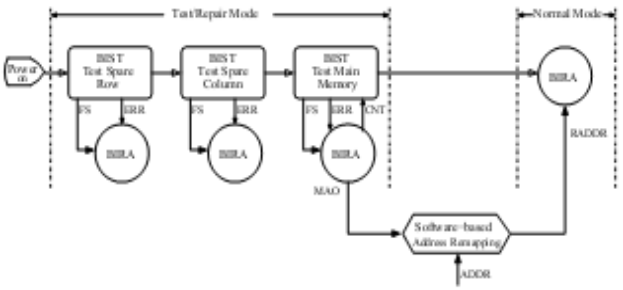

Figure.3. Power-on BISR procedure.

When the procedure is completed and the memory testing is not finished yet, the BIRA module issues a continue signal (CNT) to resume the test process. During the RA procedure, if a spare row is requested but there is no more spare row, the BIRA module exports the faulty row address through the EMA (export mask address) and MAO (mask address output)

signals. The memory will then be operated at a down-graded mode (i.e., with a smaller usable capacity) by software-based address remapping. For example, assume that a memory with multiple blocks is used for buffering, and the blocks are chained by pointers. If some block is faulty and should be masked, then the pointers are updated to invalidate the block.

The size of the memory is reduced, as one block is removed.

The system still works if a smaller buffer is allowed, though the performance may be affected. This approach effectively increases the yield of the products. The number of blocks that can be invalidated normally depends on the performance penalty that can be tolerated. If the down-grade mode is not allowed, the MAO is removed and the EMA indicates whether the memory is repairable. When the main memory test and RA are finished, the REF (repair end flag) signal goes high and the BIRA module switches to the normal mode. The BIRA module then serves as the address remapper, and the memory can be accessed using the original address (ADDR). When the memory is accessed, ADDR is compared with the fault addresses stored in the BIRA module. If ADDR is the same as any of the fault address, the BIRA module controls the wrapper to remap the access to the spare memory.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	134	3,840	3%
Number of 4 input LUTs	439	3,840	11%
Number of occupied Slices	277	1,920	14%
Number of Slices containing only related logic	277	277	100%
Number of Slices containing unrelated logic	0	277	0%
Total Number of 4 input LUTs	532	3,840	13%
Number used as logic	439		
Number used as a route-thru	93		
Number of bonded IOBs	13	173	7%
Number of BUFGMUXs	1	8	12%
Average Fanout of Non-Clock Nets	3.33		

V. CONCLUSION

We have proposed a BISR scheme for RAM. The BISR circuit is composed of a BIST module and a BIRA module. The BIST circuit supports three operation modes— main memories testing, spare memory testing, and repair. The BIRA circuit executes a proposed RA algorithm for 2- D redundancy—spare rows and spare columns. The spare columns are grouped and segmented. A software-based address remapping (masking) is performed in the downgraded operation mode, where certain amount of un-repairable faulty rows can be tolerated. The experimental results show that high repair rate can be obtained. Compared with the conventional approach (without grouping and segmentation) using exhaustive search, the proposed scheme outperforms in many instances and can be implemented with low area cost. A BISR design for an industrial 8K_64 SRAM with 4 spare rows and 2 spare column groups has also been implemented. In this case, full repair can be achieved if the number of random faults is no more than 10. Moreover, the area overhead of the BISR design is low—about only 4.6% for the 8K_64 SRAM.

REFERENCES

1. S. E. Schuster, ―Multiple word/bit line redundancy for semiconductor memories‖, IEEE Journal of Solid- State Circuits, vol. 13, no. 5, pp. 698–703.
2. M. Horiguchi, J. Etoh, M. Masakazu, K. Itoh, and T. Matsumoto, ―A flexible redundancy technique for high-density DRAM's‖, IEEE Journal of Solid-State Circuits, vol. 26, no. 1, pp. 12–17.
3. T. Yamagata, H. Sato, K. Fujita, Y. Nishimura, and K. Anami, ―A distributed globally replaceable redundancy scheme for sub-half-micron ULSI memories and beyond‖, IEEE Journal of Solid-State Circuits, vol. 31, no. 2, pp. 195–201, Feb. 1996.
4. I. Kim, Y. Zorian, G. Komoriya, H. Pham, F. P. Higgins, and J. L. Lweandowski, ―Built in self repair for embedded high density SRAM‖, in Proc. Int. Test Conf. (ITC), Oct. 1998, pp. 1112–1119.
5. S. Runyon, ―Testing big chips becomes an internal affair‖, IEEE Spectrum, pp. 49–55, Apr. 2006.
6. C.-T. Huang, J.-R. Huang, C.-F. Wu, C.-W. Wu, and T.-Y. Chang, ―A programmable BIST core for embedded DRAM‖, IEEE Design & Test of Computers, vol. 16, no. 1, pp. 59–70, Jan.-Mar. 2009.