On Using Cell-Aware Models for Representing SRAM Architecture

Xhesila Xhafa, Aymen Ladhar, Eric Faehn, Lorena Anghel, Gregory Di Pendina, Patrick Girard, Arnaud Virazel

On Using Cell-Aware Models for Representing SRAM Architecture

X. Xhafa[1], A. Ladhar[2], E. Faehn[2], L. Anghel[3], G. Di Pendina[3], P. Girard[1], A. Virazel[1]

[1]LIRMM - Univ.of Montpellier/CNRS
Montpellier, France
xxhafa, girard, virazel@lirmm.fr

[2]STMicroelectronics
Crolles, France
aymen.ladhar@st.com, eric.faehn@st.com

[3]SPINTEC - CEA
Grenoble, France
lorena.anghel@phelma.grenoble-inp.fr,
gregory.dipendina@cea.fr

***Abstract*—This paper presents a novel approach to memory testing which relies on Cell-Aware (CA) test to further improve the yield in SoCs. Therefore, using CA test shifts the memory testing methodology from functional to structural testing. In this paper, a 4 by 4 SRAM architecture is used as a case study. Through transistor to gate-level transformation, the SRAM is represented as an ensemble of "special" standard cells on which the CA test can then be applied.**

***Index Terms*—Memory testing, SRAM, Cell-Aware Test**

I. Introduction

With the increase in demand for fast processing of large amounts of data in today's applications of integrated circuits, memory blocks may represent more than 90% of the system-on-chip (SoC) area. The shrinking of technology nodes has also led to high density memories containing large amounts of transistors which are prone to defects and reliability issues. Hence, a crucial step into preventing defective memories such as SRAMs, is to efficiently test these memories, so that the required defect-part-per-million (DPPM) criteria can be met.

A commonly practiced approach in testing memories is using March algorithms [1]. These algorithms consist of a sequence of operations which are sequentially applied to a specific memory cell, hence targeting previously modeled faults. The most prevalent type of fault models is the functional fault models (FFMs), which are models generated considering potential realistic defects in the memory [2]. This approach has dominated memory testing in the past few decades. However, with shrinking technologies, the need for novel and more efficient testing methodologies arises.

Cell-Aware (CA) test is a prominent testing methodology which has been developed to further reduce the defective-parts-per-million (DPPM) in digital ICs, consequently increasing the quality of products delivered to customers [3]. The need for such an approach has arisen as a result of several studies in the field, which have shown that most of the escapes during test can be attributed to defects within cells, rather than just in interconnections [4]. An important property of CA test is that it also includes the layout information when creating cell-aware models, hence leading to a more accurate representation of possible manufacturing defects.

The CA method is predominantly used in testing digital circuits composed of standard cells. However, a novel approach in the quest of further reducing DPPM in SoCs, would be to adapt CA test for advanced and emerging memories. This is the general purpose of our work. The theoretical process flow of developing CA models for memory testing is shown in Fig. 1. The first step in realizing the latter, is to describe the memory at the gate-level, where each memory block is represented a "special" standard cell, so that CA models can be used like in the case of logic standard cells. This paper presents the first step of the flow on a simple 4 by 4 SRAM memory architecture.

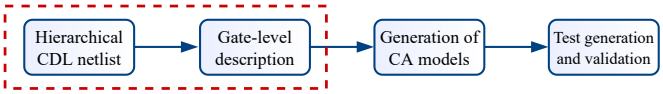

Fig. 1. The flow of testing memory circuits by using CA approach

II. Design of a 4 by 4 SRAM Architecture

In order to devise a working flow for the gate level description of the memory, a simple 4 by 4 SRAM architecture has been used as a case study. The architecture, presented in Fig. 2, has been designed at transistor level using a 28nm technology. A standard 6T topology, composed of two back to back inverters and two NMOS access transistors, has been used for the SRAM bit cell. Each bit line (BL) and bit line bar (BLB) are connected to load capacitors, which are pre-charged to VDD before read and write operations.

The row and column decoders have also been designed at transistor level since a common practice in industrial memory design is to use custom-made digital blocks. The designed sense amplifier (SA) is an inverter based differential amplifier with an access signal activated only during the read cycle. The used write driver topology is presented in Section III.This block includes a write signal controller block which distinguishes between Write 1 (W1) and Write 0 (W0) signals and then transmits this information to bit and bit bar lines through the rest of the write driver topology. Figure 3 shows the simulation of the memory when writing and reading a zero and a one on a single bit, consecutively.

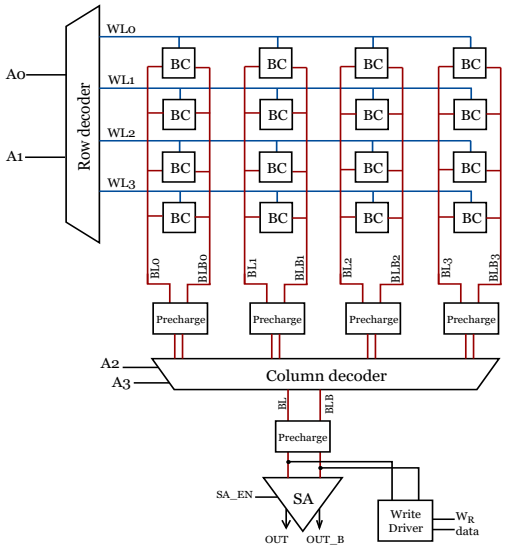

Fig. 2. The 4 by 4 SRAM architecture

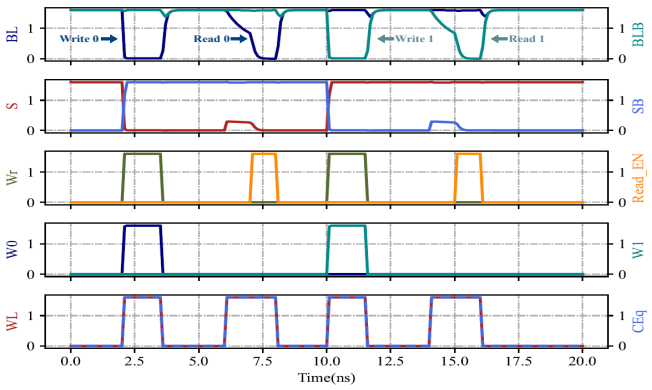

Fig. 3. Simulation of a single bit-cell during read and write operations

III. Gate-level description of the designed memory

The Spice netlist of the aforementioned 4 by 4 SRAM architecture has been organized in a hierarchical manner. This decision in purposefully taken to help the next step of the flow shown in Fig. 1. In order to be able to use CA test, a gate-level description of the memory is mandatory. After extracting the hierarchical CDL netlist, a functional verification tool, namely ESP, which is commonly used to perform equivalence checking between Spice and Verilog netlists, has been employed. This tool can use gate or switch primitives to describe the initial netlist into a gate-level netlist. Since the Spice netlist of the SRAM memory is at transistor level, ESP interprets NMOS and PMOS as transistor primitives, while still retaining the hierarchical structure of the initial netlist. Depending on the structure of each sub-circuit, the tool can interpret the transistors as bidirectional (trainif0/1). This issue has been solved through a python script which detects the latter misinterpretation and substitutes the primitives with the correct ones (n/pmos).

The SRAM memory is composed of combinational, sequential and analog circuit blocks. The combinational (e.g decoders) and sequential blocks (e.g scan-chain) can be straightforwardly represented at gate-level, using the previously described flow. However, for the SA block, a delay constraint needs to be added before enabling it. This allows BL and BLB to differ significantly in voltage levels, so that the SA can read the values correctly. Moreover, initial conditions for BL and BLB are used to emulate the load capacitances, which are not representable at gate-level.

To demonstrate the latter process, an example is given in Fig. 4. The schematic shows the topology of the write driver block. The Spice netlist of this block and the necessary transistor model information has been used as an input to ESP. A Verilog file has then been generated and its equivalence has been verified using gate-level simulations. The right part of Fig. 4 shows the simulation results when a sequence of "001011" data has been used as an input, while the write signal (Wr) is kept at 1. The outputs BL and BLB show the expected results, where BL perfectly follows the data and BLB inverts it.

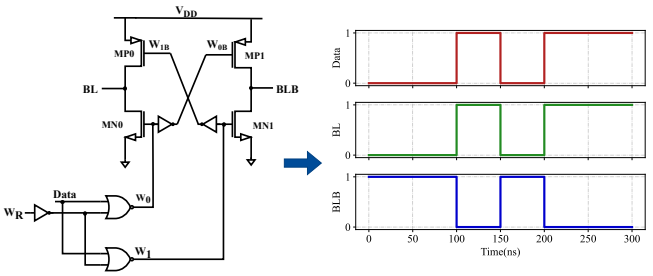

Fig. 4. Gate-level simulation of the write driver

IV. Conclusion

This paper demonstrates that it is indeed possible to represent an SRAM at gate-level by using "special" standard cells. The next step of our work will be to use CA models to generate test patterns and hence be able to test SRAM as one could do for logic circuits. The ultimate goal would be to adapt a similar flow based on CA models for testing other emerging memory types, such as STT and SOT-MRAMs.

References

[1] A. J. Van de Goor, *Testing semiconductor memories: theory and practice*. John Wiley & Sons, Inc., 1991.

[2] A. Bosio, L. Dilillo, P. Girard, S. Pravossoudovitch, and A. Virazel, "Advanced test methods for srams," in *2012 IEEE 30th VLSI Test Symposium (VTS)*, pp. 300–301, IEEE, 2012.

[3] F. Hapke, W. Redemund, A. Glowatz, J. Rajski, M. Reese, M. Hustava, M. Keim, J. Schloeffel, and A. Fast, "Cell-aware test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 9, pp. 1396–1409, 2014.

[4] S. Eichenberger, J. Geuzebroek, C. Hora, B. Kruseman, and A. Majhi, "Towards a world without test escapes: The use of volume diagnosis to improve test quality," in *2008 IEEE International Test Conference*, pp. 1–10, IEEE, 2008.