Vrezh Sargsyan
National Research University «MIET»
Russia, Zelenograd
E-Mail: s.vrezh@gmail.com

Memory reconfiguration for system-on-chip yield improvement

Abstract: One of the most important issues in the design and manufacturing process of system-on-chip is the difficulty of achieving a profitable chip yield. In modern system-on-chip embedded memories are the dominating components. System chips usually contain hundreds, and in some cases - thousands of different types of memory elements. And hence, overall chip yield becomes largely dependent on memory yield. Modern system-on-chips include dedicated built–in infrastructures intended for testing, diagnosis and repair of embedded memory devices. The advantages of using built–in infrastructures are the absence of any additional external equipment and relatively low cost, as well as the ability to test the device by user. Memory repair is performed by disabling memory defective elements (row / column) and enabling redundant elements based on repair signature.

In this paper, memory array reconfiguration mechanism in described. Memory built-in self-test and repair infrastructure is designed, which significantly reduces the memory repair signature loading time.

Keywords: System-on-chip; intellectual property; embedded test and repair; yield; embedded memory deuce; memory reconfiguration; integrated circuit; power domain, simulation and synthesis.

Identification number of article 170TAVN214

1. Introduction

With the progress in the VLSI (very large scale integration) design and fabrication process, a complex system can now be integrated into a single chip, known as system-on-chip (SOC) [1]. Modern SoC's move from logic-dominant to memory-dominant chips containing several thousands of memory cores, designed with the most aggressive rules and representing a significant portion of the chip area [2-3].

The use of external tester is traditional way for memory test and repair. However, the use of external equipment often constitutes as much as 40% of a chip's overall manufacturing cost and provides limited repair efficiency [3-4]. Infrastructure IP is a viable solution to many issues related with memory test and repair process. The infrastructure IP in general is a dedicated module, which is separate from the functional circuitry of the IC design [5]. Examples of such infrastructure IP are Built-in-Self-Test (BIST) for logic and memories, Built-in-Self-Repair (BISR) for embedded memories, embedded core test logic for SoC's. There are various solutions of BIST implementing schemes. Figure 1 show the composite infrastructure BIST/BISR IP [6-7].

In this paper we described memory reconfiguration signature loading mechanism and designed FCU (fuse control unite) module, which optimizes memory repair process.

The rest of the paper is structured as follows. In section 2 we discuss hardware components of embedded memory test and repair infrastructure IP and discuss reconfiguration signature delivery process. In Section 3 we present a novel optimized memory reconfiguration signature loading mechanism. In Section 4 we bring experimental results and compare the modified circuit parameter with original circuit. Finally, Section 5 presents our conclusions and further perspectives.

2. Memory reconfiguration signature delivery process

BIST/BISR infrastructure IP is hierarchical system, performing memory test, diagnosis and repair (see Figure 1). Each memory group communicates with the BIST Processor through the IEEE 1500 standard compliant Intelligent Wrapper (IW) for testing and repairing the memory groups. BIST Processor executes the test algorithms and controls the entire test and repair process. The FCU exercise the interconnection between IEEE 1149.1 Standard JTAG Test Access Port (TAP) and IEEE 1500 compliant dedicated IP interface, connects chip level signals to memory groups, schedules the test execution and handles repair signature dataflow. Because of FCU block manages memory reconfiguration signature dataflow, further we will consider peculiarities of this module.

After BIST Processor executes the test algorithm, BIST Processor collects the responses. If defects are found, BIST Processor determines the best way to allocate redundant resources to replace any defective element and generate reconfiguration data. After reconfiguration data is generated, the following steps of hard repair are done:

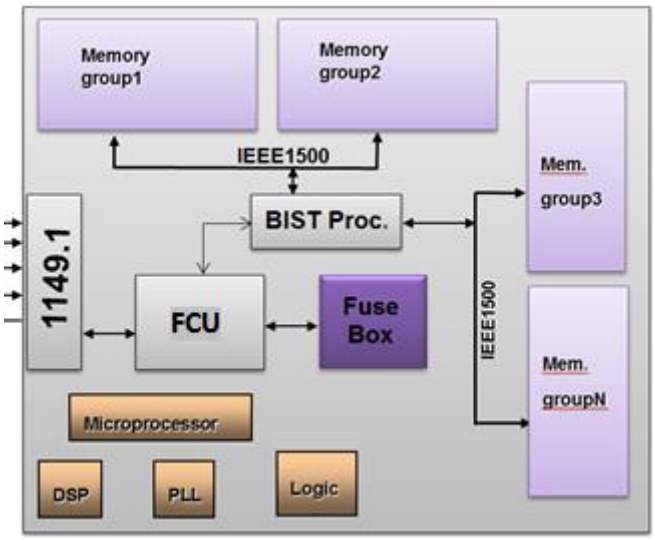

Figure 1. *Embedded memory self-test and repair infrastructure IP*

1. Repair signature transfer to permanent storage (eFUSE). The signature collected in MRR is transferred to FCU and programed in eFUSE boxes (see Figure 2). For complex memory groups, repair signature is usually long chain, and hence FCU may contain an encoding (compression) algorithm. In such cases, repair signature is first compressed, and then programed in eFUSE boxes.

2. After every power-up FCU reads the repair signature from eFUSE, decodes, and loads into the MRR.

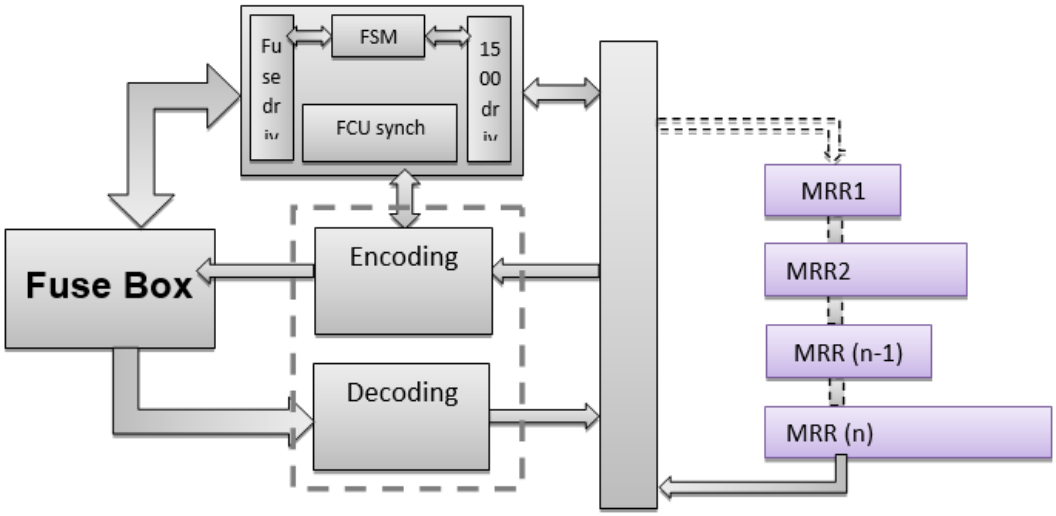

Figure 2. *Repair signature dataflow, FCU module*

The testing of modern memory groups need to have flexible tests such as corner conditions for different process, voltage, and temperature levels (PVT tests). In these types of tests the power of a particular memory group can be switched several times. Hence the second process, loading repair signature to MRR, is repeated as long as the test requires power switches. The second step, itself consists of the following.

- Loading RSC_CHAIN (reconfiguration chain) instructions into IR (instruction register) of all BIST processors containing repairable memories. The memory BIST processors containing non-repairable memories are loaded with BYPASS instruction.
- Reading repair signature from eFUSE, decoding and loading MRR.

- Loading BYPASS instruction into IR of all BIST processors, putting them into functional mode.

The BYPASS is IEEE1500 standard mandatory instruction [7], which puts the DUT in functional mode; RSC_CHAIN instruction activates MRR's.

3. Reconfiguration signature loading mechanism

The process described in previews chapter, have the following timing formula:

$$T = T(1) + T(2) + T(3)$$

$T(1) = system\_rsc\_chain\_length$: is the time, which is consumed for shifting reconfiguration signature to MRR.

$T(2) = (total\_num\_of\_fuses * 2) + compres\_factor * (num\_of\_compr\_els + num\_of\_un\ compr\_els)$: is the time which is consumed for reading signature from eFUSE. In our case, we use generic compression algorithm. The second component describes decompression process timing.

$T(3) = num\_of\_mems * 22$: is the time which is consumed for loading REC_CHAIN and BYPASS instructions to IR of BIST processors containing repairable memories.

We suggest using signature container in the FCU clock domain. This signature container will be loaded with repair signature at first BIHR (Built-In Hard-repair) run. In the next BIHR runs, the repair signature will be scanned out and shifted to MRR from signature container, skipping the fuse reading and signature decompression processes. The timing formula will be as follows:

$$T = T(1) + T(3)$$

In the Figure 3 the FCU architecture is illustrated with embedded signature container.

As it is shown, a serially connected registers and controlling logic are placed in FCU controller block, which is used as a container for storing reconfiguration signature (cont.).

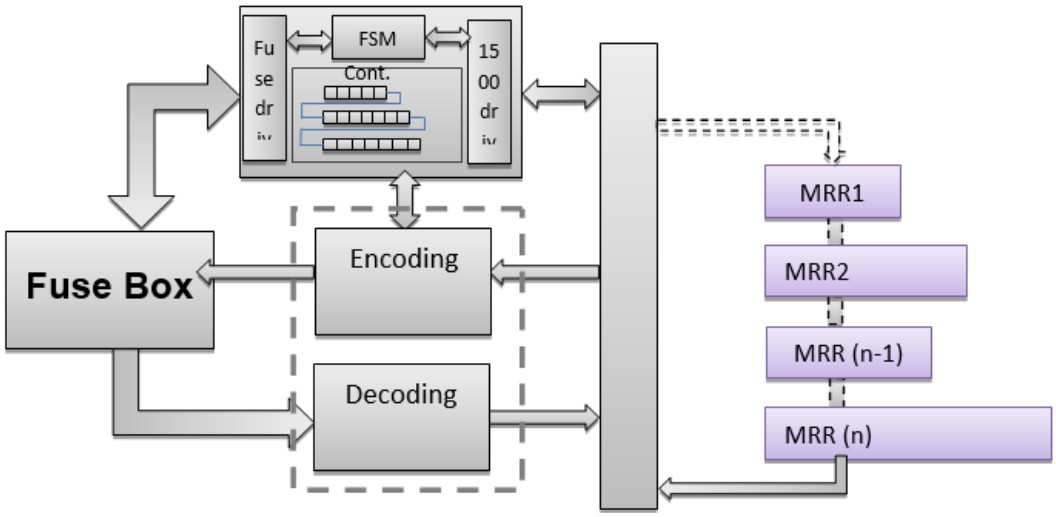

Figure 3. *Block diagram of FCU module with signature container*

4. Experimental results

In this section, we compare the simulation and synthesis results obtained from the original and the modified FCU circuit. We have measured the FCU module area, as well as we calculated the repair signature load cycles for the original circuit, and for modified circuit.

For our experiments, we used the following memory group configuration (see Table 1). The optimization of loading process mainly depends on memory reconfiguration chain size.

Table 1

Memory group configuration

Group	Memory configuration		Reconfiguration chain size
	Size	Instance count	
G1	128x640	3	72
	9x512	1	
G2	128x640	5	132
	16x8	4	
G3	128x640	4	716
	99x256	6	
	99x256	6	

Table 2

Simulation results

Project configuration	Area/ gate count	RSCR load time/ clock cycles	Load time saving
Without copies	8416	1320	
With all copies	13168	820	37.8%

Simulation and synthesis results are shown in Table 2. As you can see from simulation results, the optimization of repair signature loading time has a liner dependency from overall reconfiguration chain size.

Meanwhile, it is important to note, that signature container will add area overhead proportional to reconfiguration chain size. This kind of estimation also is useful to find best tradeoff between area over head and load time, allowing SoC designers beforehand distribute appropriate memory groups.

5. Conclusions

Memory reconfiguration information loading mechanism is introduced. These method reduces reconfiguration signature loading time and allows beforehand distribute memory groups, finding best tradeoff between reconfiguration signature load time and BIST circuit area overhead. However, there are some limitation and shortcomings. One of them is the area overhead which is proportional to MRR size. Another one is the limitation of memory group bypassing mechanism, in first cycle of delivery process. Further investigation is focused on finding ways of having selectable containers.

REFERENCES

1. Gupta, R. K. and Zorian, Y. “Introducing Core-Based System Design,” IEEE Design & Test of Computers, 14(4), pp 15-25, (1997).
2. Zorian, Y. Marinissen, E. and Dey, S. “Testing Embedded-Core-based System Chips,” Proceedings of the International Test Conference (ITC), pp 130-143, (1998).
3. A. Allan et al., “2001 Technology Roadmap for Semiconductors,” Computer, vol. 35, no.1, Jan. 2002, pp. 42-53.
4. Y. Zorian, “Embedded memory test and repair: Infrastructure IP for SoC yield,” in Proc. ITC, Oct. 2002, pp. 340–349
5. J. Dreibelbis, J. Barth, H. Kalter, and R. Kho. Processor based built-in self-test for embedded DRAM. IEEE Journal of Solid-State Circuits, pages 1731-1740, Nov. 1998.
6. Y. Zorian, “Guest Editor Introduction: What is Infrastructure IP?”, IEEE Design and Test of Computers, vol. 19, no. 3, May-June 2002, pp. 5-7.
7. IEEE 1500 Web Site. http://grouper.ieee.org/groups/1500/

УДК 004.031.6, 004.31

Саргсян Вреж Каренович
Национальный исследовательский университет «МИЭТ»
Россия, Зеленоград[1]
Аспирант кафедры ПКИМС
E-Mail: s.vrezh@gmail.com

Реконфигурация памяти для повышения процента выхода годных систем-на-кристалле

Аннотация: Одним из важнейших вопросов в проектировании и процессе производства систем-на-кристалле является сложность достижения прибыльного уровеня выхода годных микросхем. В современных системах на кристалле доминирующими компонентоми являются встроенные устройства памяти. Системные чипы содержат, как правило, сотни, а в некоторых случаях — тысячи элементов памяти различных типов. Следовательно, процент выхода годных микросхем становится в значительной степени зависимым от процента выхода годных устройств памяти. Современные системы-на-кристалле включают в себя специальные встроенные инфраструктуры, предназначенные для тестирования, диагностирования и ремонта устройств памяти. Преимуществами использования встроенных инфраструктур самотестирования являются отсутствие необходимости использования какого-либо внешнего дополнительного оборудования и относительно небольшая стоимость, а также возможность тестирования устройства конечным пользователем. Ремонт памяти осуществляется путем отключения дефектных элементов(строк и/ столбцов) и подключения резервных элементов на основе сигнатуры по восстановлению.

В данной статье описан механизм реконфигурации матрицы памяти и реализована встроенная инфраструктура самотестирования и ремонта памяти, которая значительно сокращает время загрузки сигнатуры по восстановлению работоспособности памяти.

Ключевые слова: Система-на-кристалле; сложнофункциональный блок; встроенное тестирование и ремонт; процент выхода годных; встроенное устройство памяти; реконфигурация памяти; интегральная схема; домен питания; моделирование и синтез.

Идентификационный номер статьи в журнале 170TAVN214

[1] 124498, Москва, Зеленоград, проезд 4806, дом 5.

ЛИТЕРАТУРА

1. Р. К. Гупта и Е. Зорян, “Introducing Core-Based System Design,” IEEE Design & Test of Computers, 14(4), ст 15-25, (1997).
2. Е. Зорян, Е. Мариниссен и С. Дей, “Testing Embedded-Core-based System Chips,” Proceedings of the International Test Conference (ITC), ст. 130-143, (1998).
3. А. Аллан , “2001 Technology Roadmap for Semiconductors,” Computer, том. 35, № 1, 2002, ст. 42-53.
4. Е. Зорян, “Embedded memory test and repair: Infrastructure IP for SoC yield,” in Proc. ITC, Окт. 2002, ст. 340–349
5. Дж. Дреибелбис, Дж. Бартх, Х. Калтер, и Р. Кхо. “Processor based built-in self-test for embedded DRAM”, IEEE Journal of Solid-State Circuits, ст. 1731-1740, Ноябрь 1998.
6. Е. Зорян, “Guest Editor Introduction: What is Infrastructure IP?”, IEEE Design and Test of Computers, том. 19, № 3, May-June 2002, ст. 5-7.
7. Домашняя страница IEEE 1500 http://grouper.ieee.org/groups/1500/

Рецензат: Шукурян Самвел Кимович, д.ф.-м.н., профессор, действительный член Национальной Академии наук Республики Армения, старший управляющий отдела вложенного тестирования и ремонта компании «Синопсис», Samvel.Shoukourian@synopsys.com